



Universidad de Valladolid

Departamento de Informática

TESIS DOCTORAL

**SELECCIÓN DE SOCIOS EN LAS
EMPRESAS VIRTUALES
DINÁMICAS**

Pedro Sanz Angulo

2008

TESIS DOCTORAL

Programa de Doctorado en Informática

SELECCIÓN DE SOCIOS EN LAS EMPRESAS VIRTUALES DINÁMICAS

Departamento de Informática
Universidad de Valladolid

Autor: D. Pedro Sanz Angulo

Director: Dr. D. Juan José de Benito Martín

Valladolid, octubre de 2008

*A Elena y Mariana,
por estar siempre a mi lado
y a Daniel,
que pronto lo estará*

AGRADECIMIENTOS

Quisiera dedicar las siguientes líneas a expresar mi más sincero agradecimiento a todas aquellas personas que, de una forma u otra, han contribuido a la realización de este trabajo.

Al profesor Dr. D. Juan José de Benito Martín, no sólo por sus consejos, aportaciones y apoyo incondicional en la realización de este trabajo como director de la tesis, sino por su labor como mentor en mi vida académica y de investigación.

Al Profesor Dr. D. Pablo Lucio de la Fuente Redondo, ponente de la tesis, por sus aportaciones y su infinita comprensión y disponibilidad.

Al Profesor Dr. D. Cesáreo Hernández Iglesias y al resto de mis compañeros del Departamento de Organización de Empresas y CIM por el interés que han mostrado y por el apoyo recibido.

Gracias también a Craig A. Smith, Software Licensing Manager de Sandia National Laboratories, por todas las facilidades que me ha conferido para emplear el código Jess en esta investigación.

Finalmente, deseo agradecer a toda mi familia y a todos mis amig@s, grandes y pequeños, su estímulo, apoyo y cariño en estos años de arduo trabajo en los que las ausencias han sido significativas.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
ANTECEDENTES.....	1
MOTIVACIÓN.....	6
OBJETIVOS.....	9
ORGANIZACIÓN DEL DOCUMENTO.....	11
CAPÍTULO 1 EL PARADIGMA DE EMPRESA VIRTUAL.....	15
1.1 INTRODUCCIÓN.....	15
1.2 EL NUEVO ENTORNO EMPRESARIAL.....	20
1.3 DEFINICIÓN.....	25
1.3.1 <i>Conceptos relacionados</i>	26
1.3.2 <i>Definiciones de empresa virtual</i>	29
1.4 CRITERIOS DE CLASIFICACIÓN.....	33
1.5 TIPOLOGÍAS.....	40
1.5.1 <i>Empresas virtuales intra e interorganizacionales</i>	40
1.5.2 <i>Empresas virtuales estáticas</i>	41
1.5.3 <i>Empresas virtuales dinámicas</i>	43
1.5.4 <i>VE estáticas versus VE dinámicas/ágiles</i>	48
1.6 VIRTUALIDAD Y VIRTUALIZACIÓN.....	52
1.7 RESUMEN.....	58

CAPÍTULO 2 EL CICLO DE VIDA DE LA VE DINÁMICA Y LOS ENTORNOS DE GESTACIÓN.....61

2.1	INTRODUCCIÓN.....	61
2.2	EL MODELO DE CICLO DE VIDA DE LAS DVE.....	64
2.2.1	<i>Creación.....</i>	67
2.2.2	<i>Operación.....</i>	78
2.2.3	<i>Disolución.....</i>	79
2.2.4	<i>Evolución.....</i>	80
2.3	LOS ENTORNOS DE GESTACIÓN.....	80
2.3.1	<i>Definición.....</i>	83
2.3.2	<i>Tipos.....</i>	84
2.3.3	<i>Ventajas.....</i>	84
2.3.4	<i>Entorno de gestación versus universo abierto.....</i>	86
2.3.5	<i>El ciclo de vida.....</i>	89
2.3.6	<i>Algunos ejemplos.....</i>	91
2.4	ROLES DE LAS DVE.....	92
2.5	RESUMEN.....	96

CAPÍTULO 3 LA SELECCIÓN DE SOCIOS EN LA LITERATURA DE INVESTIGACIÓN.....99

3.1	INTRODUCCIÓN.....	99
3.2	CONSIDERACIONES INICIALES.....	101
3.2.1	<i>El enfoque computacional o asistido.....</i>	103
3.2.2	<i>El enfoque basado en los sistemas multiagente.....</i>	104
3.2.3	<i>El enfoque basado en los servicios web.....</i>	105
3.3	REVISIÓN DE PROYECTOS I+D+I RELACIONADOS.....	106
3.4	LA SELECCIÓN DE SOCIOS DESDE UN ENFOQUE CUANTITATIVO.....	113
3.4.1	<i>AHP y su aplicación al problema de selección.....</i>	115
3.5	LOS MAS Y LA SELECCIÓN DE SOCIOS EN LA DVE.....	119
3.5.1	<i>Breve reseña histórica.....</i>	120
3.5.2	<i>Repaso a la utilización de los MAS en la selección de socios.....</i>	121
3.6	EL ENFOQUE FEDERACIÓN DE SERVICIOS.....	130
3.6.1	<i>¿Qué es un servicio web?.....</i>	130
3.6.2	<i>Aplicación a la selección de socios en la DVE.....</i>	132
3.6.3	<i>Unión MAS - web services.....</i>	134
3.7	RESUMEN.....	135

CAPÍTULO 4 EL MODELO Y LA JUSTIFICACIÓN DE LA TECNOLOGÍA.....139

4.1	INTRODUCCIÓN.....	139
4.2	CONSIDERACIONES PREVIAS.....	141
4.2.1	<i>Los procesos de negocio y su descomposición.....</i>	141

4.2.2	<i>Los modelos de selección de socios en la literatura.....</i>	144
4.3	EL MODELO PARA LA SELECCIÓN DE SOCIOS.....	148
4.3.1	<i>De los modelos basados en un único enfoque.....</i>	148
4.3.2	<i>... a nuestro modelo multienfoque.....</i>	149
4.4	LOS SISTEMAS MULTIAGENTE.....	152
4.4.1	<i>El concepto de agente software.....</i>	153
4.4.2	<i>Los sistemas multiagente.....</i>	155
4.4.3	<i>Interacción entre los agentes.....</i>	157
4.4.4	<i>Utilidad de los sistemas multiagente.....</i>	158
4.4.5	<i>¿Por qué son adecuados los MAS en las DVE?.....</i>	160
4.5	LOS SISTEMAS EXPERTOS.....	162
4.5.1	<i>¿Qué es un sistema experto?.....</i>	163
4.5.2	<i>Estructura de un sistema experto.....</i>	164
4.5.3	<i>Ventajas y limitaciones.....</i>	168
4.5.4	<i>Estudio de la viabilidad de un sistema experto.....</i>	171
4.5.5	<i>Tipos de sistemas expertos.....</i>	172
4.5.6	<i>Utilidad de un sistema experto.....</i>	181
4.5.7	<i>¿Por qué son adecuados los ES en nuestra solución?.....</i>	182
4.6	RESUMEN.....	183

CAPÍTULO 5 ELECCIÓN DE LAS HERRAMIENTAS Y METODOLOGÍAS DE DESARROLLO.....185

5.1	INTRODUCCIÓN.....	185
5.2	ESTÁNDARES DE DESARROLLO MULTIAGENTE.....	187
5.2.1	<i>MASIF y CORBA.....</i>	188
5.2.2	<i>KQML y KIF.....</i>	189
5.2.3	<i>FIPA y ACL.....</i>	190
5.2.4	<i>XML.....</i>	193
5.2.5	<i>Discusión.....</i>	194
5.3	ENTORNOS DE DESARROLLO MULTIAGENTE.....	194
5.3.1	<i>AgentTool.....</i>	195
5.3.2	<i>AgentBuilder.....</i>	196
5.3.3	<i>ZEUS.....</i>	196
5.3.4	<i>JADE.....</i>	198
5.3.5	<i>Discusión.....</i>	199
5.4	METODOLOGÍAS DE DESARROLLO MULTIAGENTE.....	201
5.4.1	<i>AUML.....</i>	202
5.4.2	<i>MAS-CommonKADS.....</i>	203
5.4.3	<i>Técnica de modelado de agentes BDI.....</i>	205
5.4.4	<i>GAIA.....</i>	206
5.4.5	<i>MESSAGE e INGENIAS.....</i>	207
5.4.6	<i>Metodología para Jade.....</i>	209
5.4.7	<i>Discusión.....</i>	211

5.5	ONTOLOGÍAS.....	212
5.5.1	<i>Definición.....</i>	212
5.5.2	<i>Taxonomía y ejemplos.....</i>	213
5.5.3	<i>Desarrollo de ontologías.....</i>	214
5.6	SISTEMAS EXPERTOS.....	216
5.6.1	<i>Desarrollo de un sistema experto.....</i>	216
5.6.2	<i>Herramientas de desarrollo de sistemas expertos.....</i>	218
5.6.3	<i>Lenguajes.....</i>	220
5.6.4	<i>CLIPS.....</i>	222
5.6.5	<i>JESS.....</i>	223
5.6.6	<i>Discusión.....</i>	225
5.7	RESUMEN.....	227

CAPÍTULO 6 ANÁLISIS Y DISEÑO DE LA PLATAFORMA MULTIAGENTE.....229

6.1	INTRODUCCIÓN.....	229
6.2	BREVE DESCRIPCIÓN DE LA FUNCIONALIDAD DE LA PLATAFORMA.....	232
6.3	ANÁLISIS.....	233
6.3.1	<i>Paso 1: construcción de los casos de uso.....</i>	235
6.3.2	<i>Paso 2: identificación de los tipos de agente iniciales.....</i>	239
6.3.3	<i>Paso 3: identificación de las responsabilidades.....</i>	240
6.3.4	<i>Paso 4: identificación de las amistades.....</i>	241
6.3.5	<i>Paso 5: refinamiento de los agentes.....</i>	243
6.3.6	<i>Paso 6: información de despliegue de los agentes.....</i>	248
6.4	DISEÑO.....	249
6.4.1	<i>Paso 1: división/unión/renombrado de los agentes.....</i>	252
6.4.2	<i>Paso 2: especificación de las interacciones.....</i>	253
6.4.3	<i>Paso 3: definición ad hoc del protocolo de interacción.....</i>	254
6.4.4	<i>Paso 4: plantillas de mensajes.....</i>	255
6.4.5	<i>Paso 5: descripción a ser registrada/buscada (Yellow Pages).....</i>	256
6.4.6	<i>Paso 6: interacciones agente-recurso.....</i>	257
6.4.7	<i>Paso 7: interacciones agente-usuario.....</i>	257
6.4.8	<i>Paso 8: comportamientos internos de los agentes.....</i>	258
6.4.9	<i>Paso 9: definición de las ontologías.....</i>	259
6.4.10	<i>Paso 10: selección del lenguaje de contenido.....</i>	261
6.5	EL DOMINIO DE APLICACIÓN.....	261
6.5.1	<i>Paneles o módulos fotovoltaicos.....</i>	262
6.5.2	<i>Baterías.....</i>	266
6.5.3	<i>Reguladores.....</i>	267
6.5.4	<i>Inversores.....</i>	268
6.5.5	<i>Otros elementos.....</i>	269
6.5.6	<i>Conexión a la red.....</i>	269
6.5.7	<i>La distribución física y los operadores logísticos.....</i>	270
6.5.8	<i>La ontología de dominio.....</i>	271
6.6	RESUMEN.....	273

CAPÍTULO 7 CONSTRUCCIÓN DE LA PLATAFORMA.....275

7.1 INTRODUCCIÓN.....275

7.2 BREVE DESCRIPCIÓN DE LAS HERRAMIENTAS EMPLEADAS.....276

 7.2.1 *Una visión panorámica*.....276

 7.2.2 *JADE*.....276

 7.2.3 *La IDE NetBeans*.....282

 7.2.4 *Jess*.....284

 7.2.5 *Protégé*.....287

7.3 LA PLATAFORMA.....290

 7.3.1 *El agente administrador*.....290

 7.3.2 *El agente de acceso*.....296

 7.3.3 *El agente empresa*.....297

 7.3.4 *El agente cliente*.....313

 7.3.5 *Los agentes de servicio*.....313

 7.3.6 *Javadoc de la aplicación*.....313

7.4 RESUMEN.....315

CAPÍTULO 8 VALIDACIÓN Y VALORACIÓN FINAL.....319

8.1 INTRODUCCIÓN.....319

8.2 VALIDACIÓN.....320

 8.2.1 *Nomenclatura*.....321

 8.2.2 *Escenario 1*.....325

 8.2.3 *Escenario 2*.....326

 8.2.4 *Escenario 3*.....328

 8.2.5 *Escenario 4*.....330

 8.2.6 *Escenario 5*.....331

 8.2.7 *Escenario 6*.....332

 8.2.8 *Escenario 7*.....333

 8.2.9 *Conclusiones principales*.....333

8.3 VALORACIÓN.....335

 8.3.1 *La VE dinámica y su contexto*.....335

 8.3.2 *La tecnología empleada*.....342

 8.3.3 *Los logros conseguidos*.....349

8.4 RESUMEN.....359

CONCLUSIONES Y EXTENSIONES FUTURAS.....361

INTRODUCCIÓN.....361

CONCLUSIONES.....363

LÍNEAS DE TRABAJO FUTURO.....372

Mejoras en la plataforma.....372

Nuevas tecnologías.....376
Hacia una mayor aplicación práctica.....381

BIBLIOGRAFÍA.....385

REFERENCIAS BIBLIOGRÁFICAS.....385
RECURSOS WEB.....420

LISTA DE ACRÓNIMOS

ACC	Canal de Comunicación de Agentes (<i>Agent Communication Chanel</i>)
ACL	Lenguaje de Comunicación de Agentes (<i>Agent Communication Language</i>)
AHP	Proceso de Análisis Jerárquico (<i>Analytical Hierarchy Process</i>)
AI	Inteligencia Artificial (<i>Artificial Intelligence</i>)
AID	Identificador de Agente (<i>Agent IDetifier</i>)
AMS	Servicio de Gestión de Agentes (<i>Agent Management Service</i>)
API	Interfaz de Programación de Aplicaciones (<i>Application Programming Interface</i>)
ASIF	Asociación de la Industria Fotovoltaica
AUML	<i>Agent UML</i>
AVE	Empresa Virtual Ágil (<i>Agile Virtual Enterprise</i>)
B2B	<i>Business to Business</i>

B2C	<i>Business to Consumer</i>
BDI	<i>Belief, Desire, Intention</i>
BMO	<i>Business Management Ontology</i>
BP	Proceso de Negocio (<i>Business Processes</i>)
CAD	Diseño Asistido por Ordenador (<i>Computer Aided Design</i>)
CBR	Razonamiento Basado en Casos (<i>Case Based Reasoning</i>)
CIM	Fabricación Integrada por Ordenador (<i>Computer Integrated Manufacturing</i>)
CLIPS	Sistema de Producción Integrado en Lenguaje C (<i>C Language Integrated Production System</i>)
CN	Red Colaborativa (<i>Collaborative Network</i>)
CNE	Comisión Nacional de Energía
CO	Oportunidad de Colaboración (<i>Collaboration Opportunity</i>)
CORBA	<i>Common Object Request Broker Architecture</i>
CSVE	Empresa virtual Estática Centralizada (<i>Centralised Static VE</i>)
DAI	Inteligencia Artificial Distribuida (<i>Distributed Artificial Intelligence</i>)
DARPA	Agencia de Proyectos de Investigación Avanzada de la Defensa (<i>Defense Advanced Research Projects Agency</i>)
DBP	Proceso de Negocio Distribuido (<i>Distributed Business Process</i>)
DF	Facilitador de Directorios (<i>Directory Facilitator</i>)
DOM	Modelo de Objetos de Documento (<i>Document Object Model</i>)
DSVE	Empresa Virtual Estática Descentralizada (<i>Decentralised Static VE</i>)
DVE	Empresa Virtual Dinámica (<i>Dynamic Virtual Enterprise</i>)
EE	Empresa Extendida (<i>Extended Enterprise</i>)
EI	Institución Electrónica (<i>Electronic Institution</i>)
ES	Sistema Experto (<i>Expert System</i>)
FGKL	<i>Fifth Generation Kernel Language</i>
FIPA	Fundación para los Agentes Físicos Inteligentes (<i>Foundation for Intelligent Physical Agents</i>)
FIPA-SL	Lenguaje de Contenidos SL de FIPA (<i>FIPA - Semantic Language</i>)
FPC	<i>Función de Probabilidad Conjunta</i>

FS	Software Libre (<i>Free Software</i>)
GOPRR	<i>Graph, Object, Property, Relationship, and Role</i>
GUI	Interfaz Gráfica de Usuario (<i>Graphical User Interface</i>)
I+D+i	Investigación, Desarrollo e Innovación (<i>Investigation, Development and Innovation</i>)
ICT	Tecnologías de la Información y la Comunicación (<i>Information and Communication Technologies</i>)
IDE	Entorno de Desarrollo Integrado (<i>Integrated Development Environment</i>)
IDL	Lenguaje de Especificación de Interfaces (<i>Interface Definition Language o Interface Description Language</i>)
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos (<i>The Institute of Electrical and Electronics Engineers</i>)
IHOP	<i>Internet Inter-ORB Protocol</i>
IMS	Sistemas de Fabricación Inteligentes (<i>Intelligent Manufacturing Systems</i>)
IP	Protocolo de Interacción (<i>Interaction Protocol</i>)
ISO	Organización Internacional para la Estandarización (<i>International Organization for Standardization</i>)
IST	Tecnologías de la Sociedad de la Información (<i>Information Society Technologies</i>)
J2ME	<i>Java 2 Micro Edition</i>
J2SE	<i>Java 2 Standard Edition</i>
JADE	<i>Java Agent DEvelopment framework</i>
JDOM	Documento de Modelado de Objetos en Java (<i>Java Document Object Model</i>)
JESS	<i>Java Expert System Shell</i>
JIT	Justo a Tiempo (<i>Just in Time</i>)
JRE	Entorno de Ejecución Java (<i>Java Run-time Enviroment</i>)
JVM	Máquina Virtual de Java (<i>Java Virtual Machine</i>)
KISS	Mantenlo simple y funcionará (<i>Keep It Simple, Smart</i>) aunque posteriormente, apareció la versión modificada “Mantenlo simple, estúpido” (<i>Keep It Simple, Stupid</i>)

KQML	<i>Knowledge Queryng and Manipulation Language</i>
KSE	<i>Knowledge Sharing Effort</i>
LCM	Modelo de Ciclo de Vida (<i>Life-Cycle Model</i>)
LEAP	<i>Light Extensible Agent Platform</i>
LISP	<i>LISt Processor</i>
MAS	Sistema Multiagente (<i>Multi Agent System</i>)
MASE	<i>Multiagent Systems Engineering</i>
MASIF	<i>Mobile Agent System Interoperabilities Facility</i>
MCYT	Ministerio de Industria, Comercio y Turismo
MESSAGE	<i>Methodology for Engineering Systems of Software Agents</i>
MIT	Instituto de Tecnología de Massachusetts (<i>Massachusetts Institute of Technology</i>)
MTS	Sistema de Transporte de Mensajes (<i>Message Transport System</i>)
NASA	Administración Nacional de Aeronáutica y del Espacio (<i>National Aeronautics and Space Administration</i>)
NBS	Oficina Nacional de Estándares (<i>National Bureau of Standards</i>) siglas con las que se conocía al Instituto Nacional de Estándares y Tecnología (NIST) entre 1901 y 1988
NIIP	<i>Nacional Industrial Information Infrastructure Protocols</i>
NIST	Instituto Nacional de Estándares y Tecnología (<i>National Institute of Standards and Technology</i>)
NO	Organización en Red (<i>Networked Organization</i>)
OMG	<i>Object Management Group</i>
OOP	Programación Orientada a Objetos (<i>Object-oriented programming</i>)
OOPL	Lenguaje de Programación Orientada a Objetos (<i>Object-Oriented Programming Language</i>)
OPSS	<i>Official Production System 5</i>
ORB	<i>Object Request Bus</i>
P2P	<i>Peer-to-Peer</i>
PDA	Asistente Digital Personal (<i>Personal Digital Assistant</i>)
PROLOG	<i>PROgrammation avec LOGique</i>

RMA	Gestión Remota de Agentes (<i>Remote Agent Management</i>)
RMI	Invocación de Métodos Remotos (<i>Remote Method Invocation</i>)
SAX	<i>Simple API for XML</i>
SCM	Gestión de la Cadena de Suministro (<i>Supply Chain Management</i>)
SDK	Kit de Desarrollo de Software (<i>Software Development Kit</i>)
SGML	<i>Standard Generalized Markup Language</i>
SME	Pequeñas y Medianas Empresas (<i>Small and Medium Enterprises</i>)
SOAP	<i>Simple Object Access Protocol</i>
SVE	Empresa Virtual Estática (<i>Static Virtual Enterprises</i>)
TOVE	<i>TOronto Virtual Enterprise</i>
UDDI	<i>Universal Description, Discovery, and Integration</i>
UML	Lenguaje Unificado de Modelado (<i>Unified Modeling Language</i>)
VAS	Servicio de Valor Añadido (<i>Value Added Service</i>)
VBE	Entornos de Gestación de VE (<i>VE Breeding Environment</i>)
VE	Empresa Virtual (<i>Virtual Enterprise</i>)
VO	Organización Virtual (<i>Virtual Organization</i>)
W3C	<i>World Wide Web Consortium</i>
WSDL	Lenguaje de Descripción de Servicios Web (<i>Web Services Description Language</i>)
XML	Lenguaje de Marcado Extensible (<i>Extensible Markup Language</i>)

INTRODUCCIÓN

ANTECEDENTES

Las últimas décadas del siglo XX han sido testigos de una expansión de los negocios sin precedentes en la historia de la humanidad. Este proceso de continuo crecimiento y transformación tiene su origen en un desafío común a todos los seres vivos: la lucha por la supervivencia. En el caso particular de las empresas, esta lucha supone la obligación de satisfacer permanentemente los deseos de los clientes, cada vez más exigentes, mejorando la eficiencia productiva y adaptándose continuamente a los cambios de un entorno global, competitivo y dinámico.

Para lograr este objetivo las empresas han ido realineando sus prioridades competitivas a lo largo del tiempo. Si durante los años 70 el énfasis se puso en la reducción de los costes y en el aumento de la eficacia, en la década de los 80 la prioridad fue la calidad, y en los 90 la velocidad de entrega se convirtió en el imperativo principal. Los supervivientes en este proceso de evolución y selección natural emergieron con fuerzas de trabajo más pequeñas, estructuras organizacionales más planas y operaciones generalmente más eficaces (Petersen, 2003).

Las innovaciones en el ámbito de las tecnologías de la información y comunicación (ICT, *Information and Communication Technologies*), tales como el desarrollo de arquitecturas abiertas o la introducción de Internet en el mundo empresarial, se convirtieron en soporte y combustible del mayor incremento de competencia y globalización que ha conocido la humanidad. De hecho, las ICT todavía están sosteniendo, e incluso acelerando, este cambio durante los primeros años del nuevo milenio, en los que la flexibilidad y, muy especialmente, la agilidad se han convertido en las prioridades competitivas de las modernas empresas (Hales y Barker, 2003).

Por lo que se refiere a los modelos organizativos, fue a finales de los 80 cuando un gran número de investigadores empezaron a observar que las megacorporaciones no siempre eran capaces de superar a sus competidores más pequeños y especializados: su elevada diversificación e integración vertical les privaban de la flexibilidad y agilidad necesarias para seguir siendo competitivas en un mercado global. Enseguida empezaron a buscarse modelos de empresa alternativos, en los que la externalización de actividades a diferentes niveles desempeñaba un papel decisivo. Es lo que se conoce como “externalización de lo interno” (Sorensen, 1996).

El desarrollo y uso de las ICT ha creado un marco favorable para la creación de estas formas organizativas, ya que ofrecen innovadoras e interesantes posibilidades a las políticas de externalización. Como resultado, durante las últimas décadas hemos asistido a la introducción de numerosos conceptos, estructuras y paradigmas organizativos que tratan de dar respuesta a las exigencias de los procesos de dirección y organización en unos entornos cada vez más complejos (Sanz *et al.*, 2004). El paradigma de empresa virtual (VE, *Virtual Enterprise*), considerado por numerosos autores como la estrategia organizativa del siglo XXI, destaca de entre todas ellas.

Para comprender el éxito de la VE debemos tener presente que en un entorno de negocio tan hipercompetitivo como el actual, una empresa no puede permitirse el lujo de ser siempre una “segundona”. Sin embargo, y desde un punto de vista realista, tampoco puede esperar ser mejor que el resto de los competidores en todos sus procesos de negocio. Las VE proporcionan la posibilidad de unir varias organizaciones que son las mejores en su *Core Business* para configurar una organización que sí es la mejor en todo. Representa, por tanto, el medio más eficiente para aunar los requisitos de excelencia, dinamismo y agilidad que toda organización debe poseer si quiere sobrevivir en un entorno de negocio tan dinámico y global como el actual.

En la literatura de investigación consultada no existe una definición académica estricta del concepto de empresa virtual, aunque las diferentes definiciones y modelos presentes revelan características técnicas y de negocio comunes. Entre ellas podemos destacar las siguientes:

- Se tiende a describirla como un tipo de red orientada a satisfacer una necesidad de mercado específica suministrando al cliente un producto o servicio en cuya elaboración están involucradas varias entidades organizativas heterogéneas, autónomas y legalmente independientes.
- Cada entidad aporta al conjunto sus procesos esenciales o *Core Business*. En otras palabras, asigna al consorcio sus mejores habilidades, capacidades, recursos y experiencia, lo que asegura la creación de ventajas competitivas y, por tanto, de rentas superiores¹.
- Estas entidades también comparten infraestructuras, investigación, desarrollo e innovación (I+D+i, *Investigation, Development and Innovation*), riesgos, costes, beneficios, mercados, clientes,...
- Su correcto funcionamiento exige que las entidades involucradas actúen como una única empresa, por lo que han de definir y coordinar sus objetivos de forma complementaria. En este sentido, el éxito sólo será posible cuando se satisfagan simultáneamente las necesidades del mercado y las de los propios participantes, y cuando exista un alto grado de confianza entre las partes involucradas, para lo que es necesario una comunicación fluida.
- Las relaciones de negocio entre los diferentes agentes tienen lugar durante un periodo de tiempo limitado, que puede ir desde la provisión de un único producto o servicio hasta la ejecución de un conjunto de procesos de negocio definido a largo plazo.
- La forma en que una entidad emplea un recurso o se vale de un proceso de otra, puede ser estática (predefinida y estable) o bien dinámica (evolucionando con el tiempo).
- El número de socios de la VE también puede ser estático o dinámico, dependiendo tanto de la oportunidad de negocio como de las necesidades y requisitos de los socios involucrados.
- Los socios se encuentran geográficamente distribuidos y están conectados a través de sistemas electrónicos. Las ICT son elementos clave de esta cooperación.

¹ La excelencia que se logra con la especialización constituye un aspecto crucial en el éxito de la VE, dado que se pretende crear una empresa que sea *best of everything*.

- El cliente reconoce a la VE como una única entidad puesto que obtiene el producto o servicio de un modo transparente, sin llegar a conocer su estructura, sus componentes, etc.

En base a estas características podemos identificar dos categorías bien definidas de empresa virtual: las estáticas (*SVE, Static Virtual Enterprises*) y las VE dinámicas (*DVE, Dynamic Virtual Enterprises*).

- Las VE estáticas están formadas por un conjunto de entidades que se unen de un modo fijo, dando lugar a relaciones de negocio perfectamente integradas, personalizadas y, por lo general, de larga duración. La red también es invariable, configurando una estructura estática y predeterminada.
- En las VE dinámicas, también conocidas como empresas virtuales ágiles (Goranson, 1999), un conjunto de socios de negocio se unen dinámicamente, bajo demanda y en base a las necesidades de los clientes. Las entidades involucradas no poseen relaciones de negocio fijas por lo que la empresa virtual puede cambiar continuamente dependiendo de criterios orientados al mercado.

De los dos arquetipos de VE, son precisamente los dinámicos los que han despertado mayores expectativas en diferentes dominios de aplicación como consecuencia de sus múltiples beneficios potenciales: acceso más rápido a nuevos mercados y a nuevas oportunidades de negocio; se superan desafíos, se alcanzan metas de negocio, se tiene acceso a recursos (habilidades, *know-how*,...), etc., que normalmente están fuera de las posibilidades de un único participante; se incrementa la utilización de los activos; se mejora el servicio al cliente y la calidad del producto o servicio; se disminuyen y comporten riesgos, costes, etc.; se logran economías de escala; las pequeñas y medianas empresas logran presencia internacional; etc.

El simple hecho de que ante la aparición de una oportunidad de negocio pueda configurarse rápidamente una empresa virtual dinámica/ágil hecha a su medida, es muestra de una flexibilidad y agilidad envidiables. La agilidad es precisamente, y en un contexto tan turbulento como el actual, el elemento clave para asegurar la supervivencia de las empresas. Por ese motivo, en la nueva y abierta economía nacida de Internet son preferibles las VE dinámicas y ágiles capaces de aprovecharse de las condiciones variables del mercado (Ouzounis, 2001) ofreciendo aquellos productos que demandan los clientes en el momento justo, con la máxima calidad y un coste óptimo.

Sin embargo, y aunque desde el punto de vista de los negocios las VE dinámicas son modelos más prometedores, desde un punto de vista técnico requieren soluciones y sistemas técnicos mucho más complejos, sofisticados y distribuidos (Alzaga y Martin, 1999). Estas dificultades técnicas son las principales responsables de que la implantación práctica de este concepto se encuentre todavía lejos de ser una realidad a gran escala. En otras palabras, si bien la agilidad potencial que otorga una rápida reacción a las oportunidades es, ciertamente, una característica deseable en un escenario donde las condiciones del mercado cambian tan rápidamente, la creación de VE dinámicas presenta todavía obstáculos que deben superarse para lograr que dicha agilidad sea una realidad.

Durante las dos últimas décadas, y motivados por las elevadas expectativas asociadas a este tipo de empresas, han surgido un gran número de proyectos de I+D+i en los que se intentan ofrecer estas soluciones, estableciendo tanto las bases tecnológicas como las prácticas operativas necesarias para soportar las VE dinámicas. Este esfuerzo es particularmente visible en Europa, a través de los múltiples programas financiados por la Comisión Europea, pero también en EEUU, Japón, China, etc. Incluso algunos programas, como es el caso de IMS (*Intelligent Manufacturing Systems*) (IMS web, 2006), soportan proyectos que involucran organizaciones de varios continentes.

Como resultado de las actividades de muchos de estos proyectos, se ha constatado la necesidad de investigar con mayor profundidad todos los aspectos relacionados con la creación eficaz de las empresas virtuales dinámicas. Dentro de esa etapa de creación, la identificación y selección de los socios más adecuados en cada momento desempeña un papel crucial. Tanto es así, que numerosos autores (Mikhailov, 2002; Lau y Wong, 2001; Ouzounis, 2001; Reid *et al.*, 1996; Goldman, 1995; Camarinha-Matos y Afsarmanesh, 2001; etc.) la han calificado como el elemento clave en el éxito de este tipo de empresas.

Dada la trascendencia que posee el proceso de selección en el desarrollo práctico de este modelo organizativo, no resulta nada extraño la gran proliferación de trabajos y enfoques alternativos que han surgido en los últimos años alrededor de este tema. La presente tesis doctoral pretende unirse a esta investigación proporcionando un modelo innovador y realista con el que poder afrontar de forma eficiente la selección de socios en las VE dinámicas. Obviamente, este objetivo sólo será posible tras un análisis profundo del problema.

MOTIVACIÓN

A medida que aumenta la presión competitiva sobre las compañías, éstas ven crecer de un modo significativo sus necesidades de innovación, eficiencia, flexibilidad,... y, sobre todo, rapidez en la respuesta. Estas exigencias son, como hemos visto, consecuencia directa de una necesidad de nivel superior: garantizar su supervivencia. En un contexto como el actual, las empresas han de colaborar las unas con las otras para lograr este objetivo, lo que hace necesario no sólo encontrar el conjunto de socios que mejor se adapte a una oportunidad de negocio, y constituir así una red colaborativa que la explote, sino hacerlo en el menor tiempo posible.

El simple hecho de que entidades legalmente independientes sean capaces de unirse para configurar una organización supra-empresarial orientada a la consecución de una meta u objetivo, pone de manifiesto la enorme agilidad asociada a la noción de empresa virtual dinámica. Sin embargo, encontrar los socios adecuados y establecer las condiciones necesarias para iniciar el proceso de colaboración puede llegar a consumir mucho tiempo, esfuerzo y dinero convirtiéndose, de este modo, en un proceso inhibitorio de la tan anhelada agilidad.

Entre los obstáculos más relevantes se pueden citar: la ausencia de información (sobre los posibles socios, sus procesos de negocio, su infraestructura tecnológica,...), la falta de preparación de las organizaciones para unirse y trabajar constituyendo redes cooperativas, así como los problemas asociados a la heterogeneidad de los socios potenciales (diferentes infraestructuras, cultura corporativa, métodos de trabajo, prácticas de negocio, etc.). No obstante, el obstáculo por excelencia es la falta de confianza, un requisito previo para cualquier colaboración efectiva.

Además, la selección de los socios no puede limitarse a un simple problema de “optimización” (enfoque tradicional), donde solamente se asignan los procesos en función del potencial y de las habilidades de los posibles candidatos. Se han de tener presentes otros muchos factores entre los que se incluyen algunos de naturaleza subjetiva como son las preferencias personales y la confianza resultante del conjunto de experiencias previas. Este hecho sugiere que la utilización de procesos totalmente automatizados no constituyen un enfoque plenamente realista. Es preferible, por tanto, concebir un marco informático que asista al planificador humano en la toma de decisiones (Camarinha-Matos, 2003).

En resumen, nos encontramos ante un problema decisivo en el éxito de las empresas virtuales dinámicas aunque de difícil resolución dada su elevada complejidad. Es, por este motivo, que a lo largo de los últimos años han surgido diferentes enfoques que han dado lugar a una gran variedad de soluciones. De entre todos ellos destaca el enfoque basado en los sistemas multiagente (*MAS, Multi Agent System*). De hecho, existe un gran número de características en el dominio de las empresas virtuales dinámicas que hacen de estos modelos organizativos un área de aplicación adecuado para los sistemas multiagente.

A título ilustrativo, los MAS son adecuados para dominios que requieren una constante adaptación a entornos distribuidos y/o a demandas cambiantes; también abordan cuestiones como la coordinación y resolución de problemas distribuidos, ambos aspectos críticos en la gestión de las DVE; la fase de formación muestra características de mercado y necesidades de negociación que han sido cuestiones ampliamente estudiadas en los MAS; la reconfiguración de las DVE necesita un paradigma de modelado flexible que pueda adaptarse rápidamente a las circunstancias; la propiedad de escalabilidad de los MAS parece particularmente adecuada para soportar DVE en las que se pueden establecer diferentes niveles de cooperación; etc.

Estas y otras razones son las que han propiciado que a lo largo de los últimos años hayan surgido algunas plataformas software que emplean la tecnología multiagente y los mecanismos de negociación orientados al mercado para llevar a cabo la selección de socios en las empresas virtuales. No obstante, los sistemas desarrollados presentan un grado de automatización elevado, sobre todo en los procesos de resolución, lo que impide reflejar plenamente la realidad de las empresas y de sus mecanismos de toma de decisiones. En definitiva, son sistemas que no conducen a una solución plenamente realista y satisfactoria del problema.

Para poder suministrar una solución adecuada al problema conviene detenerse un instante a examinar cómo toman sus decisiones las empresas, sobre todo aquellas relacionadas con el establecimiento de relaciones con otras entidades. Así, podemos apreciar que en casi todas las empresas, y sobre todo en las de pequeño y mediano tamaño, las decisiones no se toman en base a enrevesadas funciones matemáticas ni a complejas hipótesis estadísticas; más bien, los directivos de estas empresas suelen recurrir a su “experiencia”, “conocimiento” e “intuición”, relegando la información cuantitativa recopilada para ese instante a un segundo plano.

Por tanto, es preciso desarrollar un sistema capaz de reflejar y hacer uso de conceptos como confianza, eficiencia, compromiso, experiencia previa, preferencias, etc., sin olvidar todos aquellos aspectos relativos a la optimización “clásica” como son los costes o el emparejamiento habilidades-requisitos. Sólo de esta forma estaremos en disposición de ofrecer una respuesta plenamente realista al problema planteado. En definitiva, necesitamos un sistema capaz de pensar y actuar tal y como lo haría cualquier planificador humano, es decir, necesitamos dotar al sistema multiagente de un módulo de decisión que simule el comportamiento de los decisores humanos.

Pero, ¿cómo podemos representar el “conocimiento”, la “experiencia” y la “intuición” que posee un decisor en una aplicación informática?. Está claro que para poder hacerlo necesitamos un sistema que “piense” y que “razone” como lo haría un experto humano; necesitamos, en definitiva, un sistema experto (*ES, Expert System*). Los sistemas expertos son sistemas informáticos que simulan el proceso de aprendizaje, memorización, razonamiento, comunicación y, en consecuencia, de acción de un experto humano en cualquier rama de la ciencia. Estas características le permiten almacenar datos y conocimiento, sacar conclusiones lógicas, tomar decisiones, comunicarse con expertos humanos, explicar el porqué de las decisiones tomadas y realizar acciones como consecuencia de todo lo anterior.

Todas estas consideraciones permiten vislumbrar claramente nuestra motivación. No obstante, y antes de concretarla, quizá resulte ilustrativo establecer un símil entre los conceptos de ciclo de vida de una DVE y ciclo de vida de un producto. Desde un punto de vista clásico, éste último suele estar constituido por cinco etapas: I+D, lanzamiento, crecimiento, madurez y declive. Uno de los objetivos que persigue cualquier empresa en relación a sus productos no es otro que el de reducir la duración de las etapas iniciales; de este modo, la empresa puede centrarse rápidamente en aquellas etapas de mayor rentabilidad, es decir, las etapas de crecimiento y madurez. Al mismo tiempo, esta forma de actuar permite lanzar el producto más rápidamente al mercado, lo que se convierte en una ventaja competitiva.

Siguiendo un razonamiento similar, a nosotros nos interesa reducir la duración de las etapas iniciales del ciclo vital de la DVE y, en particular, la duración de la fase de formación. Sólo de esta forma podremos centrarnos lo antes posible en la operación de la VE dinámica con el objeto de explotar, de una manera más rentable, rápida y eficaz, la oportunidad de negocio

especifica que ha surgido, antes de que ésta desaparezca o pueda ser aprovechada por otra entidad. Para ello, nos centraremos en el proceso de selección de socios, proporcionando un modelo operativo que sirva de base en la construcción de herramientas de información eficientes que asistan a los responsables humanos en sus procesos de decisión.

OBJETIVOS

La presente tesis surge con la ya declarada intención de *construir un modelo operativo, innovador y realista del proceso de selección de socios en las DVE que guíe la creación de herramientas de soporte con las que dar una respuesta eficiente y satisfactoria a dicho problema*. Para verificar la idoneidad de este modelo será preciso desarrollar una plataforma software con la que poder experimentar y, en definitiva, extraer conclusiones.

Con el fin de alcanzar el objetivo planteado se proponen una serie de pasos que han de delimitar el camino a seguir. Éstos se enumeran a continuación de forma esquemática:

- Confección de un “estado del arte” sobre la VE dinámica y la problemática asociada al proceso de selección de socios, estudiando los diferentes enfoques planteados para su resolución.
- Elaboración de un modelo realista y funcional del proceso de selección en las DVE. El objetivo de todo modelo es proporcionar una representación simplificada de la realidad; revelar, en definitiva, la esencia de lo que está pasando. En nuestro caso, pretendemos crear un modelo que englobe el amplio abanico de alternativas que pueden darse en la formación de las DVE y que, al mismo tiempo, sea lo más “simple”² posible. Por ello, y siguiendo el conocido principio KISS (*Keep It Simple, Smart*), deberemos esforzarnos en lograr reducir nuestro modelo justamente a aquellas piezas que se requieren para hacerlo trabajar, rechazando lo enrevesado e innecesario.
- Justificación de la tecnología más adecuada para la construcción del sistema software con el que verificar la idoneidad del modelo. También será necesario, por tanto, seleccionar las especificaciones, metodologías y herramientas que mejor se ajusten a nuestros propósitos de entre aquellas que tengan una mayor aceptación.

2 Einstein dijo en una ocasión “Todo debe ser lo mas simple posible, pero no más”.

- Análisis, diseño y construcción (utilizando para ello las herramientas y las metodologías seleccionadas) de una plataforma software que ofrezca una solución innovadora y eficiente al problema de la selección de socios en las VE dinámicas.
- Planteamiento y construcción de diferentes escenarios dentro del dominio de aplicación. El análisis de los resultados obtenidos nos permitirá obtener conclusiones sobre la adecuación del modelo propuesto y de la tecnología empleada.

Pero, ¿por qué centrarnos exclusivamente en el modelo dinámico de VE cuando los resultados de esta investigación pueden ser plenamente válidos para los distintos modelos organizacionales propuestos hasta la fecha?. La respuesta se halla en la propia naturaleza de la DVE. En efecto, este modelo requiere soluciones tecnológicas avanzadas para poder dar respuesta a las continuas exigencias de cambio y adaptación que puedan surgir durante su existencia (nuevas oportunidades de negocio, nuevos socios, cambios en el desempeño de los integrantes, etc.). Es, por tanto, en este contexto donde los modelos y sistemas como el que aquí propondremos tienen un significado pleno, frente al resto de los casos donde estarían claramente infrutilizados.

Por último, para lograr un desarrollo idóneo de la plataforma necesitaremos hacer uso de una amplia variedad de herramientas software que faciliten las distintas tareas a realizar. En otras palabras, la edición, codificación, verificación, generación, análisis, gestión, etc., de los diferentes documentos a elaborar precisa de suites ofimáticas, lenguajes de programación, entornos integrados de desarrollo, editores de ontologías, etc. Esta necesidad conduce a fijar un principio que trataremos de seguir en el desarrollo del proyecto: hacer uso de Software Libre (FS, *Free Software*) o, en su caso, software con licencia académica para investigación, lo que constituirá nuestro pequeño grano de arena en la construcción de un mundo con mayores libertades.

Desde que en 1983 Richard M. Stallman diera a conocer el proyecto GNU, que a la postre se convertiría en el primer movimiento visible para difundir el Software Libre, el número de aplicaciones desarrolladas bajo las distintas licencias de Software Libre ha crecido de forma vertiginosa. Aunque una de las ventajas de su utilización se halla en que la mayor parte del FS es gratuito, su verdadero poder reside en la libertad de distribución y modificación de los programas; esta realidad explica la total transparencia del FS y posibilita que sus usuarios puedan aprender de él, mejorarlo, e incluso utilizarlo en sus proyectos.

El crecimiento de este “gran movimiento global” no sería posible sin el apoyo desinteresado de un extenso conjunto de programadores que trabajan por todo el mundo colaborando para mejorar este tipo de programas de acuerdo con las necesidades reales de los usuarios. Gracias a ellos, en la actualidad la mayor parte del software libre no tiene nada que envidiar al software propietario. Sirvan estas líneas como medio para reflejar nuestra gratitud y reconocimiento por su trabajo.

ORGANIZACIÓN DEL DOCUMENTO³

El presente documento está constituido por 8 capítulos que se pueden englobar, desde un punto de vista funcional, en dos grandes bloques. El primero de ellos, al que pertenecen los tres capítulos iniciales, pretende establecer el marco teórico asociado a la empresa virtual y, en especial, a la selección de socios en las VE dinámicas. El segundo bloque, por su parte, se centra en el modelo operativo del proceso de selección y en el desarrollo de la plataforma software que servirá para verificar la adecuación del modelo a la realidad del proceso de selección. A continuación, pasamos a describir brevemente los contenidos de cada uno de los diferentes capítulos.

El *capítulo 1* introduce el concepto de empresa virtual. Partiendo de una breve descripción del contexto empresarial que ha dado lugar a la necesidad de modelos organizacionales flexibles, adaptativos y de costes variables como el que representa la empresa virtual, este capítulo detalla las características más relevantes de este paradigma. Al mismo tiempo, también propone una serie de variables con las que clasificar los diferentes tipos de empresa virtual existentes, de entre los que destaca muy significativamente la VE dinámica. Finalmente se discuten algunas de las teorías que pretenden describir cómo debe evolucionar una empresa para llegar a formar parte de este revolucionario modelo.

La identificación y selección de socios en la DVE es un tema complejo al que se ha tratado de dar respuesta desde muy diferentes orientaciones. Parte de esta complejidad se deriva de la falta de un marco conceptual común que

3 Algunos de los resultados del presente trabajo han sido incluidos en el proyecto de investigación titulado “Integración empresarial y gestión de la cadena de suministro basada en sistemas multiagente”, con referencia DPI2004-06590 y financiado por el MCYT (Ministerio de Industria, Comercio y Turismo).

constituya el punto de partida para un análisis acertado del problema. Partiendo de esta reflexión, el *capítulo 2* nace con el simple propósito de establecer ese punto de partida. Por ello, en este capítulo se describe el modelo de ciclo de vida de la DVE prestando especial atención a aquellas etapas más interesantes desde el punto de vista de la selección de socios (creación y evolución). También se esbozan algunas de las características más importantes de los entornos de gestación de VE, por considerarlos el contexto natural del proceso de creación de estas organizaciones. El capítulo finaliza con la descripción de los principales roles que pueden aparecer durante la existencia de una VE dinámica, incluyendo los más representativos de los entornos de gestación.

El primer bloque funcional finaliza con el *capítulo 3*, en el que se muestran los aspectos más relevantes de la investigación desarrollada hasta el momento en torno al proceso de identificación y selección de socios en la DVE. Con dicho propósito, en este capítulo se analizan algunos de los proyectos I+D+i más importantes sobre la VE y se presentan los trabajos más representativos asociados a tres formas distintas de enfocar el problema: análisis cuantitativo, utilización de la tecnología multiagente y aplicación del paradigma de los servicios web. De este análisis se descubre que el enfoque multiagente destaca claramente como la mejor alternativa, al menos en la actualidad, para afrontar el problema de la selección de socios en este tipo de consorcios. No obstante, y pensando en un futuro de investigación próximo, también se considera la posibilidad de unir las tecnologías *web services* y multiagente.

Con el *capítulo 4* se inaugura el bloque más práctico de la presente memoria. Parte de la definición de un modelo simple, realista y funcional del proceso de selección, que deja paso a la justificación de la tecnología software que emplearemos en la construcción de la aplicación. Con respecto al modelo, su principal característica reside en el uso de una perspectiva multienfoque que permite aunar los beneficios de las plataformas existentes en la literatura y, más importante aún, nos da la posibilidad de construir cualquier tipo de VE dinámica de forma fácil y sencilla. La justificación se lleva a cabo para los dos tipos de tecnologías que se han considerado más adecuadas a nuestros propósitos: la tecnología multiagente y los sistemas expertos. Por ello, se exponen sus conceptos fundamentales y se argumentan las razones que aconsejan la utilización de cada una de ellas⁴ para afrontar el problema de la selección de socios.

4 Constituye la primera parte del proceso de planificación de la aplicación.

El *capítulo 5* prosigue con la planificación de la plataforma multiagente a través de la elección de las herramientas y metodologías de desarrollo que emplearemos en la construcción de la aplicación software. En concreto, se presentan algunas de las características de los principales estándares, plataformas y metodologías de desarrollo multiagente existentes, ofreciendo diferentes comparativas que guiarán nuestra decisión final. También se aborda el tema de las ontologías dada su creciente relevancia y el papel clave que poseen en la construcción de aplicaciones multiagente, incluida la nuestra. Por último, se afronta la elección de la herramienta que facilitará la construcción del sistema experto; esta elección se realizará en base al estudio de los principales sistemas de desarrollo existentes.

Tras la planificación les llega el turno al análisis y al diseño de la plataforma de agentes, y para ello haremos uso de las guías suministradas por Nikraz *et al.* (Nikraz *et al.*, 2006). Así, el *capítulo 6* comienza con el análisis de la plataforma, donde se busca clarificar el problema identificando los requisitos del dominio, y a continuación se aborda el diseño, es decir, la especificación de la solución del problema. Es importante tener presente que en la metodología adoptada la fase de análisis es genérica e independiente de la plataforma mientras que la fase de diseño asume específicamente que Jade es la plataforma de desarrollo y se centra directamente en las clases y los conceptos proporcionados por dicho entorno. En este capítulo, y como parte de la fase de diseño, también se describen los conceptos fundamentales de la ontología del dominio que emplearemos en la validación del modelo; en concreto, nos estamos refiriendo a la instalación de módulos fotovoltaicos, un proceso de negocio cada vez más en auge.

El *capítulo 7* está dedicado por completo a la construcción de la aplicación multiagente. Por ello, y con el propósito de favorecer una mejor comprensión de la solución desarrollada, comienza con una breve descripción de las herramientas empleadas en dicho proceso: la plataforma de desarrollo multiagente Jade, las IDE Netbeans y Eclipse, el sistema experto Jess, la herramienta de desarrollo de ontologías Protégé, etc., así como varios *plugins* y *add-ons* disponibles para estas herramientas. A este breve resumen le sigue la descripción de la plataforma desarrollada. Tal descripción se realiza prestando una especial atención a sus usuarios, por lo que principalmente se consideran las interfaces de los agentes y la forma en que deben interactuar los usuarios. No obstante, también se abordan algunos de los aspectos más relevantes de su implementación allí donde se ha considerado necesario.

Una vez construida la aplicación multiagente, llega el momento de validar tanto el modelo de partida como la plataforma. Dicha validación se realiza a lo largo de la primera parte del *capítulo 8*, utilizando, para ello, el proceso de negocio ficticio descrito en el capítulo 6. A través de la definición de varios escenarios se comprueba el correcto funcionamiento de la plataforma, lo que conduce a establecer tanto la adecuación de las tecnologías empleadas en su construcción como la validez del modelo que constituía la base de su desarrollo. En este capítulo también se valoran todos los elementos clave que han intervenido en la memoria, desde la empresa virtual o los entornos de gestación hasta el trabajo realizado, pasando por las tecnologías empleadas en su construcción.

La memoria finaliza con un análisis de los principales resultados obtenidos y planteando posibles líneas de actuación que continúen la labor desarrollada en este tiempo. Por lo que se refiere a las conclusiones, éstas se dividen de acuerdo con la estructura definida en los objetivos del presente documento: el marco teórico, el modelo de negocio, la planificación, análisis, diseño y construcción de la plataforma software y su posterior validación. En cuanto a las posibles extensiones futuras hacemos mención tanto a las mejoras de la plataforma como a las distintas alternativas de actuación que surgen, tanto tecnológicas como prácticas.

CAPÍTULO

1

EL PARADIGMA DE EMPRESA VIRTUAL

1.1 INTRODUCCIÓN

Si en algo nos parecemos los diferentes seres vivos que compartimos este planeta es, sin el menor atisbo de duda, en que todos perseguimos un mismo objetivo vital: la supervivencia. Desde el león más fiero hasta la bacteria más pequeña, tienen grabado en su código genético la necesidad de sobrevivir, no sólo haciendo frente a posibles depredadores sino también, como pusieron de manifiesto Charles Darwin y Alfred Wallace, compitiendo con otros individuos de su misma especie.

En ese mismo sentido, las empresas también son entes vivos ya que los principios de “selección natural” y “lucha por la supervivencia” forman parte de su esencia. Conocer esta realidad permite comprender mejor su historia y, en particular, el porqué de la búsqueda permanentemente de nuevas formas y modelos organizativos como mecanismo de supervivencia que les permita afrontar con éxito el continuo crecimiento de la presión competitiva.

En la actualidad, el éxito de las empresas depende de su capacidad para acortar los ciclos de desarrollo y fabricación, reducir los tiempos de acceso a los mercados o *time to market*, optimizar los costes operacionales, actuar a escala global, adaptarse rápidamente a los cambios del mercado, etc., buscando, en todo momento, satisfacer a unos clientes cada vez más exigentes que demandan productos y servicios de más alta calidad (Sanz, 2004). Las empresas que ignoren estos desafíos están destinadas a sucumbir ante sus rivales (Giménez y Lourenço, 2004) perdiendo, así, la batalla por la supervivencia.

Durante las últimas décadas la respuesta de las empresas a estas necesidades ha venido de la mano de la automatización, tanto de los procesos como de las operaciones de negocio (sistemas de fabricación flexibles, fabricación integrada por ordenador, sistemas Just in Time, etc.) y de la gestión de la empresa (sistemas ERP). Sin embargo, en un mercado tan competitivo, globalizado y dinámico como el actual esta respuesta por sí sola ya no es suficiente (Sanz, 2004). Crear valor añadido para los clientes se ha convertido en un proceso cada vez más complejo que exige la combinación de diferentes clases de conocimiento que las compañías no necesariamente poseen (Beer *et al.*, 1990) ni les es posible adquirir.

Ante esta situación, ¿qué se puede hacer?. Si acudimos de nuevo a la naturaleza observaremos que en ella no todo es competición y que a menudo la cooperación es más eficiente que la competencia. La profesora Lynn Margulis, una de las personas que han revolucionado la biología contemporánea, explica en su teoría de la simbiogénesis esta realidad. Según ella, los animales, las plantas, los hongos,... , hemos surgido por evolución a partir de la asociación y la cooperación entre bacterias (Margulis y Sagan, 1996). En otras palabras, la vida es bacteriana y en la evolución no todo es competición, sino que hay mucha simbiosis (Ramoneda, 2006).

Si definimos cooperación como el proceso en el que se comparten competencias para alcanzar una meta común (Picard, 2005), advertiremos que tal proceso también ha estado presente a lo largo de la historia del hombre, desde sus orígenes hasta nuestros días. A través de la unión de sus competencias, el ser humano ha demostrado ser capaz de alcanzar grandes hitos y superar retos antes inimaginables: la construcción de las pirámides egipcias, la gran muralla china, las grandes infraestructuras ferroviarias estadounidenses, etc., constituyen algunos de los ejemplos más significativos que evidencian esta colaboración.

En definitiva, si las empresas quieren asegurar su supervivencia, y alcanzar metas cada vez más ambiciosas, deben aprender a cooperar, a veces incluso con los competidores directos (Sanz, 2004; Byrne *et al.*, 1993; Davidow y Malone, 1998). Esta cooperación va a permitir que sus miembros puedan abarcar proyectos más complejos que de otra forma no serían factibles por diferentes motivos como, por ejemplo, una capacidad de inversión limitada, el desconocimiento de la tecnología apropiada para su producción, la falta del tiempo necesario para su desarrollo, el excesivo riesgo, etc.

Pero la cooperación no sólo supone una forma de compartir riesgos ante nuevos proyectos o ante la entrada a nuevos mercados, sino que también es un prerequisite para una mayor innovación, una mayor capacidad de resolución de problemas y una mayor *performance* (Gisela y Korsunsky, 2003). Representa, en definitiva, una vía de acceso a los mercados globales no sólo para las grandes empresas sino, y muy especialmente, para las pequeñas y medianas ya que les permite adquirir la apariencia de las grandes, y su atractivo a los ojos de los consumidores, sin ver incrementados sus costes.

Sin embargo, los sistemas de información de las empresas, desde las pequeñas a las grandes, presentan una gran diversidad y heterogeneidad, lo que constituye un impedimento a la hora de establecer una cooperación eficaz (Ouzounis, 2001). Los recientes avances en telecomunicaciones, redes abiertas como Internet, lenguajes de programación independientes de la plataforma como Java, metalenguajes flexibles como XML, etc., han abierto un abanico de nuevas oportunidades para que los negocios se gestionen electrónicamente tanto a nivel intra como inter-organizacional.

Pero Internet, junto con los nuevos avances tecnológicos, no sólo permite una mejor colaboración, más rápida y fiable, sino que al mismo tiempo abre nuevos canales comerciales a las empresas (Giménez y Lourenço, 2004). Éstas se ven obligadas a detectar y aprovechar las oportunidades que ofrece la nueva economía digital, o *e-Business*, para poder adquirir una posición estratégica que les permita sobrevivir en un mundo tan globalizado, dinámico e interconectado como el actual.

Para lograrlo, las compañías deben cooperar en las diferentes fases del desarrollo del producto y compartir entre ellas los procesos de negocio críticos, los recursos, competencias nucleares, habilidades y su *know-how* (Applegate *et al.*, 1996; Goldman *et al.*, 1995). Este nuevo modelo de

negocio ha derivado en el concepto de Empresa Virtual (VE, *Virtual Enterprise*), considerada por numerosos investigadores como la estrategia organizativa del siglo XXI (Davidow y Malone, 1992; Byrne *et al.*, 1993; Fischer *et al.*, 1996; Mowshowitz, 1997; Park y Favrel, 1999; Camarinha-Matos *et al.*, 2001; etc.).

La empresa virtual permite que un número de organizaciones, instituciones o individuos legalmente independientes y geográficamente dispersas, desarrollen un entorno de cooperación y trabajo común orientado a la consecución de una meta concreta. Este entorno posibilita la fabricación de productos y/o la prestación de servicios de mayor calidad y adaptados a las necesidades del mercado, pero incurriendo en un menor coste, una distribución del riesgo y una reducción del *time to market*, lo que se traduce en una mejor respuesta a las exigencias de los clientes.

Cada miembro de la VE lleva sus competencias nucleares relevantes para la misión y se concentra en aquellas áreas donde puede aportar una ventaja competitiva única (Mikhailov, 2002), lo que permite generar márgenes más altos (Katzy y Miralles, 1999). Por tanto, dado que cada socio aporta su fortaleza o *core business*⁵ a la empresa, el éxito del proyecto depende de que todos cooperen como una unidad (Sanz *et al.*, 2003). En otras palabras, la ventaja competitiva que pueda alcanzar la VE dependerá del grado en que las organizaciones individuales se complementen entre sí y de su habilidad para integrarse las unas con las otras. Se busca, en definitiva, crear una organización que sea la mejor en todo (Adams *et al.*, 2001).

El concepto de empresa virtual permite aunar las ventajas de las grandes empresas (con gran poder financiero y un gran número de recursos, pero no lo suficientemente reactivas), y las de las pequeñas empresas (más débiles y sin tantos recursos pero más fáciles de dirigir y coordinar, y con una mayor capacidad de reacción y adaptación a los cambios) (Gornev *et al.*, 1997; Do *et al.*, 2000). Las pequeñas y medianas empresas que participan en una empresa virtual tienen acceso a los recursos de las grandes organizaciones mientras retienen la agilidad y la independencia propias de las pequeñas (Browne y Zhang, 1999).

5 A principios de los 90, Prahalad y Hamel (Prahalad y Hamel, 1990) definieron las competencias nucleares como el “conocimiento colectivo de la organización”, es decir, “cómo coordinan las diferentes técnicas de producción e integran las múltiples tecnologías”.

Los participantes comparten recursos, costes, infraestructura, personal, investigación, conocimiento, *know-how*, habilidades y competencias, lo que les permite acceder de manera colectiva a mercados globales con soluciones de clase mundial que no podrían proporcionar individualmente (Park y Favrel, 1999; Browne y Zhang, 1999). Tal hecho puede hacer que las empresas pequeñas aparenten, se sientan y actúen como si fuesen grandes, alcanzando a clientes antes fuera de su radio de acción, mientras que las grandes compañías pueden llegar a sentirse pequeñas a la hora de atender a esos clientes. En otras palabras, la VE permite ampliar el rango de alcance de las organizaciones sin aumentar sus costes fijos (Adams *et al.*, 2001).

La empresa virtual es, en definitiva, un modelo corporativo y de cooperación que responde a las necesidades del mercado actual mejor que otras estructuras, aportando ventajas superiores no sólo respecto a las organizaciones independientes o a los modelos de cooperación tradicionales (adquisiciones, fusiones, *joint ventures*,...) (Dang, 2005), sino también respecto a otros modelos de cooperación actuales. Este hecho se debe, entre otras razones, a su facilidad de adaptación, pero sobre todo a la utilización intensiva de las tecnologías de la información: durante todas las fases del ciclo de vida de la VE, la tecnología de información juega un papel decisivo en la integración de los socios de negocio (Kanet *et al.*, 1999).

Nos encontramos, por tanto, ante la necesidad de conocer los aspectos fundamentales de este nuevo paradigma: necesitamos familiarizarnos con este nuevo concepto y su definición, aprender a distinguirlo de otros conceptos existentes en la literatura, ser capaces de reconocer los diferentes tipos de empresas virtuales así como sus características principales y, finalmente, descubrir el proceso necesario para llegar a formar parte de este prometedor modelo organizacional. Conocer y entender todas estas peculiaridades constituye, precisamente, el objetivo de este capítulo.

Pero para llegar a comprender el paradigma de empresa virtual en su conjunto, primero necesitamos conocer los cambios acaecidos a lo largo de los últimos años y, por tanto, el nuevo contexto en el que desarrollan su actividad las organizaciones empresariales de principios del siglo XXI. A fin de cuentas, son estos cambios y este nuevo entorno los que han dado lugar a la necesidad de modelos organizacionales flexibles, adaptativos y de costes variables, como el que representa la empresa virtual, capaces de sobrevivir al asumir que la cooperación es un requisito imprescindible para alcanzar la excelencia.

1.2 EL NUEVO ENTORNO EMPRESARIAL

Desde el punto de vista político e histórico, acontecimientos como las sucesivas ampliaciones de la Unión Europea, los tratados de libre comercio, etc., han engendrado mercados más amplios y liberalizados, abriendo nuevas y prometedoras oportunidades a aquellas empresas que sean capaces de adaptarse (Cuesta, 1998). Los mercados se configuran a nivel supranacional, lo que para las empresas se traduce en un aumento de la competencia, pero también en la posibilidad de alcanzar economías de escala si logran unir los diferentes microsegmentos internacionales en los que participan.

Este escenario condiciona claramente el nuevo entorno económico, cuya principal característica es la destrucción de empleo en el primer mundo. De hecho, a lo largo de los últimos años la creación de puestos de trabajo se ha ido desplazando paulatinamente hacia los países menos desarrollados, principalmente al continente asiático, donde algunos factores productivos (en especial la mano de obra) suelen ser significativamente más atractivos para muchas empresas. Esta circunstancia está en el origen del fenómeno conocido como deslocalización de empresas.

La deslocalización, junto con los cambios producidos en las empresas por las reducciones de costes, fusiones, adquisiciones, etc., provoca incertidumbre e inseguridad en torno al puesto de trabajo. Como consecuencia, el concepto de empleo está empezando a ser cuestionado en los nuevos modelos empresariales y está siendo sustituido por el de empleabilidad, ineludiblemente vinculado a la inseguridad del concepto de trabajo por proyecto (Cuesta, 1998).

A nivel comercial, la principal característica es el estancamiento de los mercados tradicionales. Estamos hablando de mercados muy maduros, generalmente vinculados a países desarrollados, con un alto grado de saturación y, por tanto, con enormes dificultades para crecer. Esto provoca que las oportunidades de negocio actuales se presenten en momentos puntuales, en regiones geográficas diferentes y durante periodos de tiempo cada vez más breves (Katzy y Dissel, 2001).

Esta madurez de los mercados también tiene sus efectos sobre el comportamiento de los consumidores: las empresas deben hacer frente a clientes cada vez más exigentes que demandan de los productos aquellas expectativas que las empresas han despertado en ellos (Sanz *et al.*, 2003).

Para competir en este nuevo entorno se hace imprescindible conocer las necesidades de los clientes, sus aficiones, tendencias, etc., de forma que se llegue a confeccionar una oferta totalmente personalizada.

El resultado es una evolución en la concepción del producto. Frente a una visión técnica (enfoque del producto centrado en sí mismo) que considera al producto como una suma de características y atributos físicos, se ha impuesto un enfoque centrado en las necesidades del consumidor. Desde este nuevo punto de vista, las personas compran los productos no por sí mismos, sino por los problemas que resuelven (Santesmases, 1993). En otras palabras, los consumidores no compran productos, adquieren la esperanza de ver satisfechas sus necesidades.

Todos estos cambios conducen a uno de los retos empresariales actuales más importantes: la necesidad de buscar una solución de compromiso entre el concepto de globalización y la realidad de la microsegmentación de los mercados locales (Cuesta, 1998). La empresa necesita internacionalizar su oferta en busca de economías de escala sin olvidar que los consumidores poseen características y necesidades particulares.

Tampoco podemos dejar pasar por alto que el intervalo de tiempo en el que un producto puede ser comercializado con éxito está disminuyendo dramáticamente (Fischer *et al.*, 1996), convirtiéndose el *time-to-market* en un elemento decisivo dentro de las empresas. En consecuencia, otra característica que debemos exigir es la rapidez en la respuesta: el producto y/o servicio que se preste ha de satisfacer las necesidades y exigencias de los clientes de la manera más rápida posible.

Pero, ¿cómo vamos a ser capaces de aunar ambos requisitos?. La respuesta se halla en el conocimiento permanente de los clientes y en la anticipación, que no son posibles si no se dispone de un sistema de comunicación adecuado. Una comunicación fluida con los clientes, y una actuación posterior en consonancia, serán la mejor garantía para ofrecer a los clientes lo que realmente desean con unos costes optimizados y en el menor tiempo posible (Sanz *et al.*, 2004a).

Esta fuente de oportunidades de negocio que es la comunicación entre los diferentes agentes, incluida la comunicación con los clientes, se ha visto favorecida y facilitada por el impresionante desarrollo de las nuevas tecnologías de la información y las comunicaciones, principal característica

del nuevo entorno tecnológico (Cuesta, 1998). En un contexto caracterizado por la turbulencia y el cambio, la tecnología se ha convertido en uno de los elementos fundamentales para alcanzar y mantener la competitividad (Pérez, 1996).

Cuando la Unión Europea se fijó en marzo de 2000, en Lisboa, el ambicioso objetivo de convertirse en la mayor “economía competitiva y dinámica basada en el conocimiento” (COM, 2001) del mundo en los diez años siguientes, reconocía que alcanzar este objetivo dependía de saber utilizar las tecnologías de la información y la comunicación (ICT, *Information and Communication Technologies*) de la mejor manera posible.

Hoy en día resulta evidente que la convergencia tecnológica ha supuesto más de un cambio en la manera de gestionar los negocios, ha convertido en obsoletos los supuestos económicos fundamentales sobre los que se sustentaban la mayoría de empresas, y ha transformado la economía industrial en lo que hoy denominamos economía digital (Accenture, 2003).

Además, el aumento acelerado de la productividad y, en consecuencia, de la competitividad de las empresas y del crecimiento económico en su conjunto, está estrechamente relacionado con los avances en las ICT. Estos avances se han caracterizado tanto por el descenso de los precios como, sobre todo, por el aumento de la capacidad y la facilidad de uso, lo que ha supuesto una mejora en la calidad de los productos y los servicios.

Pero la importancia que poseen las ICT en la economía va más allá. De hecho, facilitan la adopción de innovaciones y reformas en la organización empresarial y, además, modifican las condiciones de competencia y la configuración de los mercados, causando un gran impacto sobre la estructura y el ciclo vital de las empresas (COM, 2001):

- Reducen las consecuencias económicas de la distancia y los costes de acceso a la información, propiciando una mayor competencia en los mercados.
- Tienden a disminuir el coste de creación de las pequeñas empresas, lo que puede dar lugar a una competencia adicional.
- Permiten que la entrega de productos/servicios se realice con nuevos medios cooperativos, aumentando la calidad y la rentabilidad.
- Finalmente, y tal vez lo más importante, hacen posible el desarrollo de nuevos productos y servicios.

Otro aspecto a considerar dentro del sector tecnológico es el movimiento físico de los materiales y productos. La mayoría de los nuevos modelos empresariales dependen de una logística eficaz; ésta desempeña un papel fundamental para lograr un aumento de la eficacia de las cadenas de suministro y para la creación de nuevas formas de comercio en colaboración. Debemos conseguir que este movimiento físico se produzca de manera eficiente y rentable, alcanzando economías de escala y dentro del nuevo orden global (Cuesta, 1998).

En el terreno de los recursos humanos también se ha venido produciendo un cambio significativo a lo largo de los últimos años. A medida que la competitividad de las empresas iba dependiendo cada vez más de su capacidad de respuesta a las condiciones cambiantes de su entorno y de su disposición a la mejora continua, las organizaciones han ido elevando su valoración de los recursos humanos (Pérez, 1996).

Fue la mentalidad de la producción en masa la que supuso que la revolución electrónica y la robotización terminarían expulsando al ser humano del proceso productivo. Hoy, sin embargo, se hace cada vez más evidente que los recursos humanos son el activo más importante que posee una empresa (Stewart, 1997) para competir con éxito. Las personas, con su conocimiento, se han convertido en auténticos elementos diferenciadores de las organizaciones en detrimento de otros tipos de recursos, entre ellos la tecnología, que pueden ser adquiridos o reemplazados fácilmente al estar inmersos en un mercado globalizado (Cuesta, 1998).

La empresa moderna debe ser un tipo de organización flexible y adaptable, y para ello precisa integrar profesionales, especialmente ingenieros, con un alto nivel de formación, conocimientos actualizados, capacidades para la toma de decisiones y autodirección. Sin embargo, la formación de estos recursos humanos conlleva tiempo y dinero. Para paliar estas carencias temporales de recursos humanos las empresas se ven obligadas a recurrir a su entorno (expertos internacionales, apoyo de socios externos, consultoras, etc.), así como a la maximización del uso de la capacidad propia, independientemente del lugar en el que se encuentre.

Por lo que se refiere a la competencia, en la era de la información se espera que ésta tenga lugar cada vez menos entre compañías individuales, para dejar paso a una competencia entre agrupaciones de compañías (Laubacher y Malone, 1997) que trabajen juntas para explotar el valor de una oportunidad

de negocio (Katzy y Dissel, 2001). La razón es simple: las competencias que son necesarias para acceder a las nuevas oportunidades, así como las inversiones y el tiempo requeridos para su construcción, exceden normalmente los recursos propios de una única entidad (Sanz *et al.*, 2004a).

Para responder de un modo eficiente a todos estos cambios y desafíos, la empresa debe optimizar sus estructuras de costes (alcanzando así ventajas competitivas), especializar sus unidades productivas y lograr la consecución de economías de escala. Y para ello, además de contar con la siempre necesaria involucración total de la Dirección, la empresa del futuro tiene que (Cuesta, 1998; Accenture, 2004; Sanz, 2004):

- *Reemplazar el concepto organizativo de integración vertical por el de proceso horizontal*, reduciendo así los niveles jerárquicos de la empresa. En la economía industrial tradicional, los fuertes costes de interacción y transformación provocaron altos niveles de integración vertical⁶. En la economía digital, las tecnologías IC aconsejan una desintegración vertical: aconsejan montar una organización virtual.
- *Contemplar a la empresa como un sistema abierto* (cadena de valor de Porter) *y global, en el que se integran proveedores, clientes, etc., e incluso competidores*. Las empresas, después de “virtualizar” sus procesos internos básicos, deben abrirse a los proveedores (y a los proveedores de sus proveedores) y a los clientes (y a los clientes de sus clientes), participando en iniciativas orientadas a satisfacer a estos últimos. Mediante acuerdos con socios tecnológicamente punteros e innovadores, especializados en determinados eslabones de la cadena de valor, se pueden alcanzar nuevos niveles de calidad, flexibilidad y ahorro antes inimaginables.

Las tendencias económicas actuales exigen un movimiento de las compañías integradas verticalmente hacia organizaciones en red más flexibles, donde la habilidad para establecer, desarrollar y disolver rápida y eficientemente asociaciones con otros socios es un factor de éxito crítico (Osterle *et al.*, 2000). Una respuesta estructural a estos retos la hallamos en el concepto de empresa virtual, considerada por numerosos investigadores como la estrategia organizativa del siglo XXI (Davidow y Malone, 1992; Byrne *et al.*, 1993; Fischer *et al.*, 1996; Mowshowitz, 1997; Park y Favrel, 1999).

6 Para las empresas resultaba más fácil y económico poseer muchos eslabones de la cadena de valor, desde las materias primas hasta la entrega de los productos finales.

1.3 DEFINICIÓN

La VE no es un concepto nuevo en los estudios de gestión (Malone y Rockart, 1991; Camarinha-Matos y Afsarmanesh, 1999; Hales y Barker, 2003). Muchas de las grandes compañías, como por ejemplo los fabricantes de automóviles, acostumbran a mantener relaciones de negocio remotas o “virtuales” con sus suministradores e incluso con los clientes corporativos. Sin embargo, el nivel de integración y las tecnologías de información y comunicación aplicadas son, a menudo, inapropiados (Ouzounis, 2001). Precisamente, una de las características de la VE es el uso de las nuevas ICT para soportar la red de información (Gisela y Korsunsky, 2003).

La empresa virtual constituye un área de investigación y desarrollo tecnológico muy destacada para las principales industrias de las economías desarrolladas. Sin embargo, y a pesar de ser un campo de investigación multidisciplinar y en crecimiento, carece todavía de una definición precisa de los conceptos y de un acuerdo sobre la terminología a utilizar (Camarinha-Matos y Afsarmanesh, 2001). Después de más de una década de investigación en torno a la misma, muchos de los aspectos relacionados con su formación, operación, mantenimiento y disolución permanecen bastante difusos.

Además, los múltiples desafíos socioeconómicos a los que se enfrentan las empresas actuales (y la sociedad en general) han propiciado la aparición, en los últimos años, de un gran número de formas organizativas orientadas a dar respuesta a los nuevos retos. Fruto de esta proliferación, términos como “empresa extendida”, “organización virtual”, “redes cooperativas”, “organizaciones en red”, “gestión de la cadena de suministro”, “cluster de empresas”, etc., compiten en la literatura de investigación volviéndola aún más confusa.

En este contexto de indefinición no resulta extraño que un gran número de investigadores lleguen a emplear estos términos como sinónimos, aunque en realidad muestren características diferentes relativas tanto a su duración, propósito y/o cultura. Por tanto, y antes de poder adentrarnos de lleno en el universo de la empresa virtual, conviene definir cada una de estas formas organizacionales de manera que seamos capaces de ubicar correctamente el concepto de empresa virtual y, de este modo, diferenciarlo claramente del resto de términos “presuntamente equivalentes”.

1.3.1 Conceptos relacionados

El concepto de Empresa Extendida (EE, *Extended Enterprise*) es el rival más cercano al concepto de empresa virtual. Se aplica a una organización en la que una empresa dominante “extiende” sus límites de negocio a todos o algunos de sus proveedores (Browne y Zhang, 1999), mientras que la Empresa Virtual se puede ver como un concepto más general que incluye otros tipos de organizaciones como, por ejemplo, una estructura más democrática en la que la cooperación es de igual a igual, *peer-to-peer* (Camarinha-Matos y Afsarmanesh, 2001). En este sentido, la empresa extendida puede ser considerada como un caso particular de empresa virtual.

El concepto de Organización Virtual (VO, *Virtual Organization*) es similar al de VE, comprendiendo una red de organizaciones que comparten recursos y habilidades para alcanzar una misión u objetivo, aunque no se ve limitada a una alianza de empresas (Doz y Hamel, 1998). Un ejemplo de organización virtual podría ser un municipio virtual que asocie, a través de redes informáticas, todas las organizaciones del municipio como, por ejemplo, los servicios de distribución del agua, servicios de ocio, servicios catastrales, etc. (Camarinha-Matos y Afsarmanesh, 2001). Una VE es, por tanto, un caso particular de organización virtual (Lau y Wong, 2001; Doz y Hamel, 1998).

Tanto las empresas extendidas como las empresas y las organizaciones virtuales constituyen ejemplos de redes colaborativas (CN, *Collaborative Network*). Una red colaborativa está constituida por una variedad de entidades (organizaciones e individuos) autónomas, geográficamente distribuidas y heterogéneas en términos de sus entornos operativos, cultura, capital social y metas (Camarinha-Matos *et al.*, 2005c). A pesar de sus diferencias, estas entidades colaboran⁷ para alcanzar una serie de metas comunes (y compatibles).

El término Organización en Red (NO, *Networked Organization*) es, posiblemente, el concepto más general para referirnos a cualquier grupo de organizaciones interconectadas por una red de ordenadores, pero sin compartir necesariamente habilidades, recursos, procesos, o tener una meta común (NIIP, 2003). Las NO se corresponden con un tipo de organización muy abierta (Ouzounis, 2001) que engloba a los modelos anteriores.

7 Los miembros de este tipo de redes comparten la creencia de que juntos pueden alcanzar metas que no serían posibles si se intentasen de manera individual.

En la figura 1.1 se muestra la relación entre los diferentes tipos de organizaciones presentadas hasta el momento.

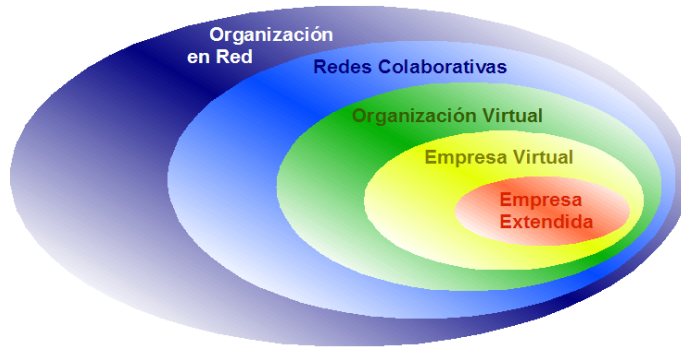


Figura 1.1. Descripción visual de los nexos existentes entre las diversas formas de organización en red.

Como se puede observar, la Organización en Red representa el concepto más amplio de organización interconectada, en el que la cooperación no es una característica definitoria. Sí lo es, no obstante, en las Redes Colaborativas, dentro de las cuales encontramos las Organizaciones Virtuales, donde dicha colaboración implica a diferentes tipos de entidades jurídicas; si estas alianzas se dan principalmente entre empresas tenemos las Empresas Virtuales. Finalmente, cuando las relaciones están marcadas por la existencia de una entidad dominante hablamos de Empresas Extendidas.

Por lo que se refiere a la Gestión de la Cadena de Suministro (SCM, *Supply Chain Management*), en 1998 The Global Supply Chain Forum la definió como “la integración de los procesos de negocio clave desde los suministradores iniciales al usuario final que proporciona productos, servicios e información que añaden valor al consumidor y otros *stakeholders*” (Lambert *et al.*, 1998).

Es, por tanto, el término utilizado para describir el conjunto de procesos de producción y logística cuyo objetivo final es la entrega de un producto a un cliente. Se trata de una integración de tipo vertical en la que están involucrados suministradores de materia prima, consumidores, fabricantes del producto, distribuidores, vendedores, etc., frente a la integración horizontal entre iguales que habitualmente incluye la empresa virtual (Hao *et al.*, 2005).

Internet y las nuevas tecnologías han revolucionado el campo de la gestión de la cadena de suministro (mejora de los procesos de aprovisionamiento, conocimiento en tiempo real de los *stocks* para saber anticipar los pedidos a proveedores, etc.), engendrando lo que se ha dado a conocer como e-SCM, término empleado para reflejar el impacto que tiene Internet en la SCM (Giménez y Lourenço, 2004).

La figura 1.2 permite ejemplificar gráficamente la diferencia entre los conceptos de cadena de suministro, empresa extendida y empresa virtual.

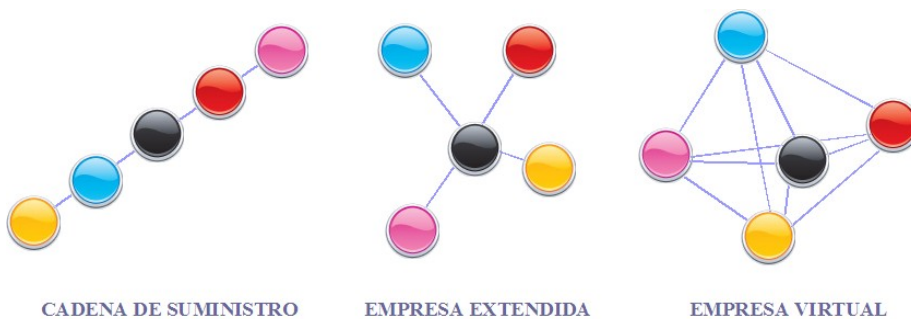


Figura 1.2. Comparación entre la SC, EE y VE. (Katzy y Loeh, 2003).

Finalmente, encontramos los *Clusters* de Empresas. El término *cluster* fue acuñado por Porter en su estudio sobre la ventaja competitiva de las naciones (Porter, 1990), en el que consideraba que la agrupación de empresas y su correspondiente especialización en determinadas actividades productivas contribuía favorablemente a la consecución de ventajas competitivas. Por tanto, se trata de asociaciones de pequeñas y medianas empresas complementarias que se interrelacionan en un territorio definido para aumentar su competitividad.

Los clusters se extienden verticalmente en la cadena de valor, incluyendo a proveedores e industrias auxiliares, y horizontalmente, hasta la tecnología y sectores relacionados. Muchos, además, suelen incluir instituciones públicas, educativas (universidades, centros especializados de formación,...), parques tecnológicos y servicios de información, reciclaje y apoyo técnico (Mixteo *et al.*, 2003). Una ventaja significativa de los clusters es que sus miembros poseen una cultura de negocio común, un sentimiento de comunidad y, de manera habitual, una infraestructura interoperativa común.

Por todas estas características, que son difíciles de conseguir en relaciones temporales, los cluster han constituido la base para la definición de los Entornos de Gestación de VE (VBE, *VE Breeding Environment*). Se trata de asociaciones de organizaciones que tienen tanto el potencial como el interés para cooperar entre sí a través del establecimiento de acuerdos de cooperación a largo plazo (Afsarmanesh, 2006). Su principal propósito consiste en preparar a sus miembros para aprovechar las oportunidades de negocio que surjan, mediante la creación eficiente de VE.

1.3.2 Definiciones de empresa virtual

Una vez aclaradas las diferencias entre algunos de los conceptos que erróneamente se utilizan para hacer referencia a la VE, es el momento de definir *qué es* realmente la empresa virtual. Esta tarea, en apariencia sencilla, ha sido el objeto de discusión, principalmente en Europa, de un gran número de investigadores durante los últimos años. Entre las causas principales de esta falta de precisión que rodea a este modelo podemos destacar las diferencias existentes en cuanto al significado de la palabra “virtual” y los diferentes tipos de VE operativos (Hales y Barker, 2000).

La palabra virtual tiene muchos significados y los investigadores tienden a verla desde su propia comprensión de la realidad (Hales y Barker, 2003). En nuestra vida cotidiana empleamos el término virtual para referirnos a algo que tiene existencia aparente y no real. Desde una perspectiva tecnológica, la palabra virtual permite describir una forma de memoria que puede ampliarse cuando es necesario (Mowshowitz, 1997), pero también se usa para describir aplicaciones que utilizan imágenes generadas por ordenador, impresiones del mundo real (realidad virtual), etc. Finalmente, en los últimos años ha crecido su uso para describir la presencia en Internet.

Por lo que se refiere a la tipología, las empresas virtuales existen con una gran variedad de formas, y en entornos muy variados. Así, pueden ser comunidades en Internet compartiendo intereses comunes, grupos o individuos de una misma empresa (intra-empresa) ubicados en diferentes lugares geográficos pero que comporten una meta común, entidades inter-organizacionales con sistemas de soporte distribuidos independientes, u organizaciones independientes que se unen para acometer un proyecto particular.

Si a todo esto le añadimos la dificultad propia de tratar un concepto de empresa que por definición está sujeto a un cambio y a una adaptación constante, podemos comprender porqué aun no existe una definición unificada para este paradigma. Así, en la literatura consultada durante la realización de la presente tesis se han encontrado un sinfín de definiciones procedentes de diferentes autores y comunidades de investigadores. Dado que mostrar y comentar cada una de estas definiciones supondría un trabajo considerable, a la par que innecesario para los propósitos de este trabajo, a continuación se presentan solamente algunas de las más relevantes⁸:

“... es un consorcio temporal o alianza de compañías formada para compartir costes y destrezas, y para explotar oportunidades de negocio rápidamente cambiantes.”
(NIIP, 1996).

“... es una red temporal de compañías independientes (proveedores, clientes e incluso competidores) unidos por tecnología de información para compartir costes, habilidades y el acceso a los mercados de los otros. No tendrán una oficina central ni un diagrama organizacional. Tampoco tendrán una jerarquía ni integración vertical.”
(Byrne *et al.*, 1993).

“... alianza temporal de empresas que se unen para compartir sus habilidades, sus competencias nucleares y sus recursos con el fin de atender una oportunidad de negocio, y cuya cooperación está soportada por redes informáticas y por las herramientas y protocolos IT apropiados.” (Camarinha-Matos y Afsarmanesh, 1999 y 2001; Camarinha-Matos, 2002; Afsarmanesh, 2005; PRODNET II web, 2005).

“... es una forma de organización en la que un conjunto de empresas, instituciones y/o personas legalmente independientes se unen rápidamente para cooperar en una misión concreta. Existe una comprensión compartida de los objetivos de negocio y cada miembro lleva a dicha cooperación sus competencias nucleares relevantes para la misión. Normalmente no se establece una burocracia de gestión formal, y debido a la necesidad de rapidez, debe

8 En (Sanz, 2004) puede encontrarse una amplia explicación y discusión sobre estas y otras definiciones.

existir un elevado grado de confianza entre los miembros. Éstos actúan para los elementos externos como una única corporación. Las ICT son los facilitadores de esta cooperación.” (Fischer *et al.*, 1996; Kanet *et al.*, 1999).

“... es un escenario que emerge en un mundo donde entidades individuales (personas, agentes software o empresas de negocio) se unen como un equipo para alcanzar un fin específico. Estas entidades individuales, los socios de la empresa virtual, cooperan para alcanzar un conjunto de metas y una vez lo han conseguido, vuelven a formar otra VE. Las empresas virtuales no tienen un marco organizacional permanente y rígido. Más bien, son un equipo de socios que tienen metas comunes y que se comprometen a conseguir esas metas.” (Petersen *et al.*, 2002).

“... es una red de diferentes dominios de negocio que cooperan compartiendo procesos de negocio y recursos para proporcionar un servicio de valor añadido al cliente. Cada miembro de la empresa virtual aporta principalmente sus competencias clave, es decir, sus recursos y procesos de negocio fundamentales. La corporación virtual existe durante un periodo de tiempo determinado por la consecución de su propósito de negocio. Desde el punto de vista de un observador externo, es decir, un cliente, la empresa virtual aparece como una única empresa.” (Ouzounis, 2001).

“... es una empresa orientada a una meta operando bajo meta-gestión.” (Mowshowitz, 2000).

A pesar de la gran divergencia de ideas y opiniones en torno al concepto de empresa virtual, sí se aprecia la existencia de un cierto grado de consenso en cuanto a algunas de sus características. A continuación se muestran, de forma resumida, algunos de sus rasgos más significativos:

- Claramente, impera una tendencia a describirla como un tipo de red que se orienta a la satisfacción de una necesidad de mercado específica proporcionando al cliente un producto y/o servicio en cuya elaboración están involucradas varias entidades organizativas heterogéneas, autónomas y legalmente independientes.

- Cada una de las entidades aporta al conjunto sus procesos esenciales o *core business*. En otras palabras, cada entidad aporta lo mejor de sus habilidades, capacidades, recursos y experiencia, lo que asegura la consecución de ventajas competitivas y la creación de rentas superiores. La excelencia es un aspecto crucial de este modelo, dado que se pretende crear una empresa que sea la mejor en todo lo que hace.
- Las entidades no sólo ponen a disposición de la empresa virtual sus *core business*, sino que también comparten infraestructuras, I+D+i, riesgos, costes, beneficios, mercados, clientes, etc.
- Su correcto funcionamiento exige que las entidades involucradas en esta cooperación actúen como una única empresa (propiciando la aparición de beneficiosos efectos sinérgicos), por lo que han de definir y coordinar sus objetivos de forma complementaria. En este sentido, el éxito del consorcio sólo será posible cuando se satisfagan simultáneamente las necesidades del mercado y las de los propios participantes.
- Esta cooperación sólo es posible si existe un alto grado de confianza entre las partes involucradas.
- Las relaciones de negocio entre los diferentes agentes tienen lugar durante un periodo de tiempo limitado, que puede ir desde la provisión de un único servicio hasta la ejecución de un conjunto de procesos de negocio definido a largo plazo.
- La forma en que esos procesos de negocio se acoplan entre los diferentes dominios de negocio, es decir, la forma en que una entidad emplea un recurso o se vale de un proceso de otra, puede ser bien estática (lo que implica que ha sido definida previamente y que goza de estabilidad en el tiempo) o bien dinámica (cambiando en función de las circunstancias y los miembros).
- El número de socios que componen la empresa virtual también puede ser estático o dinámico, dependiendo de la oportunidad de negocio y de las necesidades y requisitos de los socios involucrados en su operación.
- Los socios se encuentran geográficamente distribuidos y están conectados a través de sistemas electrónicos. Las ICT ocupan un lugar central en este nuevo modelo al ser elementos facilitadores de la cooperación.
- El cliente reconoce a la empresa virtual como una única entidad de la que obtiene el producto o servicio de un modo completamente transparente.

1.4 CRITERIOS DE CLASIFICACIÓN

Del estudio de las características anteriores y de las observaciones efectuadas por varios autores (Doz y Hamel, 1998; Ouzounis, 2001; Camarinha-Matos y Afsarmanesh, 2001; Mowshowitz, 2000; Hales y Barker, 2000; Gisela y Korsunsky, 2003; Adams *et al.*, 2001), podemos llegar a concretar algunos de los criterios de selección clave que se usarán para clasificar los modelos y conceptos de empresa virtual existentes. A continuación se proponen los más representativos:

- *Uso de la meta-gestión.* Este concepto, asociado a la definición proporcionada por Mowshowitz⁹, describe tanto la actitud como el grado de flexibilidad de una empresa, y de su gestión, en la selección de *satisfiers* que cumplan los requisitos de la organización, así como la posibilidad de sustituirlos por mejores alternativas. Esta idea tiene importantes repercusiones en otros conceptos como la topología de la red, la coordinación, la integración, etc.
- *Comportamiento intraorganizacional vs. interorganizacional.* ¿La empresa virtual trasciende los límites de la corporación o es simplemente una única compañía aplicando ciertos conceptos dentro de sus fronteras?. La virtualidad no supone necesariamente operar a nivel interorganizacional (análisis externo), traspasando y difuminando los límites de la empresa, sino que también puede considerarse como una forma de organización en sentido interno (intraorganización). Este es el caso, por ejemplo, de los equipos virtuales (*virtual teams*) en donde se asignan los individuos según su *know-how* y su experiencia en la función requerida.
- *Complementariedad de los core business de los participantes.* Tal y como describieron Hamel y Prahalad, toda empresa posee un conjunto de competencias clave, que son la razón central de su existencia, y una colección de competencias no nucleares, que son las encargadas de sustentar la empresa (Hamel y Prahalad, 1994). Aunque la subcontratación de los procesos no nucleares es una práctica de negocio común, en la VE los participantes no sólo pueden subcontratar los procesos menos importantes, sino también aquellos procesos esenciales donde no poseen una experiencia o unos conocimientos técnicos significativos (Hales y Barker, 2000). En este caso solamente se conservan aquellos procesos y competencias que

⁹ Se puede encontrar una explicación de esta definición en (Sanz, 2004).

proporcionan una ventaja competitiva sostenible y que son difíciles de copiar por parte de los competidores (Boudreau *et al.*, 1998). Así, por ejemplo, una empresa puede poseer como competencias significativas la relación con los clientes, la innovación en los procesos y productos o la infraestructura, pero su verdadera fortaleza residirá en sólo una de ellas, su *core business*, y no en todas (Hagel y Singer, 1999). Sin embargo, la VE no se queda solamente en la subcontratación de procesos sino que va más allá al permitir el acceso de los participantes a las fortalezas de los otros. La combinación de competencias distintivas complementarias procedentes de diferentes entidades es la esencia misma de las empresas virtuales ya que permite generar sinergias y satisfacer las condiciones del mercado de una manera flexible (Sanz, 2004), con el mínimo de inversión en la propia plantilla, capital fijo y capital de explotación (Hales y Barker, 2000). Es lo que se conoce como “internalización de lo externo” (Sorensen, 1996). De estas palabras se desprende que el concepto de empresa virtual está ineludiblemente vinculado al pensamiento sistémico: aunque cada una de las partes individuales que forman la VE tienen un valor relativo de forma aislada, sólo cuando estas mismas partes actúan de forma conjunta (como una única empresa) adquieren un valor que es el que catapulta a la empresa virtual a la consecución de los fines que se ha marcado. Es decir, emerge el efecto sinérgico inherente a la VE en la que la unidad, el conjunto, es superior a la suma de sus partes (Sanz *et al.*, 2003), o como reflejan diferentes autores (Adams *et al.*, 2001; Lipnack y Stamps, 1993), la VE es una estructura “uno más uno igual a tres”. Obviamente, esto requiere un alto grado de confianza y la comprensión por parte de todos los miembros de la existencia de un destino común.

- *Topología de la red.* Existen situaciones que muestran una naturaleza variable y dinámica, en la que algunas empresas pueden unirse o abandonar la alianza dinámicamente dependiendo de las fases del proceso de negocio o de otros factores del mercado. Pero también podemos encontrar relaciones con una estructura casi fija, en las que durante el ciclo de vida de la VE sólo se dan pequeñas variaciones en términos de suministradores o clientes. Esta característica condiciona el número y tamaño de los integrantes de la corporación, el grado de evolución y escalabilidad de la red y las posibilidades de interacción con socios potenciales.

- ✓ *Número y tamaño de los agentes.* Dado que el nivel de virtualidad se refleja claramente en la dificultad para identificar los agentes participantes y sus aportaciones, una red tendrá un mayor carácter virtual cuanto menor sea el tamaño medio de sus agentes y mayor sea su número (Pelechano *et al.*, 2005). En el primer caso, cuando la red tiene un carácter dinámico, el número de socios también es dinámico y flexible, cambiando de acuerdo con la evolución de la empresa virtual. Además, suele involucrar entidades de pequeño y mediano tamaño. En el segundo caso, cuando la topología de la red presenta un carácter estático, el número de socios es estable y predeterminado, involucrando, por lo general, a una o más entidades de gran tamaño.
- ✓ *Grado de evolución y escalabilidad.* Indica si la empresa virtual puede evolucionar en términos de nuevos miembros y relaciones, y si el modelo de negocio es escalable, es decir, puede expandirse y/o contraerse. El modelo dinámico de VE es, por definición, escalable y permite la transformación de la red, mientras que en el modelo estático la escalabilidad suele estar más restringida.
- ✓ *Interacción temporal con socios exteriores.* Otra posibilidad a tener en cuenta es la interacción temporal con otros socios que no pertenezcan a la empresa virtual, tales como suministradores ocasionales o clientes espontáneos.
- *Duración de las relaciones.* Es decir, si el marco de tiempo de la relación de negocio entre los socios de la VE es a corto, medio o largo plazo. Algunas alianzas de empresas se establecen para satisfacer una oportunidad de negocio específica, y se disuelven una vez dicha oportunidad ha sido satisfecha (Nayak *et al.*, 2001; Hales y Barker, 2000; Christie y Levary, 1998). Esta situación se corresponde, quizás, con la clase más representativa de empresa virtual (Camarinha-Matos y Afsarmanesh, 2001): podemos encontrar ejemplos en muchas de las industrias orientadas a proyectos como consultoría, seguros comerciales, construcción, etc., que satisfacen los requisitos de los clientes con productos y servicios no estándares (Nayak *et al.*, 2001), o en aquellas industrias en las que la necesidad de responder rápidamente a las condiciones de mercado y a la alta volatilidad de la demanda, exige un rápido desarrollo de productos con ciclos de vida muy cortos, como es el caso de la industria electrónica. Pero existen también alianzas que duran un número

indefinido de procesos de negocio o durante un periodo de tiempo definido a largo plazo, como es el caso de la industria automovilística (Ouzounis, 2001). En el primer caso es necesario una infraestructura que soporte una creación y disolución dinámica del consorcio. En el segundo, sin embargo, el énfasis se pone en la operación de la VE y en el soporte para la definición y supervisión dinámica de los procesos de negocio (Camarinha-Matos y Afsarmanesh, 2001). Esta dimensión también determina el periodo de tiempo en el que deben obtenerse los retornos de la inversión.

- ✓ *Plazo de recuperación de la inversión.* Los retornos de la inversión pueden producirse bien sólo durante la vida útil de la relación, durante una única provisión, o bien pueden producirse a lo largo de todo el periodo de tiempo que dure la participación de la entidad en un entorno de gestación.
- ✓ *Oportunismo.* El establecimiento de las alianzas se basa en las posibilidades de dar satisfacción a una oportunidad de negocio y se disuelven al final de tal proceso (Nayak *et al.*, 2001), es decir, mientras resulten productivas y beneficiosas para los agentes (Byrne *et al.*, 93). La única razón para preservar la cooperación es la existencia de una oportunidad específica (Adams *et al.*, 2001).
- *Grado de dispersión de los agentes.* Podemos considerar esta característica desde tres puntos de vista diferentes, aunque al mismo tiempo complementarios: desde una perspectiva geográfica, cultural y funcional (Pelechano *et al.*, 2005).
 - ✓ La *dispersión geográfica* atiende a la localización de los distintos agentes participantes en la red. A pesar de que la lejanía física de los agentes no es un requisito esencial en el concepto de VE, ya que el concepto en sí se mantiene independientemente de la localización geográfica de las entidades, lo cierto es que si queremos seleccionar en todo momento los socios más adecuados no podemos limitarnos al entorno geográfico más cercano. En la actualidad la distancia no influye en el movimiento de información, pero sí es una importante restricción a la hora de cumplir los plazos en la provisión del servicio y para la logística de materias primas y productos (Hales y Barker, 2000). Por lo tanto, la dispersión geográfica es más factible cuando los servicios o los productos involucrados son no físicos, como es el caso de los productos digitales.

- ✓ La *dispersión cultural* es algo natural y propio de los sistemas colaborativos, y en particular de la VE, dado que al estar involucradas entidades legalmente independientes aportan al sistema patrones culturales diferentes. El éxito descansa, por tanto, en la capacidad de aunar todas esas culturas de la forma más apropiada (sinergias positivas).
 - ✓ En cuanto a la *dispersión funcional*, ésta se refiere al grado de especialización de los agentes participantes en la VE. Cuanto mayor sea dicho grado de especialización, la complementariedad de objetivos y la especificidad de los conocimientos, más nos acercaremos a un modelo ideal de empresa virtual.
- *Coordinación.* Desde el punto de vista de la coordinación de la red podemos encontrar varios enfoques. En algunos sectores, como el de la industria automovilística, existe una compañía dominante “rodeada” por una red de suministradores relativamente fija (estructura de coordinación centralizada o en estrella) (Gisela y Korsunsky, 2003). La compañía dominante define las reglas del juego e impone sus propios estándares a los otros miembros, principalmente en términos de los modelos de proceso de negocio, mecanismos de intercambio de información y derechos de acceso, entre otros. El concepto de empresa extendida puede usarse para describir este caso particular, dado que representa una empresa dominante extendiendo sus límites sobre satélites de suministradores y proveedores de servicio. También podemos encontrar un tipo de organización diferente en algunas cadenas de suministro sin una compañía dominante (alianza democrática). En estas redes todos los nodos cooperan con las mismas reglas, en igualdad de condiciones, preservando su autonomía, pero uniendo sus competencias nucleares (Camarinha-Matos y Afsarmanesh, 2001). Pero, incluso en este caso, es necesario un nodo coordinador para administrar la información general que tiene que ver con los miembros de la VE, y para vigilar la estructura organizacional y los principios de cooperación en común. En un caso extremo, y tras haberse constituido con éxito la alianza, las compañías pueden darse cuenta de los beneficios de una gestión en común de los recursos, habilidades, etc., y pueden tender a crear un clase de estructura de coordinación en común (federación).
 - ✓ *Grado de autonomía e igualdad de los participantes.* Los socios pueden tener sus procesos de negocio acoplados o bien pueden cambiar cualquier proceso interno manteniendo

su autonomía dentro del contexto de la sociedad. Normalmente se aplica algún tipo de dirección compartida donde cada participante controla sus propios recursos y el control del conjunto bien se dispersa o se concentra.

- ✓ *Grado de integración.* Refleja si un miembro de la VE tiene totalmente integrados sus procesos de negocio con otras organizaciones siguiendo un modelo perfectamente acoplado o bien presenta un modelo débilmente acoplado con una integración limitada.
- *Exclusividad de los miembros.* Otra faceta a considerar es la posibilidad de que una empresa participe simultáneamente en múltiples alianzas o que esté dedicada por completo a una única VE (exclusividad). En un caso no-exclusivo, la infraestructura de apoyo debe manejar diferentes espacios de participación y cumplir estrictas reglas de cooperación y visibilidad de la información, para preservar las exigencias de cada una de las empresas individuales. A fin de cuentas, dado que un socio puede ser miembro de varios proyectos de VE paralelos, dos empresas pueden ser simultáneamente socios en un proyecto y competidores en otro (Katzy y Miralles, 1999).
- *Implantación de las ICT (comunicación e intercambio de datos electrónicos).* Determina la medida en la que los procesos de negocio están soportados por tecnología de información y si hay interfaces y estándares comunes para el intercambio de datos (es decir, herramientas, plataformas y protocolos). Si bien es cierto que un pequeño número de autores consideran que la integración electrónica no es necesaria en la operación de la VE, desde un punto de vista estructural se asume que las tecnologías de la información son herramientas que posibilitan los canales de comunicación y la transferencia de recursos y capacidades complementarias entre los participantes (Goldman *et al.*, 1995). Las ICT permiten distribuir y procesar información en tiempo real a través de todo el sistema uniendo a clientes y distribuidores, permitiendo que todos los participantes tomen decisiones y coordinen la acción de un modo ágil y rápido (Hales y Barker, 2000).
- *Sustentada en la confianza o en contratos.* ¿Todas las responsabilidades son concretadas en contratos formales o, por el contrario, hay un cierto grado de libertad operacional sustituyendo la supervisión y los contratos por la confianza?. La confianza entre los socios requiere información precisa sobre los otros que puede obtenerse a partir de una experiencia anterior.

- *Centrada en la eficiencia del proceso o en el valor por transacción.* Es decir, si los miembros de la empresa virtual están cooperando a nivel de proceso o a nivel de transacciones, lo que tiene implicaciones directas sobre la integración y las relaciones a lo largo de su existencia. Podemos tener socios que trabajen como parte de una red estática más grande centrada en alcanzar la eficiencia del proceso total, o socios que trabajen en base a transacciones y que necesiten centrarse en alcanzar la eficiencia dentro de cada transacción particular.
- *Campo de visión.* Muestra “a qué distancia” a lo largo de la red un nodo puede ver la configuración de la VE. En la mayor parte de los casos un nodo solamente ve a sus vecinos directos (nivel elemental de visibilidad), como ocurre en la mayoría de las cadenas de suministro. Sin embargo, en situaciones que requieren una coordinación más avanzada, un nodo puede tener ciertos derechos de visibilidad sobre otras empresas que no están directamente relacionadas (visibilidad muti-nivel), incluyendo ciertos niveles de visibilidad de acceso a la información, que podrían conducir a una operación más eficiente de la empresa virtual. En concreto, el control del cumplimiento de los pedidos, la planificación, la distribución de la carga de trabajo, la gestión optimizada de los recursos disponibles, etcétera, son algunos ejemplos de supervisión y coordinación avanzadas de tareas de la empresa virtual que requieren un alcance de visibilidad extensivo. A pesar de los beneficios previsibles, este nivel de participación de información es difícil de lograr: a menos que las entidades involucradas compartan un alto nivel de confianza, la tendencia natural es que cada compañía esconda en lo posible su información estratégica (Gisela y Korsunsky, 2003). Los campos de visibilidad se acuerdan normalmente de modo bilateral entre las empresas, o a través de contratos obligatorios suscritos por todos los miembros implicados en la relación y el coordinador de la misma.

En la sección siguiente se ofrece una clasificación de los diferentes modelos de empresa virtual existentes a partir de estos criterios clave. Además, se analizarán las características de cada uno de ellos, proporcionando algunos ejemplos que permitan clarificar sus ventajas y desventajas. Este análisis nos conducirá a convenir ciertas conclusiones relativas a la aplicabilidad y complejidad de cada modelo, y a sus necesidades desde el punto de vista tecnológico.

1.5 TIPOLOGÍAS

A pesar de que no existe una definición académica estricta del concepto de empresa virtual, existen diferentes modelos que muestran características técnicas y de negocio comunes. Empleando los criterios descritos en el apartado anterior como criterios de clasificación se pueden identificar las siguientes categorizaciones de empresas virtuales:

- Empresas virtuales intraorganizacionales e interorganizacionales.
- Empresas virtuales estáticas y dinámicas.

Tal y como se discutirá más adelante, las empresas virtuales interorganizacionales y dinámicas constituyen los modelos de negocio más prometedores desde un punto de vista económico, ya que presentan un gran número de características relevantes. Sin embargo, también son más problemáticas desde una perspectiva técnica, ya que requieren soluciones y sistemas más complejos, sofisticados y distribuidos.

1.5.1 Empresas virtuales intra e interorganizacionales

Una empresa virtual puede ser bien una única compañía que opera aplicando modelos organizacionales novedosos, tal y como pusieron de manifiesto Davidow y Malone (Davidow y Malone, 1992) en su conocido libro “The Virtual Corporation-Structuring and Revitalising the Corporation for the 21st Century”, o bien varias compañías que se unen para colaborar y compartir recursos y procesos traspasando sus propias fronteras.

La *forma intraorganizacional* conlleva la utilización de entornos y equipos de trabajo virtuales. Mediante ellos se puede unir la experiencia y maestría de especialistas de diferentes dominios complementarios, por medio de tecnología de información y comunicación, para lograr una cierta meta o proyecto definido dentro de la propia empresa. Los miembros pueden proceder de localizaciones globalmente diseminadas y se disuelven una vez se satisface el motivo de su existencia.

Cuando los individuos y empresas especialistas no están circunscritos a una única compañía hablamos de una *empresa virtual interorganizacional*. Bajo este término podemos incluir tanto formas estáticas de cooperación entre compañías como sociedades dinámicas de diferentes corporaciones

interconectadas en red mediante las ICT. No obstante, algunos de los conceptos intraorganizacionales también se pueden utilizar para coordinar la cooperación entre diferentes entidades; por ejemplo, pueden existir equipos virtuales formados por especialistas de diferente procedencia y con las competencias esenciales necesarias para la cooperación interorganizacional.

En un entorno de negocio tan hipercompetitivo como el actual, las compañías no pueden permitirse el lujo de ser las segundas en nada de lo que hacen, aunque tampoco sería realista esperar que fuesen mejor que el resto de sus competidores en todo. Además, las exigencias de los nuevos mercados surgen cada vez más rápido y las organizaciones individuales a menudo no tienen las habilidades o competencias necesarias para afrontarlas de un modo eficiente y ágil. Ante estos hechos, resulta indiscutible la conveniencia de los modelos interorganizacionales, donde diferentes empresas complementarias aportan a la colaboración aquellos procesos en los que sí son mejores que sus competidoras.

1.5.2 Empresas virtuales estáticas

En las *empresas virtuales estáticas* (SVE, *Static Virtual Enterprise*) un conjunto de entidades se unen de un modo fijo y estable, dando lugar a relaciones de negocio perfectamente integradas, personalizadas y de larga duración (NIIP, 2003; Malone y Rockart, 1991). La red también es fija, configurando una estructura estática y predeterminada. En base al estilo de gestión y distribución de la red, se pueden identificar dos tipos de empresas virtuales estáticas: la centralizada y la descentralizada (Stricker *et al.*, 2000; Ouzounis, 2001).

En las *VE estáticas centralizadas* (CSVE, *Centralised Static VE*) una entidad o dominio de negocio dominante coordina las relaciones entre todos los miembros de la red, hace cumplir las interfaces técnicas entre los socios e integra sus procesos creando procesos compartidos. La gestión, tanto de estos procesos compartidos como de la infraestructura técnica subyacente, la realiza de una forma centralizada y estática (NIIP web, 2003; Wognum y Faber, 1999).

Los socios y la entidad central forman relaciones a largo plazo centrándose en los retornos de la inversión a lo largo de su vida útil. Además, los costes de reingeniería, desarrollo e integración requeridos son altos para todos los

miembros (McCaffer y Garas, 1999). Se deduce, por tanto, que la empresa virtual estática y centralizada se corresponde con el modelo organizacional de Empresa Extendida presentado al comienzo del presente capítulo.

Ejemplos típicos de CSVE son los modelos que se han venido aplicando en la fabricación de automóviles. En este caso, un gran fabricante tiene una red de suministradores, distribuidores, vendedores y proveedores de servicios complementarios que trabajan juntos en las diferentes fases del proceso de desarrollo, producción, distribución y venta del producto. El fabricante dominante tiene necesidades y exigencias específicas, y hace cumplir sus requerimientos para incrementar el grado de automatización y disminuir los costes de distribución y producción. Los miembros de la red cooperan con la empresa dominante central adoptando e integrando las interfaces previamente especificadas.

Podemos encontrar otros muchos ejemplos de CSVE en el sector textil o en la industria de los ordenadores. Este es el caso de Dell (Dell web, 2006), un fabricante de ordenadores personales que simplemente se dedica a ensamblar los componentes y distribuir el producto final al consumidor sin fabricar ni un solo componente: Dell sólo posee de manera virtual la capacidad de fabricar ordenadores. Su ventaja competitiva se halla en que posee una red de suministradores (fabricantes de componentes, servicio técnico, operadores logísticos, etc.) altamente integrada que le permiten producir *just-in-time*¹⁰.

Una *empresa virtual estática descentralizada* (DSVE, *Decentralised Static VE*) se caracteriza porque diferentes socios de negocio se unen de una manera bastante descentralizada y autónoma. Este tipo de red es similar a la anterior salvo que no existe una organización de gestión central dominante y cada miembro de la red puede cooperar con muchos otros dominios (Malone y Rockart, 1991). Ninguno de los socios tiene control total sobre la red y la infraestructura subyacente, mientras que la integración entre los procesos de negocio de los miembros se realiza de una manera conjunta, coordinada y gradual. Los socios forman relaciones de negocio a largo plazo y obtienen retornos de la inversión a lo largo de su vida útil (Frederix, 1998). Los costes de integración y desarrollo son bastante altos y la modificación de la red es casi imposible (Wognum y Faber, 1999).

10 Es decir, produce un bien personalizado en el momento demandado y sin *stocks* intermedios.

Muchas de las empresas de alta tecnología ejemplifican este modelo organizativo (Ouzounis, 2001). Por ejemplo, algunas de las empresas dedicadas a la fabricación de semiconductores, al ensamblaje de placas, etc., se centran en una actividad de la cadena de valor y se asocian con otras organizaciones para participar en múltiples cadenas de valor. Cada socio desempeña un papel en la empresa virtual y aporta esencialmente sus competencias nucleares propias (es decir, sus recursos, procesos de negocio, etc.) sin que exista una entidad dominante.

Ambos arquetipos son estáticos en el sentido de que no hay una evolución de la red una vez ésta se ha constituido. Es cuestionable si las SVE pueden catalogarse bajo el concepto de empresas virtuales dado que también se las puede ver como redes corporativas tradicionales donde las entidades colaboran en lugar de adquirir en el mercado los servicios necesarios. En este sentido, el concepto de empresa “virtual” puede ser equivalente al de empresa “aparentemente existente” (Jansen *et al.*, 1999), dado que los clientes creen que están tratando con una única compañía cuando en realidad lo hacen con una red de compañías.

1.5.3 Empresas virtuales dinámicas

El término “virtual” tiene otro significado distinto al mencionado en el apartado anterior, el de “potencialmente existente” (Jansen *et al.*, 1999). En este sentido, las redes dinámicas son organizaciones de rápida creación y rápida disolución (Browne y Zhang, 1999), que surgen solamente en el momento en que aparece una oportunidad específica de negocio (aparece un nicho de mercado, un cliente requiere que se le suministre un producto o se le preste un servicio, etcétera), desapareciendo una vez se satisface dicha oportunidad de colaboración.

En las *empresas virtuales dinámicas* (DVE, *Dynamic Virtual Enterprise*) un conjunto de socios de negocio se unen dinámicamente, bajo demanda, y de acuerdo con las necesidades y requisitos de los clientes (Ouzounis, 2001). La relación entre los integrantes es temporal, dado que acontece sólo durante el periodo de tiempo en que una orden específica está en proceso. Además, dichas relaciones no son fijas, por lo que la VE puede cambiar continuamente en base a criterios orientados al mercado (Doz y Hamel, 1998; Fielding *et al.*, 1998). Una vez la DVE se ha disuelto, sus miembros pueden buscar nuevos socios para perseguir nuevas oportunidades (Browne y Zhang, 1999).

Dada la naturaleza temporal de las DVE, la necesidad de procesos interorganizacionales comunes y la divergencia de objetivos de sus socios, es necesario el desarrollo de un sistema de gestión que sea la base para una preparación efectiva de aquellas entidades que quieren entrar a formar parte de una VE dinámica y que cubra todo su ciclo de vida. Por esta razón, a lo largo del presente trabajo se ha considerado que la formación de las DVE se da dentro del contexto de los entornos de gestación de empresas virtuales (VBE, *VE Breeding Environment*).

Los VBE representan asociaciones de organizaciones, y todas aquellas instituciones de apoyo relacionadas, que tienen tanto el potencial como el interés para cooperar entre sí a través del establecimiento de acuerdos de cooperación a largo plazo (Camarinha-Matos *et al.*, 2005c). En consecuencia, cuando uno de los miembros del VBE identifica una oportunidad de negocio se puede seleccionar rápidamente un subconjunto con aquellos miembros que mejor se adaptan a dicha oportunidad de colaboración. Dada la importancia de los VBE, en un capítulo posterior se analizará detenidamente este tipo de asociaciones.

Aunque existen ya algunas aplicaciones del concepto dinámico de empresa virtual, prácticamente ninguna muestra una forma pura, es decir, que satisfaga todas las características analizadas. Podemos encontrar ejemplos en muchas de las industrias orientadas a proyectos como consultoría, seguros comerciales, construcción, viajes, etc., que satisfacen los requisitos de los clientes con productos y servicios no estándares (Nayak *et al.*, 2001), o en aquellas industrias en las que la necesidad de responder rápidamente a las condiciones de mercado y a la alta volatilidad de la demanda, exigen un rápido desarrollo de productos con ciclos de vida muy cortos, como es el caso de la industria electrónica o la industria informática.

Dentro del mundo informático podríamos destacar los proyectos de software *Open Source* como ejemplos especiales de DVE, ya que pueden ser vistos como realizaciones puras de este concepto, claro está, en un contexto no comercial. Si serían ejemplos comerciales “Dhs consult” y “Rausser Advertainment AG” (Bischofberger, 2004; Hoffman, 2001), empresas dedicadas respectivamente al diseño gráfico en Internet y a la creación de juegos de ordenador. Se trata, en ambos casos, de empresas con muy pocos componentes permanentes (uno en el primer caso) pero que disponen de un *pool* significativo de socios potenciales, geográficamente dispersos, con los que interactúan *online* (negociación, búsqueda de socios, etc.). Dada la

necesidad de rapidez que acompaña a sus proyectos, la confianza es esencial, por lo que la experiencia en proyectos pasados se convierte en una razón de peso a la hora de volver a contratar a los socios.

En el sector textil podemos mencionar la empresa Topsy Tail que fabrica accesorios para la industria de la moda. Esta empresa actualmente genera ingresos por valor de 80 millones de dólares pero sólo cuenta con tres empleados y nunca toca el producto en todo el proceso de manufactura o distribución. La empresa contrata a empresas manufactureras para que fabriquen sus productos, agencias de diseño para que diseñen los productos y envoltorios, y una red independiente de distribuidores y representantes de ventas que se encargan de distribuir y vender los productos (Bhattachryya y Rahman, 2002).

Otro área interesante donde este concepto está siendo aplicado es el de los operadores logísticos (Ouzounis, 2001). Estas compañías pueden registrar sus procesos potenciales en un mercado especializado (por ejemplo, el reparto de un paquete) especificando las propiedades de sus procesos (destinos alcanzables, precio, tiempo necesario para transportar el paquete, garantía ofrecida, etc.). Los dominios de negocio que precisan un servicio logístico buscan automáticamente en el mercado, seleccionan al mejor socio existente en ese momento en base a ciertos requisitos y utilizan el servicio ofertado. Para el cliente inicial de la VE dinámica el proceso entero resulta totalmente transparente.

A la vista de lo anterior, resulta evidente que las empresas virtuales dinámicas mejoran significativamente a las estáticas al aprovecharse totalmente de las oportunidades globales y abiertas que ofrecen Internet y la economía global; en definitiva, porque son capaces de responder ágilmente a los retos y desafíos que plantea el nuevo entorno empresarial. Es por esta necesidad de agilidad por lo que a lo largo de los últimos años ha emergido con gran fuerza el concepto de *empresa virtual ágil* (AVE, *Agile Virtual Enterprise*) como sinónimo del paradigma de empresa virtual dinámica.

Una vez descritos los diferentes tipos de empresas virtuales, y tras introducir el concepto de VE ágil como equivalente de la VE dinámica, en el siguiente apartado se va a analizar el concepto de agilidad y el significado que aporta como adjetivo al paradigma de empresa virtual. Este análisis comienza con la visión proporcionada por Goranson, a la postre el primero en introducir en la literatura el término de empresa virtual ágil.

AGILIDAD, LA PRIORIDAD COMPETITIVA DEL FUTURO

Goranson (Goranson, 1999) identifica la agilidad como la capacidad de responder y beneficiarse completamente de los cambios inesperados, es decir, de las necesidades cambiantes de los clientes. En su opinión, la agilidad es independiente de otras prácticas de gestión: que una empresa sea capaz de suministrar un producto de calidad cuando el cliente lo necesita y a un coste óptimo, no significa que posea la capacidad de cambiar eficientemente para adaptarse mejor a las necesidades de sus clientes.

Las empresas deben buscar permanentemente la agilidad, y la mejor forma de encontrarla es a través de la virtualización. La agilidad de una compañía hace referencia a su capacidad para unirse a asociaciones temporales con el propósito de responder a condiciones cambiantes. Las prácticas de gestión sofisticadas no solamente tienen que aplicarse dentro de una compañía, sino que los socios exteriores también tienen que abrazarlas como propias para hacer frente a desafíos inesperados.

La definición de empresa virtual ágil proporcionada por Goranson es muy similar a la de empresa virtual dinámica: la empresa virtual es ágil solamente si se forma con la intención de disolverse o configurarse de nuevo ante una nueva oportunidad de negocio. No importa tanto la temporalidad de la relación como la capacidad de reconfigurarse, reconstruirse y rediseñarse continuamente, así como la posibilidad de sustitución de los distintos elementos (Adams *et al.*, 2001).

Las empresas virtuales dinámicas/ágiles son agregaciones oportunistas de unidades más pequeñas que se unen y actúan como si fueran una empresa más grande y duradera. El término “virtual” significa, en este caso, que los miembros aportan al consorcio muchas de las ventajas que poseen las grandes compañías, permitiendo configurar empresas que son las mejores en todo sin incurrir en los lastres de las grandes megacorporaciones. La empresa virtual ágil representa, por tanto, un medio de crecer rápidamente con unos costes mínimos.

La empresa virtual ágil o dinámica tiene un ciclo de vida diferente al de las compañías convencionales (Adams *et al.*, 2001). Persiguen activamente oportunidades de negocio y emprender nuevos proyectos. En el curso de esas actividades a menudo se modifican sus socios, acuerdos de cooperación y recursos disponibles (Adams y Dworkin, 1997). Una empresa virtual ágil no

es una construcción duradera. Cuando se emprende un nuevo proyecto y se designa a sus participantes, una nueva instancia de empresa virtual cobra vida puesto que una oportunidad o un proyecto específico son vistos como el único motivo de su existencia.

La AVE también se distingue de las organizaciones del pasado en otros aspectos que tienen su origen en la ausencia de límites. De manera muy simplificada, las organizaciones piramidales tradicionales están basadas en mecanismos de autoridad y control que conducen a la formación de distintas oficinas o centros de dirección donde se toman las decisiones. Alguien externo a la propia empresa sabe, en mayor o menor medida, dónde se concentra el poder y cuál es la forma y la imagen de la empresa.

En clara oposición a estas empresas tradicionales, las VE ágiles no disponen de oficinas centrales y su sistema de decisión y control es distribuido. Los integrantes de la red, los socios, se autorregulan a través de estructuras de autocontrol definidas dentro del marco de trabajo (Adams *et al.*, 2001). Por otro lado, el concepto de VE ágil depende absolutamente de la innovación: dado que la razón principal para la constitución de una AVE es mejorar la respuesta a los cambios, es preciso una innovación y evolución, tanto técnica como cultural, para alcanzar las expectativas y para reaccionar de una manera adecuada y flexible.

En este contexto, la VE ágil puede verse desde tres perspectivas diferentes (Adams *et al.*, 2001). En primer lugar, puede ser considerada desde una perspectiva externa. En este caso, el cliente la percibe como una única entidad, sin percatarse de que se trata de un conglomerado de entidades. En segundo lugar, las AVE vistas desde dentro son constelaciones de varias entidades o socios independientes. Puesto que se unen al proyecto para aportar su maestría en un cierto campo pierden su condición de miembro una vez se ha prestado el servicio.

Por último, la empresa virtual ágil puede contemplarse desde la perspectiva de uno de los nodos o socios de la red. Por ejemplo, una compañía que busca competencias complementarias, o vende exceso de capacidad, puede derivar recursos (humanos, técnicos, económicos, etc.) para unirse a una AVE. Podemos concluir que una característica fundamental es que una empresa virtual ágil tiene su propia identidad mientras que la identidad de las empresas participantes generalmente permanece no visible para los elementos externos.

La AVE, al formarse idealmente como una combinación de las competencias esenciales *best of class* complementarias, es realmente una organización que es la mejor en todo, es decir, una organización *best-of-everything* (Byrne *et al.*, 1993).

1.5.4 VE estáticas versus VE dinámicas/ágiles

La valoración y comparación de los modelos de VE básicos se realiza en base a algunos de los criterios de clasificación citados anteriormente.

- *Uso de la meta-gestión.* En las DVE el grado de flexibilidad en su gestión y en la selección de proveedores y *satisfiers* que cumplan los requisitos de la organización es mucho mayor que en las SVE donde la posibilidad de sustituirlos por mejores alternativas es prácticamente nula.
- *Centrado en las competencias nucleares de los participantes.* ¿Todos los socios de la empresa virtual contribuyen con sus competencias nucleares?. En las VE dinámicas es una condición indispensable sin la cual no puede llegar a tener éxito. Sin embargo, en las SCVE existe la posibilidad de que un socio, el dominante, colabore aportando otras competencias diferentes a las distintivas.
- *Topología de la red.* Las DVE muestran una naturaleza variable y dinámica, en la que algunas empresas (socios no estratégicos) pueden unirse o abandonar la alianza dinámicamente dependiendo de las fases del proceso de negocio u otros factores del mercado. En las VE estáticas, sin embargo, la estructura establecida es casi fija, de tal modo que a lo largo del ciclo de vida de la VE sólo se dan pequeñas variaciones en términos de proveedores o clientes¹¹.
 - ✓ *Número de socios de la VE.* En el primer caso el número de socios puede cambiar dinámicamente, de acuerdo con la ley de oferta y demanda, y en función de los requerimientos de los miembros individuales del entorno de gestación. En las SVE el número de socios que participan es estático y predeterminado debido a que se requieren actividades de integración especializadas.

11 No obstante, debemos tener presente que dentro de una empresa estática es posible que algunos de los elementos “extendidos” creen, a su vez, VE dinámicas para cumplir plenamente con sus funciones.

- ✓ *Grado de evolución y escalabilidad.* El modelo dinámico de VE es, por definición, escalable y permite la evolución de la red en términos de nuevos miembros y relaciones. En las DVE no hay interfaces plenamente acopladas entre los socios y, por lo tanto, la escalabilidad y la evolución del negocio son una cuestión clave. Por el contrario, en las VE estáticas las relaciones entre los socios son estáticas y el nivel de escalabilidad bajo. Requieren altos costes de desarrollo para rediseñar la red y para cambiar las interfaces entre los socios.
- ✓ *Interacción temporal con socios exteriores.* En las SVE la interacción con otros socios que no pertenecen a la VE, tales como suministradores ocasionales o clientes puntuales, es muy reducida a diferencia de lo que ocurre en las DVE que a menudo modifican sus socios y acuerdos de cooperación en función de los recursos necesarios y disponibles.
- *Duración de la relación y oportunismo.* La colaboración en las VE dinámicas es fruto del oportunismo ya que se crean para satisfacer una oportunidad específica de negocio y se disuelven una vez finaliza tal oportunidad. Sin embargo, en las VE estáticas las relaciones suelen ser más duraderas manteniéndose durante un número indefinido de procesos de negocio o durante un periodo de tiempo definido a largo plazo.
 - ✓ *Plazo de recuperación de la inversión.* En las SVE los retornos de la inversión se producen a lo largo de la vida útil de la relación. En las VE dinámicas los retornos se dan durante una única transacción, aunque si la empresa pertenece a un entorno de gestión de VE, se garantiza retornos a lo largo del periodo de tiempo que dure su participación en la misma.
- *Coordinación.* En los modelos centralizados de VE la compañía dominante define las reglas del juego e impone sus propios estándares a los otros socios, principalmente en términos de los modelos de procesos de negocio, mecanismos de intercambio de información y derechos de acceso, entre otros. Por el contrario, en los modelos descentralizados, sin una compañía dominante, todos los nodos cooperan con las mismas reglas, en igualdad de condiciones, preservando su autonomía, pero uniendo sus competencias nucleares.
 - ✓ *Grado de distribución.* Todos los modelos tienen un buen nivel de distribución entre los procesos de negocio de los socios. Sin embargo, las VE estáticas se basan en un modelo

dominante centralizado, mientras que las dinámicas revelan, debido a la naturaleza del modelo, un mayor nivel de distribución entre los procesos y los socios de negocio.

- ✓ *Grado de integración.* Las empresas virtuales estáticas, fuertemente acopladas, exhiben una integración de procesos elevada entre los socios. Las dinámicas, débilmente acopladas, se encuentran en el extremo opuesto del espectro y muestran una integración de los procesos muy baja.
- ✓ *Grado de autonomía.* Dado que las SVE requieren un alto grado de integración entre los socios, el grado de autonomía es bastante bajo; es decir, los procesos de negocio de un miembro del consorcio dependen altamente del resto de socios. Por el contrario, las DVE muestran mayor autonomía ya que las relaciones entre los socios no son estáticas ni están perfectamente integradas; la modificación de los procesos de negocio se realiza más fácilmente.
- *Exclusividad de los miembros.* En las CSVE suele ser habitual que la empresa central exija exclusividad a la red de suministradores, vendedores y distribuidores que cooperan con ella y que adopten e integren las interfaces previamente especificadas. Sin embargo, las empresas que participan en relaciones dinámicas pueden estar comprometidas simultáneamente en más de una relación de negocio.
- *Integración electrónica.* Las relaciones dinámicas requieren que los procesos de negocio estén soportados por tecnología de información y que existan interfaces y estándares comunes para el intercambio de datos. Sin embargo, en las relaciones estáticas estas necesidades son más débiles o incluso pueden no existir.
- *Sustentada en la confianza o en contratos.* En las SVE las responsabilidades se acuerdan bilateralmente por medio de contratos formales entre todos los miembros (DSVE) o en base a las especificaciones de la organización dominante (CSVE) buscando satisfacer sus necesidades y exigencias. En los modelos dinámicos, y debido a necesidad de aprovechar rápidamente la oportunidad de negocio, se da mayor valor a la confianza permitiendo un mayor grado de libertad operacional.
- *Eficiencia del proceso vs. eficiencia por transacción.* Los socios que trabajan como parte de una organización virtual grande se centran en alcanzar la eficiencia del proceso total mientras que aquellos que trabajan en base a transacciones necesitan centrarse en alcanzar valor dentro de las transacciones.

La tabla 1.1 resume el conjunto de consideraciones efectuadas en los párrafos anteriores para los modelos de empresa virtual estático y dinámico.

		<i>Empresas virtuales</i>	
		<i>Estáticas</i>	<i>Dinámicas</i>
Uso de la meta-gestión		Bajo / nulo	Alto
Centrado en las competencias nucleares		No siempre	Sí
Topología de la red		Estática	Dinámica
	Número de socios	Estable	Variable
	Grado de evolución	Bajo	Alto
	Escalabilidad	Baja	Alta
	Interacción temporal con socios exteriores	Baja / nula	Alta
Duración		Medio / largo plazo	Corto plazo
	Oportunismo	Bajo	Alto
	Plazo de recuperación	Vida del proyecto	Permanencia en el VBE
Coordinación		Principalmente centralizada	Normalmente descentralizada
	Grado de distribución	Medio	Alto
	Autonomía	Baja	Alta
	Integración	Altamente acoplados	Débilmente acoplados
Exclusividad		Normalmente	Casi nunca
Integración electrónica		Baja / no necesaria	Alta / imprescindible
Centrada en la confianza o en contratos		Contratos	Confianza
Eficiencia del proceso		Alta	Media
Eficiencia de la transacción		Baja	Alta

Tabla 1.1. Comparativa entre los arquetipos estático y dinámico de empresa virtual.

En base a estos criterios de clasificación y selección, se deduce fácilmente que las DVE son modelos de negocio con un mayor número de características deseables y, por tanto, más prometedores desde el punto de vista de los negocios. Sin embargo, aunque representan una opción atractiva de entender los negocios, las DVE deben enfrentarse a múltiples dificultades. Las más obvias son aquellas de naturaleza tecnológica (Chituc y Azevedo, 2005), y se refieren principalmente a asuntos como la integración e interoperabilidad, seguridad, certificación, infraestructuras de soporte para los negocios, arquitecturas de referencia, etc., tanto a nivel intra como interorganizacional.

Gracias a los recientes avances en telecomunicaciones, redes abiertas como Internet, lenguajes de programación independientes de la plataforma como Java, metalenguajes flexibles como XML (*eXtensible Markup Language*), etc., se están empezando a construir los cimientos sobre los que se asentará el desarrollo de plataformas de gestión que puedan consumir el concepto de DVE. Gracias a los mecanismos abiertos que proporciona la nueva economía basada en Internet, las VE interorganizacionales, dinámicas, ágiles, autónomas, flexibles y capaces de aprovecharse de las condiciones del mercado son la respuesta organizativa y de gestión de futuro.

1.6 VIRTUALIDAD Y VIRTUALIZACIÓN

No podemos finalizar este capítulo de generalidades sobre la empresa virtual sin hablar sobre el proceso a través del cual una organización adquiere las características necesarias para participar de una VE (Chaffey, 2002), es decir, sin hablar del proceso de *virtualización*. Pero antes, conviene tener presente que la mayoría de las corporaciones tradicionales exhiben alguna de las características de una empresa virtual, es decir, la *virtualidad* es una característica de los sistemas empresariales que se puede presentar de diferentes formas y grados (Martins *et al.*, 2004).

En este sentido, la distinción entre corporaciones virtuales y no virtuales debe sustituirse por el concepto de virtualización gradual, que clasifica a las organizaciones en un *continuum* según su grado de adopción de las estructuras virtuales. En un extremo estarían las compañías jerárquicas tradicionales, y en el otro encontraríamos una forma ideal de empresa virtual dinámica/ágil que muestre todos los atributos mencionados anteriormente, y especialmente un alto carácter temporal y una baja estabilidad de las relaciones.

Esta nueva forma de valoración aporta algunas ventajas. En primer lugar, permite analizar el desarrollo de estructuras virtuales a lo largo del tiempo y cómo influyen los diferentes elementos ambientales (Bischofberger, 2004), lo que proporciona interesantes revelaciones sobre por qué evolucionan las estructuras virtuales. Por otro lado, comparando el funcionamiento de compañías con diferentes grados de virtualización se pueden realizar pruebas empíricas que permitan conocer los beneficios de los conceptos virtuales en ambientes dinámicos. Finalmente, si podemos clasificar una compañía según su grado de virtualización también podremos determinar su posición actual hacia la virtualización e identificar las oportunidades de desarrollo que le restan (Bauer y Köszegi, 2003).

Existen diversas corrientes de explicación que tratan de dilucidar cómo debe ser la evolución de una compañía a lo largo del *continuum*. Algunos de estos enfoques sugieren que para poder responder al dinamismo de los mercados y a las oportunidades que ofrece la tecnología, una empresa debe seguir una trayectoria evolutiva a través de diferentes fases adoptando estructuras de organización cada vez más virtuales. Este es el caso del enfoque propuesto por Mertens *et al.* (Mertens *et al.*, 1998). Estos autores proponen un modelo con cinco etapas de desarrollo que permiten a la organización evolucionar desde una estructura no-virtual, situación correspondiente a la *etapa 0*, hasta una empresa virtual (etapa 4), tal y como aparece reflejado en la figura 1.3.

Una corporación verticalmente integrada alcanza la *primera etapa* evolutiva reduciendo el número total de unidades de negocio mediante la concentración de actividades independientes en una región geográfica (por ejemplo, uniendo la producción de una pieza específica en una única instalación). Mediante decisiones de subcontratación de las unidades de negocio (previamente integradas) a corporaciones independientes se forma una red de compañías para proveer los mismos bienes y servicios. En el momento en que se alcance esta *segunda etapa* evolutiva, se requieren sistemas de información intercompañía para recoger las ventajas derivadas de la integración de entidades complementarias.

La *tercera etapa* evolutiva remarca la integración electrónica de clientes y/o suministradores dentro del proceso de generación de valor. La palabra clave aquí es la orientación al cliente; la meta es que los clientes sean informados adecuadamente sobre sus compras y que puedan configurar sus productos y servicios de acuerdo con sus deseos. La compañía confiará aún más en los sistemas de información en la *cuarta etapa*. En este momento la compañía ha

subcontratado todas las capacidades no-nucleares. Ahora puede limitar sus actividades a la función de coordinación y actuar como “broker de información” usando selectivamente sus relaciones en la red¹².

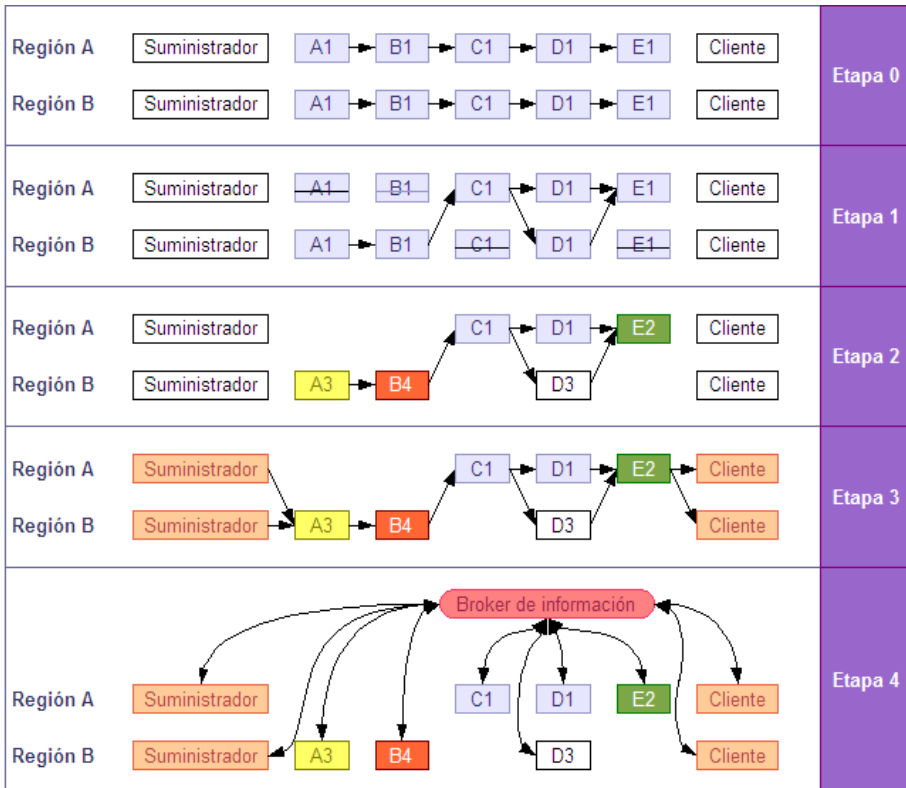


Figura 1.3. Etapas involucradas en la evolución hacia la empresa virtual (Mertens *et al.*, 1998). Los rectángulos (por ejemplo A1) representan las competencias clave: las letras indican el tipo de competencia; los números y los colores muestran las distintas empresas que están participando en la red; y las flechas señalan la integración soportada por las tecnologías de información y comunicación.

12 El término “*hollow corporation*” (corporación hueca) utilizado en este contexto representa el riesgo de que una compañía acabe finalmente sin competencias nucleares; se ha deshecho de todas las actividades clave perdiendo su capacidad de diferenciación y, por lo tanto, sus ventajas estratégicas.

Otra alternativa para clasificar a las empresas es la presentada por Venkatraman y Henderson (Venkatraman y Henderson, 1998). Estos autores proponen aplicar un enfoque estratégico puesto que consideran la virtualidad como una característica estratégica aplicable a toda organización. Para ellos, organizar “virtualmente” consiste en crear, educar y emplear activos intelectuales y de conocimiento claves mientras se subcontratan activos físicos tangibles en una compleja red de relaciones, situando la tecnología de la información en el centro del proceso.

Venkatraman y Henderson ven la virtualidad como una estrategia que refleja tres vectores distintos pero interdependientes, lo que implica que para virtualizar una compañía se han de fomentar todos ellos cada vez que se diseñe un nuevo modelo de negocio.

- El primer vector, el de *interacción con el cliente*, se ocupa de los nuevos desafíos y oportunidades para las interacciones B2C (*Business To Consumer*). La tecnología de la información permite que el cliente pruebe productos y servicios a distancia, y que participe en su personalización.
- El vector de *contratación virtual* está relacionado con las obligaciones de una empresa para integrarse virtualmente en una red de negocio. A diferencia de lo que ocurre en los modelos de organización integrada, las empresas administran una cartera dinámica de relaciones para conectar y coordinar los recursos requeridos y, de este modo, distribuir valor a los clientes por medio de Internet y otras redes.
- Finalmente, el vector de *influencia del conocimiento* se ocupa de las oportunidades para las fuentes destacadas de maestría, tanto dentro como fuera de los límites de la organización.

Partiendo de su visión estratégica, establecen tres etapas distintas de virtualidad para cada vector. La primera se centra en la unidad de tarea, es decir, el proceso de negocio se rediseña a lo largo de toda la cadena de valor añadido. La segunda etapa, en el nivel de organización, reconfigura la red de negocio realizando y desarrollando nuevas capacidades. La tercera etapa, en un nivel interorganizacional, está dirigida a crear nuevo valor redefiniendo el alcance del negocio mediante la colaboración con los socios y aprovechando su excelencia. El adalid de nuevo valor es el conocimiento y debe ser soportado necesariamente por una plataforma ICT integrada y eficaz.

Los dos acercamientos anteriores aconsejan una trayectoria predeterminada a través de diversas etapas de evolución desde una empresa no-virtual a otra virtual. Por lo tanto, las empresas se pueden clasificar según su etapa de desarrollo. Pero no está explícitamente claro porqué una empresa debe seguir tal trayectoria. Hay ejemplos en la industria de servicios donde han surgido formas ideales de empresas virtuales sin seguir ninguna trayectoria evolutiva. Además, estos modelos son teóricos y no ofrecen ningún instrumento formalizado para medir el grado de virtualización.

Una manera alternativa de medir dicha virtualización consiste en proponer una VE ideal que exhibe características predeterminadas como referencia para la valoración. Siguiendo tal método, Bauer y Köszegi (Bauer y Köszegi, 2003) midieron el grado de virtualización usando siete factores derivados de las características representativas de las empresas virtuales. A continuación, concibieron preguntas para cada factor y las reunieron en un cuestionario que enviaron a compañías consultoras. La tabla 1.2 enumera dichos factores en relación a tres dimensiones: estrategia, estructura y proceso.

<i>Factor</i>	
Estrategia	<i>Creación virtual de valor:</i> mide el grado de modularización del proceso de creación de valor.
	<i>Centrado en las competencias nucleares:</i> valora el grado en que las empresas se centran en sus competencias nucleares.
Estructura	<i>Acuerdo contractual o formal entre los socios de la cooperación:</i> evalúa el grado en que se usan contratos, reglas u organizaciones.
	<i>Independencia de los socios:</i> mide el grado de independencia horizontal y vertical.
	<i>Características de la red:</i> incluye aspectos como la duración de la corporación, la selección y combinación de los socios, la apariencia hacia los clientes y hacia los otros.
Proceso	<i>Confianza como mecanismo de coordinación:</i> mide la atmósfera general así como la confianza y la justicia dentro de la red.
	<i>Implementación de tecnología de información y comunicación:</i> tiene en cuenta los sistemas que se utilizan para facilitar la cooperación y en qué medida se utilizan.

Tabla 1.2. Factores usados para medir el grado de virtualización en (Bauer y Köszegi, 2003).

Para evaluar las respuestas, a éstas se les asignó un valor comprendido entre cero y cuatro que indicaba el impacto de la virtualización. Posteriormente se creó una escala agregada que sumaba los valores de las respuestas a todas las preguntas dando lugar al grado total de virtualización.

La dificultad de medir el grado de virtualización procede de las características propias de este tipo de organizaciones que pueden tener una breve duración y poseer fronteras fluidas y vagas (pueden desaparecer incluso antes de que uno tenga la oportunidad de observarla más de cerca). El enfoque anterior evitó este problema concentrándose en compañías individuales (los nodos en la red) y sus relaciones con los socios.

Sin embargo, con este método no se observa una empresa virtual entera sino solamente sus partes. Además, una compañía puede estar potencialmente implicada en diferentes redes y las fronteras de la red no se pueden percibir. Los estudios a largo plazo requeridos para soportar esta teoría son, por lo tanto, virtualmente inalcanzables puesto que una empresa virtual, por definición, se disuelve después de un cierto tiempo (Bauer y Köszegi, 2003).

Es también discutible si las medidas elegidas son adecuadas para deducir el grado de virtualización de una compañía. Por ejemplo, la confianza se puede considerar como inherente en muchos negocios tradicionales tales como ingeniería o consultoría. Concluir que la confianza es significativa no implica que una empresa virtual sea operativa.

Análogamente, medir la ICT no tiene un alto poder explicativo si se cuenta solamente con medios modernos (correo electrónico, videoconferencias, salas de equipo virtuales, etc.), descuidando el teléfono o la comunicación personal. Esto conduciría a inclinarse hacia compañías grandes, en las que es más probable que se generen fondos para acordar inversiones, y hacia compañías de procesamiento de datos, que consideran estos medios como engranajes de sus competencias clave.

Finalmente, debe observarse que este estudio estaba relacionado con las VE interorganizaciones. Tendencias de virtualización dentro de una compañía, tales como los equipos virtuales, solamente se tuvieron en cuenta marginalmente. Mientras no exista una definición inequívoca o un marco que delimite claramente lo que se puede incluir bajo los términos de empresa u organización virtual sirve de poco perfilar hipótesis sobre las características de una empresa virtual y verificarlas empíricamente (Bauer y Köszegi, 2003).

1.7 RESUMEN

Las empresas son seres vivos y como tales su existencia se ve profundamente influenciada por la necesidad de sobrevivir. Esta es la razón fundamental por la que durante siglos han buscado nuevos métodos y técnicas que les permitiesen hacer frente, de manera individual, a los numerosos retos que iban surgiendo, entre ellos la creciente presión competitiva. Sin embargo, las exigencias de los nuevos mercados afloran cada vez más rápido y las organizaciones individuales a menudo no tienen las habilidades ni las competencias necesarias para afrontarlas de un modo eficiente y ágil.

Sin embargo, combinando sus áreas de conocimiento/experiencia particular con la experiencia complementaria de otras compañías, es posible crear VE capaces de progresar y responder a los nuevos requerimientos. Los integrantes de estos consorcios comparten costes, habilidades y competencias clave, lo que les permite acceder a oportunidades de negocio específicas con soluciones que no podrían proporcionar individualmente. Gracias a que cada miembro de la VE se concentra en aquellas áreas donde tiene ventaja competitiva única, conseguimos una organización que es la mejor en todo.

A lo largo del presente capítulo se ha pretendido proporcionar un análisis general del concepto de VE. Para ello, se ha partido de la descripción de los cambios más relevantes acaecidos en el entorno empresarial a lo largo de los últimos años que son, a fin de cuentas, los que han dado lugar a la necesidad de modelos organizacionales flexibles, adaptativos y de costes variables. Dada la gran variedad de estos nuevos modelos y a la falta de definición que acompaña a este paradigma, se han analizado algunos conceptos “próximos” para poder apreciar más nítidamente tanto las diferencias como las posibles relaciones con el modelo objeto de estudio.

A continuación se han presentado y comentado algunas de las definiciones de empresa virtual más significativas propuestas hasta la fecha. Su análisis ha servido para establecer sus características principales y para proponer un conjunto de variables de clasificación con las que distinguir los distintos tipos de VE existentes. En concreto, podemos hablar de VE intraorganizacionales e interorganizacionales, y de VE estáticas y dinámicas/ágiles. De su evaluación se llega a la conclusión de que las VE dinámicas e interorganizacionales son más interesantes desde el punto de vista de los negocios pero más problemáticas desde una perspectiva técnica ya que requieren soluciones y sistemas mucho más complejos, sofisticados y distribuidos.

Para finalizar el capítulo se han tratado los conceptos de virtualidad y virtualización, describiéndose los distintos enfoques que pretenden mostrar cómo ha de ser la evolución una compañía que desee formar parte del modelo de empresa del siglo XXI.

CAPÍTULO

2

EL CICLO DE VIDA DE LA VE DINÁMICA Y LOS ENTORNOS DE GESTACIÓN

2.1 INTRODUCCIÓN

Competitivo, global, dinámico, turbulento, etc., son sólo algunos de los adjetivos que se emplean en la literatura de investigación para describir el mercado actual y, en definitiva, calificar los desafíos a los que deben hacer frente las empresas de comienzos del nuevo milenio. En consecuencia, estas compañías han de enfrentarse a retos que, por lo general, son difíciles de superar si se afrontan desde una perspectiva individual: están abocadas, de manera inexorable, a aprender a colaborar las unas con las otras si quieren garantizar su supervivencia.

Aunque siempre debe perseguirse que tal cooperación sea lo más eficiente posible, en un contexto como el actual, donde las oportunidades de negocio surgen en lugares y momentos concretos y durante periodos de tiempo cada

vez más breves, el atributo clave que debe definir dicha cooperación es la agilidad. Por ello, lo deseable sería tener la posibilidad de encontrar lo más rápidamente posible aquel conjunto de socios que mejor se adapte a una oportunidad de negocio concreta y configurarlos, también con celeridad, constituyendo una red colaborativa capaz de explotar la oportunidad ágilmente.

El simple hecho de que entidades legalmente independientes sean capaces de unirse y configurarse ellas mismas formando una organización colaborativa orientada a la consecución de una meta u objetivo da una idea de la enorme agilidad asociada a la noción de empresa virtual dinámica (Camarinha-Matos, 2004). Sin embargo, encontrar los socios adecuados y establecer las condiciones necesarias para iniciar el proceso de colaboración puede llegar a consumir mucho tiempo y dinero, convirtiéndose, de este modo, en una tarea inhibitoria de la tan necesaria agilidad.

Entre los principales obstáculos que aparecen encontramos la falta de información (sobre los posibles socios, sus procesos, el desempeño de su actividad,...), la heterogeneidad de los socios potenciales (diferentes infraestructuras, cultura corporativa, métodos de trabajo, prácticas de negocio,...) y la falta de preparación de las organizaciones para unirse y trabajar formando redes cooperativas, etc. Pero sin duda alguna, el obstáculo por excelencia lo constituye la falta de confianza.

La selección de los socios, por tanto, no puede limitarse a un simple problema de “optimización” (enfoque tradicional), donde solamente se asignan los procesos en función del potencial y de las habilidades (competencias nucleares) de los posibles candidatos. Se han de tener presentes otros muchos factores entre los que se incluyen algunos de naturaleza subjetiva como son las preferencias personales, la confianza resultante del conjunto de experiencias previas, etc.

Así, no es de extrañar que en la literatura y en los proyectos de investigación relacionados con las redes colaborativas, y en particular con la empresa virtual dinámica, hayan sido muchos los esfuerzos destinados a intentar afrontar eficazmente este “condicionante” del éxito, dando lugar a una amplia variedad de enfoques que van desde la simple automatización computacional del proceso hasta los sistemas multiagente, pasando por los recientes servicios web, etc.

La identificación y selección de socios en la VE dinámica es un tema complejo y, además, abierto a múltiples soluciones. Parte de esta complejidad y de esta multiplicidad de enfoques alternativos se deriva de la falta de un marco conceptual común que sirva de punto de partida al análisis del problema. Partiendo de esta reflexión, el presente capítulo se ha elaborado con objeto de establecer ese marco común que facilite la generación de una solución eficiente.

Y lo primero que debemos tener presente para establecer este ámbito de investigación y trabajo es que la necesidad de identificar y seleccionar el conjunto de socios que mejor se adapte a una oportunidad de negocio puede aparecer en diferentes momentos de la existencia de la VE dinámica: bien al principio, cuando es necesario configurar este tipo de red colaborativa (creación), o bien durante su operación, porque se detecte la necesidad de reconfigurarla (evolución). Creación y evolución son dos etapas importantes del ciclo de vida de la DVE, aunque no son las únicas.

Por otro lado, a lo largo del trabajo que se resume en la presente memoria se ha considerado que el proceso de creación de la VE dinámica acontece en el contexto de los entornos de gestación de empresas virtuales. Como ya veremos más adelante, se trata de asociaciones colaborativas a largo plazo compuestas por organizaciones que están preparadas para colaborar y, de este modo, responder rápida y ágilmente a las oportunidades de colaboración que surjan. En definitiva, se trata de asociaciones destinadas a paliar algunos de los obstáculos asociados a la temporalidad de la VE dinámica, entre los que se encuentra la ya mencionada “falta de confianza”.

Teniendo presentes estas reflexiones, el presente capítulo se ha estructurado en dos bloques principales. En el primero se describen de forma breve todas las etapas por las que evoluciona la VE dinámica, prestando una especial atención a aquellas en las que la selección de socios juega un papel tan crucial. En el segundo bloque se esbozan las características más importantes de los entornos de gestación. El capítulo finaliza con un análisis de los diferentes roles¹³ involucrados en la existencia de las DVE, incluyendo los roles más representativos, desde el punto de vista de la identificación y selección de los socios, de los entornos de gestación.

13 Un rol puede ser definido como (Kendall, 1998) una descripción abstracta de la función esperada de una entidad. Es similar a la noción de actor en una obra o a la de profesión en una organización.

2.2 EL MODELO DE CICLO DE VIDA DE LAS DVE

Un modelo de ciclo de vida (LCM, *Life-Cycle Model*) normalmente describe las fases y actividades clave requeridas durante la existencia de una entidad. De acuerdo con la especificación ISO/IEC 9596-1:1998 (ISO/IEC, 1998) de la organización ISO (*International Organization for Standardization*), el ciclo de vida puede definirse como el conjunto de pasos finitos que un sistema puede recorrer a lo largo de toda su historia. Las diferentes fases del ciclo de vida definen los tipos de actividades que son apropiados durante su existencia.

Para los objetivos de la presente tesis, tan sólo consideraremos los aspectos relacionados con el ciclo de vida de las VE ágiles, aunque muchos de los mismos se pueden aplicar, en mayor o menor grado, a cualquier tipo de organización virtual. Por ejemplo, en el caso de una empresa virtual estática primarán más aquellos aspectos relacionados con la operación de la empresa, mientras que para las DVE, en donde se precisan altas cotas de rapidez y agilidad, serán los aspectos relacionados con la creación, evolución y disolución de la red los que gozarán de una mayor relevancia.

Una VE dinámica puede evolucionar a través de cuatro fases distintas a lo largo de su existencia (Camarinha-Matos y Afsarmanesh, 2001; Do *et al.*, 2000; Dang, 2004; Camarinha-Matos *et al.*, 2005; Hao *et al.*, 2005; etc.): creación, operación, evolución y disolución. Cada una de estas fases consta, a su vez, de varios sub-procesos de decisión importantes que pueden desarrollarse bien secuencial o concurrentemente. La figura 2.1 presenta la vista conceptual de dicho ciclo de vida.



Figura 2.1. Etapas del ciclo de vida de una VE dinámica.

A modo de resumen, el ciclo de vida de la VE dinámica comienza cuando se recibe una orden de un cliente o cuando se identifica una oportunidad de

negocio. Esta fase de *creación* incluye, además de la identificación de la oportunidad de negocio, la selección de los socios, el diseño de la sociedad, etc., y, finalmente, su constitución (Sanz, 2004). La fase de *operación* viene una vez se ha creado la DVE y cubre todas las actividades de la misión actual (Do *et al.*, 2000), es decir, comprende el modo en el que las tareas se descomponen y distribuyen entre los participantes (Dang, 2004).

La fase de *evolución* acontece cuando la estructura de la DVE (los miembros y la distribución de las tareas asignadas) necesita ser modificada. Esto puede ocurrir por diversas razones entre las que se encuentran (Katzy y Miralles, 1999) un fallo en la ejecución del contrato por parte de uno de los socios, algún cambio en el entorno de negocio, etc. Finalmente, la fase de *disolución* tiene lugar cuando se considera que la VE dinámica ya no es eficaz (Dang, 2004) lo que puede deberse a que el servicio que proporciona es obsoleto, ya no se necesita o bien no aporta los beneficios suficientes.

En lo que resta de sección se van a describir los principales aspectos relacionados con cada una de estas fases centrándonos, fundamentalmente, en las etapas de creación y evolución donde la identificación y selección de socios constituye una tarea clave. Pero antes debemos introducir, aunque sea fugazmente, el concepto de entorno de gestación de empresas virtuales (VBE, *VE Breeding Environment*), concepto que trataremos con mayor profundidad en el apartado 3 del presente capítulo.

Como ya sabemos, el tiempo y la cantidad de recursos que se consumen durante el proceso de creación de una empresa virtual (cada vez que surge una nueva oportunidad de colaboración/negocio) representan una merma de la agilidad asociada a este modelo de negocio. La efectividad del proceso depende, en gran medida, de la disponibilidad de información adecuada sobre los socios potenciales, de su nivel de preparación para implicarse en una empresa virtual dinámica y, sobre todo, de la existencia de confianza.

Para solucionar este problema, se ha considerado que el proceso de creación de la empresa virtual tiene lugar en el contexto de lo que en la literatura aparece como entorno de gestación de empresas virtuales (Camarinha-Matos *et al.*, 2005; Afsarmanesh y Camarinha-Matos, 2005), pool community (Do *et al.*, 2000), virtual industrial park (Nayak *et al.*, 2001b), virtual industry cluster (Bremer y Molina, 1999; Rabelo *et al.*, 2000), etc., en definitiva, un conjunto de entidades que desean cooperar para responder ágilmente a las oportunidades de negocio que vayan surgiendo.

El entorno de gestación de empresas virtuales representa el potencial de colaboración y el rango de habilidades disponibles para completar una misión (Do *et al.*, 2000). Esta asociación colaborativa se compone de organizaciones que están preparadas para cooperar y, de este modo, responder rápidamente a oportunidades de negocio (Camarinha-Matos *et al.*, 2005), beneficiándose completamente de los cambios inesperados. En otras palabras, son entidades preparadas para operar ágilmente, siguiendo la definición proporcionada por Goranson (Goranson, 1999).

Tal y como se puede apreciar en la figura 2.2, la creación de los entornos de gestación y la génesis de las VE dinámicas son procesos diferentes que se activan por motivos bien distintos.

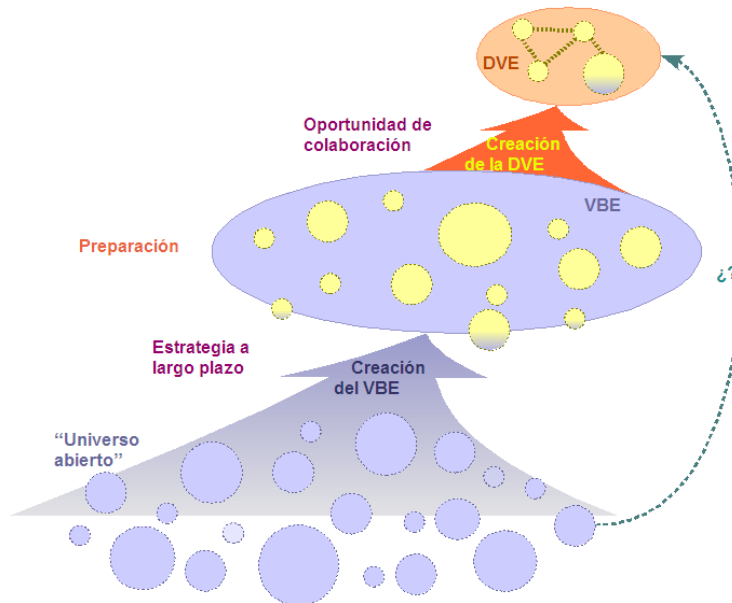


Figura 2.2. Creación de VE dinámicas en el contexto de los VBE (Afsarmanesh, 2005).

Una VBE se crea como una asociación a largo plazo con “bordes limitados” (Camarinha-Matos *et al.*, 2005) donde sus miembros son reclutados de un “universo abierto” de organizaciones de acuerdo con criterios definidos por los creadores y/o administradores del VBE. De este entorno irán surgiendo

DVE como consecuencia de la aparición de oportunidades de negocio/colaboración específicas aunque, en el caso en que exista una falta de habilidades o capacidades dentro del VBE, se pueden reclutar organizaciones no pertenecientes a la misma (Afsarmanesh, 2005b).

Considerando que cada compañía puede tener su lista de “conocidos” fuera del VBE (es decir, otros socios, actuales o recientes, en actividades no vinculadas al VBE), el espacio de búsqueda que vamos a considerar estará constituido, en primera instancia, por los miembros del VBE. A continuación, y con menor probabilidad, por los socios de estas empresas en actividades enmarcadas fuera del VBE (esta opción podría considerarse como una “extensión” de los bordes del VBE). Por último, también se puede recurrir al “universo abierto” de organizaciones, aunque por cuestiones de preparación, predisposición a la cooperación, etc., pero sobre todo de confianza, esta opción debe representar el último recurso.

2.2.1 Creación

Los procesos de decisión principales en la fase de creación son los relacionados con la planificación preliminar, formación del consorcio y lanzamiento de la VE dinámica (Afsarmanesh, 2005):

- *Planificación preliminar*: engloba tanto el proceso de identificación y caracterización de la oportunidad de colaboración (CO, *Colaboration Opportunity*) como la elaboración de un primer borrador de planificación que permita satisfacer las necesidades de la CO.
- *Formación del consorcio*: para conseguir los socios adecuados (competencias, capacidades, servicios, etc.) es necesario llevar a cabo un proceso de búsqueda y selección en el que la negociación jugará un papel crucial.
- *Lanzamiento de la DVE*: para lanzar la DVE es necesario refinar tanto su planificación como sus principios de gobierno. También es preciso formular y modelar los contratos y acuerdos y, finalmente, poner la VE dinámica en operación.

En la figura 2.3 pueden apreciarse los principales pasos involucrados en el proceso de creación, que van desde la identificación y caracterización de la CO hasta el lanzamiento de la DVE. Como puede observarse, los procesos de decisión evolucionan secuencialmente, aunque en ocasiones puede ser

necesario volver a etapas anteriores, sobre todo cuando surgen problemas en la negociación con los socios potenciales identificados (lo que exigiría una nueva selección o una nueva planificación) y cuando no es posible encontrar los mejores socios dada la planificación actual.

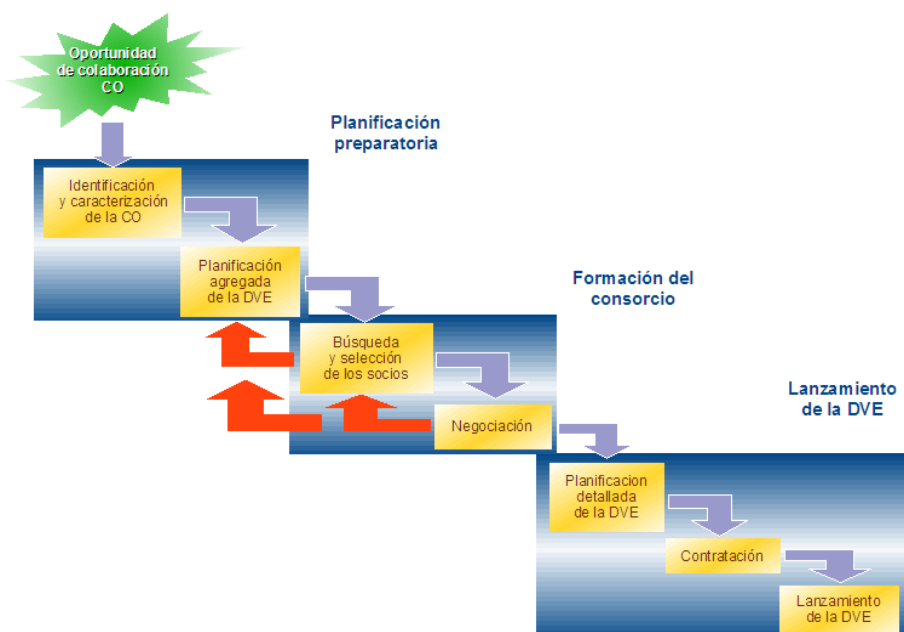


Figura 2.3. Vista simplificada del proceso de creación de una DVE. Adaptada de (Camarinha-Matos *et al.*, 2005).

Dada la importancia que posee la fase de creación para los propósitos de la presente tesis, en los siguientes subapartados vamos a intentar describir y analizar cada una de estas fases con el suficiente grado de detalle que nos permita conocerlas y comprenderlas mejor.

IDENTIFICACIÓN DE LA OPORTUNIDAD DE COLABORACIÓN

Este paso involucra la identificación y caracterización de una nueva oportunidad de negocio que pueda dar paso a la formación de una nueva DVE (Oliveira y Camarinha-Matos, 2005). Dicha CO puede ser externa, originada por un cliente (potencial) y detectada por un elemento del entorno

de gestación que actúa como representante, o puede ser generada internamente, como parte de una estrategia de desarrollo de alguno de los miembros del VBE, tal y como puede ocurrir si una empresa de fabricación desea reducir su tasa de fallos, mejorar su proceso productivo, etc.

Se pueden dar cuatro modalidades principales de colaboración (Afsarmanesh, 2005): (1) modelo de procesos de negocio, (2) de proyecto, (3) de resolución de problemas y (4) modelo *ad-hoc*.

- Un modelo de cooperación basado en *procesos de negocio* puede definirse como un conjunto de actividades heterogéneas normalmente distribuidas en subprocesos interorganizacionales. Estas actividades se pueden clasificar en actividades automáticas, materializadas a través de las tecnologías de la información y la comunicación; actividades interactivas, desempeñadas mediante la unión de trabajo cooperativo e ICT; actividades cooperativas, realizadas mediante trabajo cooperativo; actividades manuales, desempeñadas por las personas; y actividades de toma de decisión, es decir, de elección entre distintas alternativas.
- Un modelo de cooperación basado en *proyectos* constituye la base para la definición de una estructura de trabajo multi-proyecto, así como de la gestión de unos recursos humanos pertenecientes a múltiples organizaciones. En este tipo de modelos de colaboración las actividades están relacionadas mediante fuertes restricciones como son, por ejemplo, una secuencia de relaciones, alternativas y/o ciclos, etc. La ejecución de tales actividades necesita seguir un plan temporal (por ejemplo un GANTT) y que el flujo esté dirigido mediante las relaciones (por ejemplo, en tal modelo una actividad que sigue a otra no puede empezar antes de que termine la previa y hayan finalizado todos sus *outputs*). Por ello, la VE ágil que adopte este modelo precisa una serie de herramientas destinadas al control técnico-financiero, a la integración del trabajo cooperativo y de la gestión de conocimiento, además de aquellas destinadas a dar soporte a la gestión de proyectos.
- En un modelo de cooperación basado en la *resolución de problemas* se establece un itinerario en el que se define tanto la situación actual como el escenario deseado, se realiza un análisis de las diferencias observadas y se identifican los grupos de trabajo. Por tanto, en este modelo la ejecución del proceso no sigue un flujo particular; las únicas restricciones que se consideran son aquellas sobre la fecha

límite de las contribuciones. En este caso, se necesita un mediador que identifique a los colaboradores, y que recoja y evalúe sus contribuciones; un sistema de medida y recompensa de la cooperación que premie a los miembros en función de su contribución a la consecución de los objetivos marcados en el itinerario; un organizador personal del tiempo y del trabajo; herramientas específicas para la resolución de problemas, como por ejemplo herramientas de diagnóstico, razonamiento basado en casos, sistemas expertos y mejora continua.

- Finalmente, encontramos el modelo de cooperación *ad-hoc*. Este modelo puede ser útil cuando grandes organizaciones, no acostumbradas a colaborar estrechamente con otras, necesitan unir sus esfuerzos para dar una rápida respuesta a un requerimiento externo. En este modelo sólo se requieren unas pocas personas encargadas de tomar las decisiones que han de llevar a sus organizaciones hacia un objetivo común.

Entre los aspectos principales a considerar en esta etapa inicial encontramos, entre otros, la identificación y clasificación de la oportunidad de cooperación, el establecimiento de la modalidad de colaboración, la representación de la oportunidad de colaboración, el análisis de la viabilidad de la oportunidad, la interacción con el cliente (creación de un soporte para concretar citas y ofertas,...), etc.

PLANIFICACIÓN AGREGADA

Tras identificar la oportunidad de negocio llega el momento de establecer, a grandes rasgos, una estructura inicial de la DVE potencial, identificándose las capacidades y competencias requeridas, la forma de organización y los roles existentes. En esta etapa es importante definir la forma de la sociedad así como los tipos de relaciones de colaboración, que normalmente estarán reguladas mediante contratos y acuerdos de cooperación.

- Un *contrato* es un pacto o convenio entre dos o más partes en el que se realiza y se acepta una oferta y en el que todas las partes involucradas se benefician. Define los deberes, derechos y obligaciones de las partes, así como todas aquellas cláusulas que son importantes para caracterizar su propósito.

- Un *acuerdo* es una resolución entre varias partes cuyo fin es regular las acciones de cooperación entre los socios (siempre está asociado a un contrato): las formas de comunicación, los procedimientos para informar, la representación de los datos, etc. Un tipo particular de acuerdo es el de *partnership*, que establece las cláusulas que regulan la cooperación de la sociedad a largo plazo.

A continuación se describen, de un modo esquemático, algunas de las formas de organización típicas para los consorcios de colaboración (Camarinha-Matos y Afsarmanesh, 2001):

- **Caso 1** (figura 2.4): *Consortio explícito*.
 - La colaboración está regulada mediante un contrato común con el cliente y un acuerdo de consorcio.
 - Todos los socios se comprometen con el cliente dado que todos firman el contrato.
 - El acuerdo puede establecerse antes del contrato o en ese preciso instante.
 - El cliente se interesa sobre todos los miembros del consorcio.
 - Existen relaciones de cooperación entre los socios.
 - A parte de los compromisos representados por el contrato y el acuerdo, los socios son autónomos.
 - El consorcio se disuelve al final del contrato.

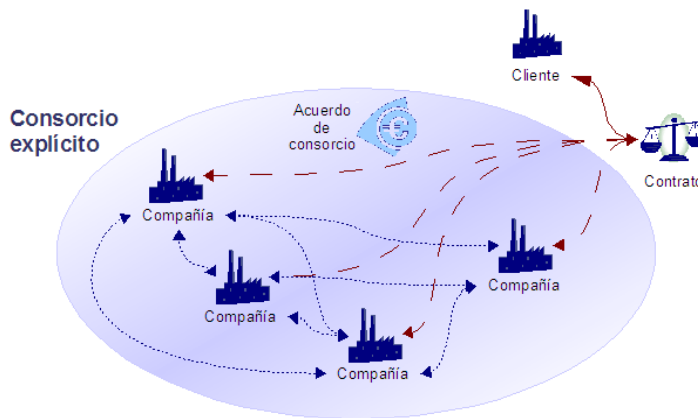


Figura 2.4. En el consorcio explícito todos los socios se comprometen con el cliente ya que todos firman el contrato.

- **Caso 2** (figura 2.5): *Consortio interno*.
 - Existe un contrato entre un representante del consorcio y el cliente.
 - El cliente no tiene porqué saber cómo se organiza el consorcio.
 - El consorcio se formaliza usando un acuerdo y un contrato interno.
 - Solamente un socio (el que firma el contrato) se compromete con el cliente. El resto de los socios sólo se comprometen con aquel que firma el contrato o con otros socios ya comprometidos.
 - Existen relaciones de cooperación entre los socios.
 - A parte de los compromisos representados por el contrato y el acuerdo, los socios son autónomos.
 - El consorcio se disuelve al final del contrato.

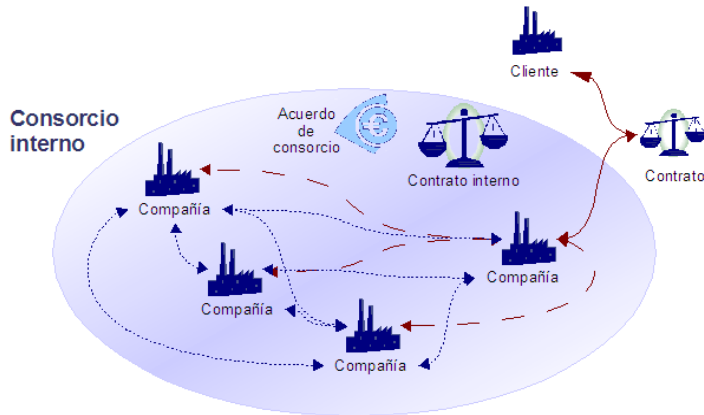


Figura 2.5. Cuando el consorcio es interno sólo existe un contrato entre un representante del consorcio y el cliente, mientras que el resto de los socios sólo se comprometen con aquel que firma el contrato.

- **Caso 3** (figura 2.6): *Subcontratación*.
 - Hay un contrato entre un socio y un cliente y subcontratos entre este socio y los otros proveedores de servicios/productos.
 - El cliente no tiene por qué conocer la manera en que se organiza el socio contratado.
 - Sólo la compañía contratada se compromete con el cliente.
 - La compañía contratada establece todos los subcontratos que son necesarios para llevar a cabo su contrato.

- No existe la necesidad de cooperación entre los socios contratados.
- A parte de los compromisos representados por el contrato y el acuerdo, los socios son autónomos.
- Puede existir una relación de cooperación a largo plazo, aunque no es obligatorio.

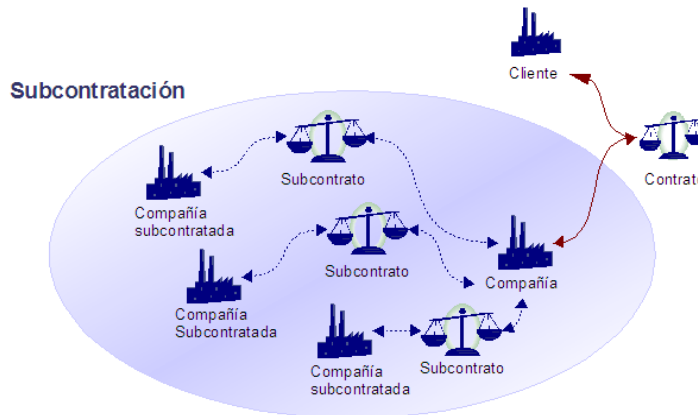


Figura 2.6. Subcontratación: el cliente firma un contrato con un socio, que a su vez tiene establecidos subcontratos con los otros proveedores de servicios/productos.

- **Caso 4** (figura 2.7): *Partnership*.
 - La sociedad crea una entidad (nueva compañía) usando un acuerdo de *partnership*.
 - La nueva compañía establece un contrato con el cliente y, por tanto, solamente esa nueva compañía es la que se compromete con el cliente.
 - La sociedad puede continuar incluso después de la finalización del contrato inicial.

Tal y como pone de manifiesto Camarinha-Matos *et al.* (Camarinha-Matos *et al.*, 2005), los casos 1 y 2 son las formas de organización más adecuadas para las empresas virtuales dinámicas, mientras que las formas 3 y 4 se ajustan mejor a relaciones de colaboración a medio y largo plazo, lo cual no supone que no puedan emplearse en modelos de cooperación dinámicos.

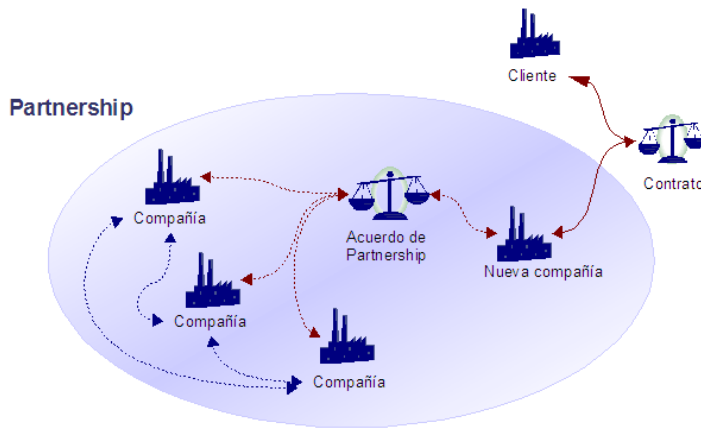


Figura 2.7. La sociedad crea una nueva compañía usando un acuerdo de partnership, de forma que solamente esa nueva compañía es la que se compromete con el cliente a través de un contrato.

Además de la forma que va a tener la sociedad, en esta etapa es necesario considerar otros aspectos como son (Oliveira y Camarinha-Matos, 2005): la identificación de las capacidades y competencias necesarias, la estructura inicial de la empresa virtual dinámica (topología, duración, niveles de visibilidad, etc.) y de los roles asociados, las macro reglas de gestión, la representación del modelo inicial de la empresa virtual dinámica (representación *top-down* frente a un enfoque *bottom-up*), las simulaciones para establecer diferentes configuraciones, etc.

BÚSQUEDA Y SELECCIÓN DE LOS SOCIOS

El proceso de búsqueda y selección de los socios es, en opinión de numerosos autores (Meade *et al.*, 1997; Mikhailov, 2002; Lau y Wong, 2001; Ouzounis, 2001; Camarinha-Matos y Afsarmanesh., 2001; Reid *et al.*, 1996; Goldman, 1995; Petersen, 2003; Fischer *et al.*, 2004; ...), el factor clave en el éxito de la empresa virtual dinámica.

Se trata de una tarea que consume grandes cantidades de tiempo y recursos de las empresas (Reid *et al.*, 1996; Goldman, 1995; Camarinha-Matos *et al.*, 2005), lo que se traduce en una pérdida significativa de su agilidad y

rentabilidad. Es por ello, quizás, que la búsqueda y selección de los socios se ha convertido en uno de los ámbitos de la DVE en los que más se ha investigado en los últimos años.

Como su propio nombre indica, esta etapa se orienta hacia la identificación de los socios potenciales, así como su valoración y posterior selección¹⁴. Esta selección se realiza principalmente dentro del entorno de gestación, aunque también se pueden reclutar del exterior cuando no existan las habilidades o capacidades necesarias. Considerando que cada compañía puede tener su propia lista de “conocidos” fuera del VBE¹⁵, el espacio de búsqueda está formado en primer lugar por los miembros del VBE, después por los “conocidos” y finalmente por un “universo abierto” de organizaciones.

En esta etapa se deben considerar diferentes aspectos, entre los que destacan (Oliveira y Camarinha-Matos, 2005): los elementos necesarios para la búsqueda y selección de los socios (técnicos, económicos, indicadores de confianza, preferencias, etcétera.), algoritmos de emparejamiento que ligen la oferta y la demanda de competencias y habilidades, criterios de selección (principalmente multi-criterio), optimización, evaluación (estado de preparación, infraestructuras, etc.), consideración de la historia de la colaboración, búsqueda externa (si la oferta interna es escasa), etc.

Se puede considerar un gran número de estrategias y algoritmos para apoyar esta tarea. Por ejemplo, una solución habitual consiste en seguir un acercamiento de arriba hacia abajo o *top-down* (enfoque planificador) en el que el planificador diseña la VE dinámica, decide los roles y selecciona los socios que mejor se ajusten a su plan. Un alternativa diferente se halla en la utilización de un enfoque de abajo a arriba o *bottom-up* (enfoque competidor). En este caso, el *broker* o planificador anuncia la oportunidad de colaboración, espera hasta que se formen algunos consorcios de forma espontánea (por iniciativa de algunos miembros del VBE), y después analiza sus ofertas globales.

Sin embargo, la selección de socios no sólo es importante en la fase de creación. Durante el funcionamiento normal de la VE dinámica puede que sea necesario, por ejemplo, encontrar algún nuevo socio/suministrador para

14 En dicha elección juega un papel muy importante la negociación entre los distintos agentes involucrados, aspecto que será analizado brevemente en el siguiente apartado.

15 Socios habituales, o incluso socios anteriores, que están fuera de los bordes del VBE

un servicio o componente particular que ningún otro socio está ofreciendo o pueda ofrecer, o incluso puede ser preciso reemplazar a un miembro poco competente o incapaz de cumplir los compromisos adquiridos, etc. Estas situaciones dan lugar a lo que se conoce como evolución de la empresa virtual dinámica, aspecto que trataremos más adelante.

NEGOCIACIÓN

La fase de negociación es un proceso iterativo encaminado a la consecución de acuerdos de forma que se alineen las necesidades y las ofertas (Ouzounis, 2001). Constituye, por tanto, una de las herramientas necesarias para que la VE dinámica pueda proporcionar respuestas eficientes a las distintas oportunidades de negocio. Puede ser visto como un proceso complementario al de selección de socios y puede requerir volver al paso anterior si no se encuentra una solución adecuada con la configuración de socios actual.

Los aspectos importantes a considerar en esta etapa son (Camarinha-Matos *et al.*, 2005): la determinación de los objetos de negociación, los protocolos de negociación, el proceso de toma de decisión (junto con los parámetros correspondientes), la representación de los acuerdos, etc. También puede considerarse el refinamiento y asignación de los procesos de negocio, aunque suele ser habitual realizarla mientras se lleva a cabo la planificación detallada de la VE dinámica.

Por todo ello, la negociación debe ser vista como un proceso más general que ha de darse en varias etapas de la creación de la VE ágil: negociación con el cliente, negociación con los socios potenciales, negociación de los detalles con los socios seleccionados, etc.

PLANIFICACIÓN DETALLADA DE LA DVE

Una vez se han seleccionado los socios mejor preparados y se ha alcanzado un acuerdo de colaboración, llega el momento de concretar el plan de la empresa virtual dinámica así como sus principios de gestión. Por ello, en esta etapa han de detallarse los procedimientos necesarios para realizar la misión global de la DVE, lo que incluye (Petersen *et al.*, 2002) tanto los requisitos funcionales y de actuación del consorcio como las necesidades de información y de recursos.

También es necesario modelar el proceso de negocio (que depende del tipo de colaboración), diseñar el producto/servicio junto con los flujos de materiales e información, planear cualquier sistema de control necesario (calidad, coste, etc.), desarrollar una representación final de la VE dinámica; asignar los roles y las responsabilidades, y definir tanto los principios compartidos como los niveles de acceso (capital, recursos, beneficios,...), políticas de funcionamiento básicas, etc.

CONTRATACIÓN

Una vez se ha llevado a cabo la planificación detallada de la cooperación, se debe establecer el contexto jurídico en el que se enmarcarán los acuerdos contractuales entre los miembros de la VE dinámica (es decir, la formulación y modelado de los contratos y de los acuerdos, así como el proceso de contratación en sí mismo) antes de que la DVE pueda ser lanzada con éxito.

En el contrato se definen los deberes, derechos y obligaciones de las partes involucradas, así como todas aquellas cláusulas que sean importantes para caracterizar el propósito del contrato (Camarinha-Matos y Afsarmanesh, 2001). Un acuerdo, por su parte, tiene como meta regular las acciones de la cooperación entre los socios (es un arreglo entre las partes con respecto a un método de acción), y siempre se asocia a un contrato (Strader *et al.*, 1998).

Además de la definición de los tipos de contratos y de su representación, se necesita tratar (Afsarmanesh, 2005) el proceso de contratación, los mecanismos e instituciones de ejecución del contrato, las cuestiones legales, etc. Obviamente, este paso ha de realizarse en paralelo a la negociación y planificación detallada de la DVE (Kanet *et al.*, 1999).

LANZAMIENTO DE LA DVE

Esta es la última fase del proceso de creación, es decir, cuando la empresa virtual dinámica se “pone en marcha”. Es responsable de tareas tales como la configuración de la infraestructura de información y comunicación, materialización de los distintos espacios de colaboración, asignación y establecimiento de los recursos, activación de los servicios, notificación de los miembros implicados, etc., además del registro de la nueva VE dinámica en el entorno de gestación.

2.2.2 Operación

La fase de operación exige, generalmente, cuatro procesos de decisión diferentes (Strader *et al.*, 1998): la gestión financiera, la fabricación, la comercialización y la distribución. El *input* de este sistema de procesos lo constituye toda la información relacionada con la oportunidad de mercado, con los socios de la alianza y con el diseño de la DVE recogidos durante la fase de creación. El *output* de estos procesos es un resumen de todas las actividades y transacciones que se producen durante la operación de la VE dinámica.

En contraste con la mayoría de los procesos de la fase de creación, los procesos de decisión en la fase de operación no son secuenciales. Cada uno de estos procesos de decisión cuenta con las entradas y salidas de los otros (ingeniería concurrente), tal y como aparece reflejado en la figura 2.8. Esto hace que ésta sea una de las fases del ciclo de vida más difícil de gobernar.

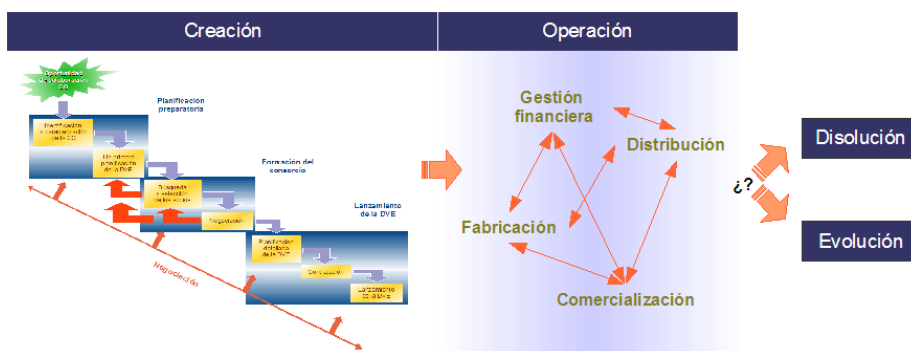


Figura 2.8. La fase de operación y su relación con las otras fases del ciclo de vida de la DVE.

En este contexto, tecnologías tales como Peer-to-Peer (P2P), Web Services, workflow, Semantic Web, agentes inteligentes, Grid, etc., son algunas de las soluciones empleadas para garantizar la integración e interoperabilidad, tanto a nivel inter como intraorganizacional, dinamizando las transacciones y soportando la coordinación de los procesos (Chituc y Azevedo, 2005). Sin embargo, la mayoría de las soluciones ICT halladas en la literatura todavía son bastante engorrosas, carecen de la flexibilidad necesaria y suelen ser específicas del problema considerado, lo que hace necesario una mayor investigación en este campo.

La fase operativa termina una vez que la oportunidad de mercado ha pasado (Strader *et al.*, 1998) o cuando se ha prestado el servicio requerido por el cliente; en otras palabras, la empresa virtual dinámica se disuelve una vez el proyecto se ha completado (Lau y Wong, 2001). Cuando ocurre esto, puede dar comienzo la disolución de la empresa virtual dinámica.

2.2.3 Disolución

La DVE es una organización temporal donde las relaciones entre los socios solamente se dan dentro de una misión específica (Mikhailov, 2002). Normalmente, se configura con el objeto de fabricar un tipo particular de producto o prestar un determinado servicio que permita aprovechar una oportunidad de negocio (la necesidad de un cliente, un nicho de mercado, etc.). Cuando el mercado de ese producto desciende, ya no proporciona los beneficios necesarios, etc., o simplemente se ha satisfecho la necesidad del cliente, entonces la DVE se disuelve fluidamente, permitiendo que los socios encuentren nuevos retos (Dang, 2005).

El *input* de este proceso es toda la información legal y contable requerida para finiquitar los contratos y para repartir cualquier activo de la sociedad entre las empresas que componen la organización (Strader *et al.*, 1998). Con esta información se procede, de forma secuencial, a la finalización de la operación y a la distribución del activo, que son los procesos de decisión principales en la fase de disolución. También se debe acordar el servicio postventa y la atención al cliente (Mikhailov, 2002). Una vez se han concretado todos estos aspectos puede dar comienzo la dispersión del activo entre los socios.

Tras completarse los procesos de finalización y distribución, las empresas son libres para perseguir nuevas oportunidades y para formar otras sociedades¹⁶. Esto significa, esencialmente, que esta asociación virtual está muerta, que su ciclo de vida ha finalizado. Sin embargo, y como habrá podido observarse, el ciclo de vida hasta aquí descrito es un proceso bastante “ideal”, en el que todavía no se ha considerado un aspecto clave en el logro de la agilidad propia de este concepto: su posibilidad de evolucionar y reconfigurarse.

16 Si existía exclusividad dentro de la VE dinámica.

2.2.4 Evolución

Durante la operación normal de una VE dinámica puede ser necesaria su evolución o reconfiguración bien porque se necesite añadir un nuevo socio, o reemplazar a uno existente, o cambiar sus roles, etc. En concreto, esta necesidad puede deberse a algunos eventos excepcionales, tales como (Katzy y Miralles, 1999) la incapacidad de un socio para cumplir con el contrato, el cambio en las metas de negocio, la omisión de los compromisos o los plazos por parte de algún socio, cambios en el entorno de negocio, etc.

Claramente, la fase de reconfiguración está contenida en la de formación, por lo que esta fase suele requerir funcionalidades similares a las ya especificadas en el período de formación (Davidrajah y Deng, 2000), eso sí, con ciertas peculiaridades. Así, cuando se precisa cambiar alguno de los socios la evolución de la red conlleva, simplemente, la identificación y selección de los nuevos socios (incluida la negociación y el establecimiento de contratos), mientras que si es necesario un cambio de roles o el rediseño de la DVE, etc., entonces convendrá pasar por todas las fases definidas en la creación salvo, claro está, la de identificación de una oportunidad de negocio.

2.3 LOS ENTORNOS DE GESTACIÓN

Como ya se ha subrayado anteriormente, la capacidad para formarse y adaptarse rápidamente a las condiciones cambiantes de los mercados hace de las empresas virtuales dinámicas un medio aventajado para hacer frente a los desafíos que plantean aquellos mercados que, como los actuales, se caracterizan por una enorme turbulencia. A su vez, el éxito de la DVE está profundamente unido al proceso de búsqueda y selección de aquellos socios que mejor se adapten a una oportunidad de negocio específica.

Existen numerosos trabajos de investigación vinculados al paradigma de la VE en los que se asume que los socios potenciales pueden identificarse y seleccionarse de un universo abierto de empresas disponibles. Esta suposición, que desde un punto de vista teórico sería la ideal, es poco realista al pasar por alto un gran número de cuestiones (Oliveira y Camarinha-Matos, 2005; Sanz *et al.*, 2007):

- ¿cómo se puede conocer la existencia de posibles socios potenciales en ese universo abierto?,

- aún conociendo su existencia, ¿cómo podemos adquirir los perfiles básicos de información sobre ellas si no existen plantillas estándar que la aporten?,
- suponiendo que se dispone de toda esta información, ¿cómo podemos estar seguros de que las empresas están dispuestas a colaborar?,
- aún sabiendo que poseen la intencionalidad adecuada, ¿cómo se puede constituir con celeridad una infraestructura de colaboración interoperable si las entidades muestran una elevada heterogeneidad y presentan una gran diversidad de sistemas de interacción?,
- aún habiendo subsanado estos problemas, ¿cómo definiremos rápidamente los acuerdos sobre roles, responsabilidades de cada socio, derechos sobre el resultado final, beneficios, etc.?,
- ...,
- pero por encima de todo esta el tema de la confianza, que es, sin duda alguna, la base para cualquier colaboración: ¿cómo se puede construir el nivel de confianza necesario entre las organizaciones?.

La formación de una DVE, y en general de cualquier red colaborativa, depende, por tanto, de la disponibilidad de información adecuada sobre los socios potenciales, de su nivel de preparación para implicarse, de que sus miembros compartan ciertas metas comunes, de que exista un cierto grado de confianza mutua, de que se hayan establecido infraestructuras informáticas comunes con las que poder interoperar, de que se hayan acordado ciertas prácticas de negocio y reglas operativas comunes, etc. En definitiva, de que los socios potenciales estén preparados para participar en tal colaboración.

En un primer intento por afrontar algunos de estos retos, y quizás influenciados por los desarrollos en el área del comercio electrónico (*e-commerce*), algunos autores han empezado a considerar mercados de empresas como fuente de socios potenciales. Estos mercados proporcionan medios para encontrar organizaciones (a través de directorios y guías de mercado) y algunos perfiles normalizados, e incluso referencias (mínimas) sobre dichas organizaciones (Sanz *et al.*, 2005b). Este es, precisamente, el enfoque adoptado en algunos sistemas de selección de socios recientes basados en agentes (Dang, 2004; Petersen, 2003; Ouzounis, 2001; Rocha y Oliveira, 1999; Shen y Norrie, 1998; etc.).

Sin embargo, cuando la meta es encontrar socios para una colaboración, la idea de un “mercado virtual de empresas” no satisface todos los requerimientos necesarios, es decir, no garantiza que los socios potenciales

estén preparados para participar en una red colaborativa. Por tanto, los mercados virtuales no pueden, no deben, ser la respuesta si queremos garantizar unos niveles de agilidad e integrabilidad elevados que permitan hacer frente a los nuevos retos empresariales.

Por otro lado, podemos darnos cuenta fácilmente que, si bien garantizar este conjunto de exigencias básicas puede llegar a suponer un reto difícil de alcanzar cuando se consideran relaciones de cooperación temporales (Caso B de la figura 2.9), la situación no es tan crítica cuando se tiene en cuenta procesos de colaboración a largo plazo que no están limitados a una única oportunidad de negocio. Este es el caso, por ejemplo, de las empresas extendidas (caso A de la figura 2.9), donde se comparten infraestructuras, metas, recursos, existen altos niveles de confianza, etc. En este tipo de relaciones los costes y el tiempo de preparación de la colaboración son afrontables dadas las perspectivas a largo plazo.

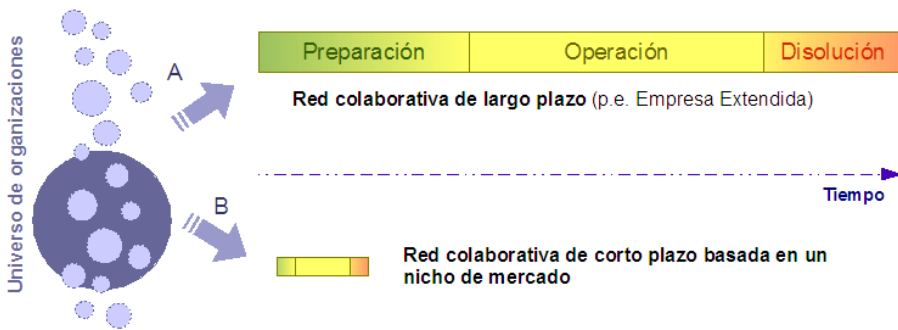


Figura 2.9. Diferencia temporal entre VE con relaciones a corto y largo plazo.

Teniendo presente esta realidad, recientemente han surgido algunos experimentos que intentan combinar las ventajas propias de las DVE con las derivadas de las relaciones a largo plazo. Mediante estas experiencias se pretende crear un club de organizaciones que estén preparadas para colaborar a largo plazo y del que surjan coaliciones temporales (constituidas por diferentes subconjuntos de estas organizaciones que se unen dinámicamente) que respondan a las distintas oportunidades de negocio. Estas nuevas formas organizacionales se conocen con el nombre de entornos de gestación de VE (caso C en la figura 2.10) y surgen como una evolución de los *clusters* y distritos industriales (Bremer *et al.*, 1999; Mejía y Molina, 2002).

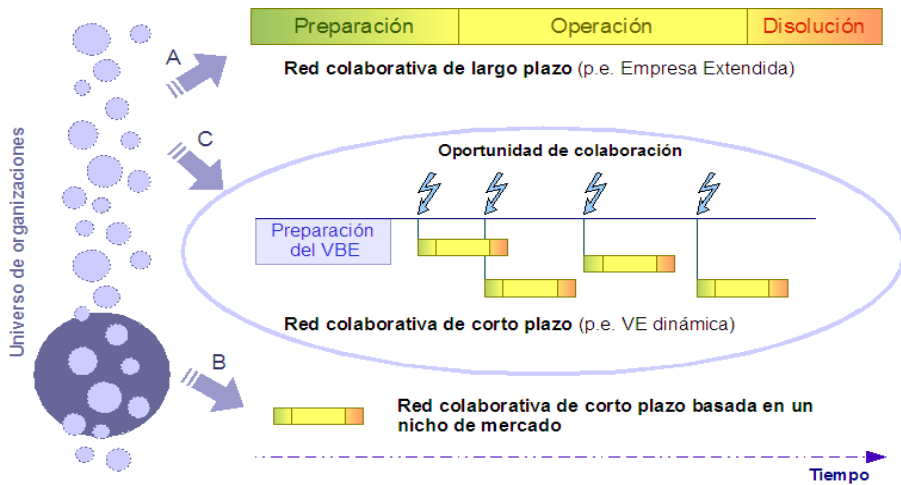


Figura 2.10. Formación de empresas virtuales en diferentes contextos. Adaptada de (Afsarmanesh, 2005).

2.3.1 Definición

Un entorno de gestación de empresas virtuales puede definirse como (Camarinha-Matos *et al.*, 2005c):

“Una asociación de organizaciones, y sus instituciones de soporte afines, unidas mediante un acuerdo de cooperación a largo plazo por el que adoptan principios e infraestructuras de operación comunes, y cuya principal meta es incrementar tanto sus oportunidades como su preparación hacia la colaboración en organizaciones virtuales”.

Los VBE aplican infraestructuras de información y comunicación efectivas que suministran una base común con la que lograr niveles de colaboración e interacción apropiados entre sus miembros, facilitan la configuración y establecimiento de VE ágiles además de ayudar en su operación, introducen mecanismos para construir confianza, definen una cultura económica basada en la cooperación estableciendo unos principios y valores comunes entre las diferentes organizaciones (independientemente de donde se encuentren), etc.

En definitiva, el propósito principal de su existencia es la creación eficiente de empresas virtuales dinámicas que respondan ágilmente a las oportunidades de negocio. Estas oportunidades de negocio pueden englobarse en dos tipos: aquellas que pueden ser *identificadas* en el mercado (por ejemplo, cuando aparece un nuevo nicho de mercado o un cliente acude a uno de los miembros del VBE en busca de un producto o la prestación de un servicio) y aquellas que pueden ser *creadas* por el VBE como fruto de la innovación (por ejemplo, cuando una empresa de fabricación desea reducir su tasa de fallos o mejorar su proceso de producción).

2.3.2 Tipos

Atendiendo a la ubicación geográfica de sus componentes, se pueden identificar dos tipos de VBE: los regionales y los globales (Afsarmanesh y Camarinha-Matos, 2005). Los VBE regionales involucran, principalmente, empresas de una región geográfica, sin importar su tamaño o capacidad de recursos; la ventaja de este tipo de entornos reside en que las empresas que lo componen poseen la misma cultura de negocio y, además, el sentimiento de comunidad. Por su parte, los VBE globales suelen englobar organizaciones y empresas geográficamente distribuidas. La ventaja de este tipo de entornos reside, en este caso, en la posibilidad de encontrar las entidades mejor preparadas a nivel mundial.

Tanto los entornos de gestación de empresas virtuales regionales como los globales pueden estar especializados bien en una única área de interés (*single-sector*) o bien pueden cubrir varias áreas (*multi-sector*). Considerando que los entornos de gestación globales son los que ofrecen, sin ninguna duda, posibilidades más atractivas para la formación de empresas virtuales dinámicas/ágiles, a lo largo de nuestra investigación consideraremos únicamente estos últimos.

2.3.3 Ventajas

Constituir estos entornos para la formación de empresas virtuales ágiles ofrece una serie de ventajas, entre las que destacan las siguientes (Galeano *et al.*, 2005; Afsarmanesh y Camarinha-Matos, 2005; Afsarmanesh, 2005):

1. Agiliza el proceso de creación de DVE basadas en oportunidades específicas de negocio. Esto se logra gracias a que permite reducir tanto la complejidad del proceso como los esfuerzos requeridos, a un aumento de la flexibilidad al soportar la reconfiguración de la DVE, a una mejora de la eficiencia en costes, etc.
2. Reduce el tiempo y el coste de encontrar los socios adecuados. Esto es especialmente significativo, no sólo en la creación de la DVE, sino durante la reconfiguración, ya que permite reducir el riesgo de grandes pérdidas ocasionadas por el fracaso de alguna de las entidades involucradas.
3. Proporciona infraestructuras ICT efectivas para sus miembros, estableciendo una base común para lograr la interoperabilidad y cooperación necesarias.
4. El entorno de gestación posee una bolsa de activos que aporta propiedades interesantes a sus miembros: herramientas software, *know-how*, información y conocimiento general, etc.
5. Suministra mecanismos, guías y servicios de asistencia para motivar y facilitar la configuración y el establecimiento de la DVE: crea sistemas de incentivos, servicios para la búsqueda de socios, negociación de contratos, etc.
6. Permite un gestión proactiva de las competencias y los recursos disponibles dentro del entorno de gestación, asegurando su cobertura.
7. Provee un servicio de mantenimiento y asesoramiento a los miembros del entorno de gestación a través de sus instituciones de soporte: seguros, formación, calificación, etc.
8. Introduce mecanismos para generar confianza entre los miembros: mediante la grabación del historial de rendimiento, a través de criterios de valoración de la confianza de las organizaciones, etc.
9. Facilita guías generales de colaboración estableciendo reglas de conducta, principios de trabajo y cooperación, sistemas de valoración, cultura y ética de colaboración, protección de datos, etc.
10. Incrementa las oportunidades de sus miembros de participar en DVE gracias a la incorporación de su perfil en el catálogo del VBE, donde se incluyen sus competencias, productos, servicios, etc.
11. Disminuye el riesgo que adquieren los promotores de la DVE ya que reduce el tiempo, coste y esfuerzo empleado en su creación y reconfiguración, facilita la disponibilidad de una amplia variedad de competencias y recursos, aporta indicadores del nivel de confianza y rendimiento pasado de los miembros del entorno de gestación, etc.

2.3.4 Entorno de gestación versus universo abierto

Un VBE tiene por objeto mejorar el nivel de preparación de sus miembros en lo concerniente a la formación y operación de futuras DVE, proporcionando un lecho sobre el que establecer, de forma ágil y dinámica, las posibles redes de colaboración. Esta realidad provoca, tal y como aparece representado en la figura 2.11, que sea mucho más rápido, mucho más eficaz y bastante menos costoso, erigir una DVE en el contexto de un VBE (rama 1b) que a través de una búsqueda generalizada de socios (representada por la rama 2).

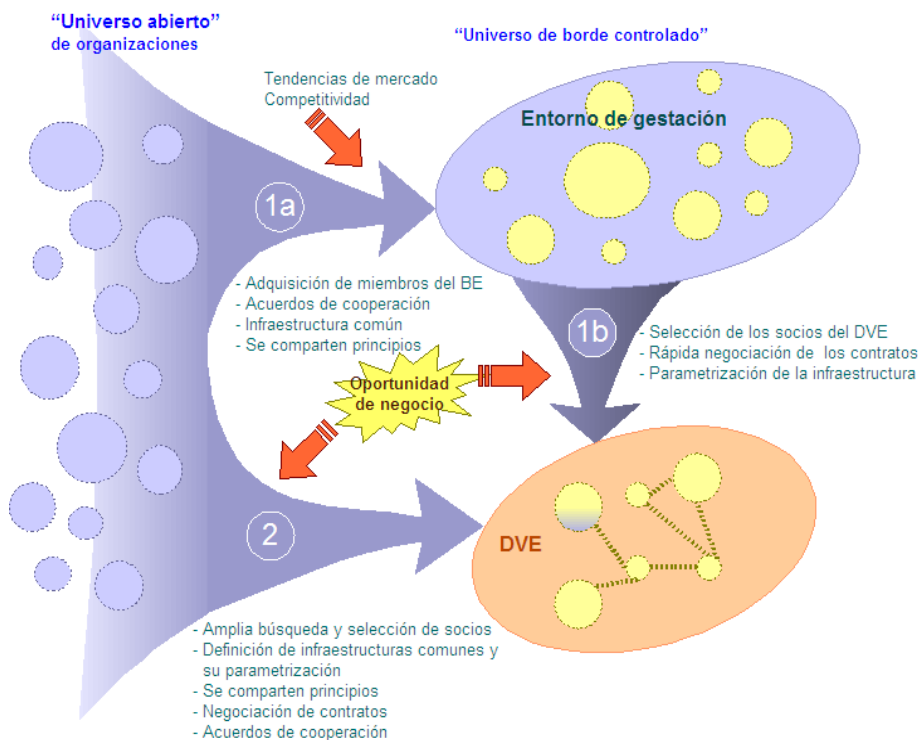


Figura 2.11. Dos posibles alternativas en la formación de VE dinámicas. Adaptado de (Afsarmanesh, 2005).

Para lograr incrementar el nivel de preparación de los miembros del entorno de gestación, y así asegurar su participación en potenciales procesos colaborativos, dicho entorno debe ser una *asociación de "borde controlado"*, lo que *no* presupone que éste deba ser *cerrado*. En otras palabras, en

cualquier momento pueden unirse nuevos miembros a la asociación siempre que obedezcan sus principios generales de funcionamiento. La figura 2.12 permite comprender mejor esta idea.

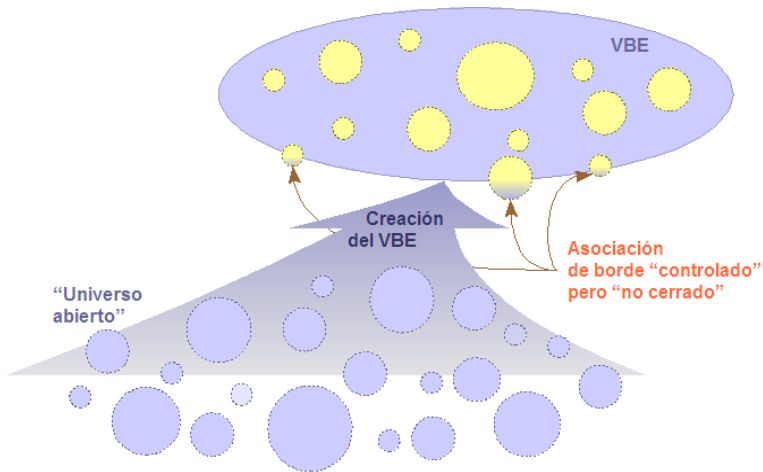


Figura 2.12. Un VBE es una asociación de borde controlado pero no cerrado.

Por tanto, los miembros del entorno de gestación deben obedecer las reglas generales y las políticas que en él se definan (como, por ejemplo, adoptar una infraestructura ICT común). Al mismo tiempo, estas organizaciones acceden y se benefician de múltiples elementos: información, servicios y herramientas comunes que constituyen la bolsa de activos del VBE; mercados y canales de distribución comunes; recursos y factores de trabajo comunes; medios para compartir el coste de nuevos proyectos; lazos culturales comunes; recursos para compartir su experiencia y *know-how*; etc.

Dentro de un VBE pueden darse distintos grados de afiliación y, asociados a cada uno de ellos, diferentes derechos y responsabilidades de los miembros¹⁷. En este sentido, los componentes de un VBE pueden situarse en un rango que va desde los que tienen una baja afiliación hasta los altamente asociados, que se situarán en el extremo opuesto del espectro (Afsarmanesh, 2005):

17 Afsarmanesh (Afsarmanesh, 2005) establece los derechos y requisitos de los roles asociados tanto a la existencia de los VBE como a la de las diferentes DVE que puedan surgir durante su actividad.

- Un ejemplo de un miembro débilmente asociado podría ser el de una entidad que deseara participar circunstancialmente en una DVE. En este caso, necesitaría un nivel de preparación elemental y tan sólo debería hacer pública una cantidad de información mínima (por ejemplo, la relativa a sus actividades relacionadas con la empresa virtual dinámica).
- Por contra, un miembro totalmente activo normalmente contribuye a su promoción y crecimiento, así como al enriquecimiento de su bolsa de activos (también puede tener un papel activo en el brokerage y planificación de la DVE). Además, es posible que esté involucrado en la expansión de la VBE en nuevos sectores.

Cuando se inicia una VE dinámica dentro de un entorno de gestación la prioridad en la participación la tienen, obviamente, los miembros del VBE. Sin embargo, el número de habilidades y competencias necesarias en cualquier proceso de producción de un bien, o en la prestación de un servicio, es muy elevado y no siempre es posible encontrarlas dentro del VBE. En estas situaciones, y en muchas otras, es preciso encontrar organizaciones externas que suplan las carencias.

Pero antes de que estas entidades lleguen a convertirse en socios de la DVE, es preciso asegurar que se cubre el nivel de conformidad mínimo exigido dentro del entorno de gestación (por ejemplo, en lo relativo a la infraestructura ICT). Por este motivo, debe invitarse a estas entidades a unirse al VBE, al menos en el nivel más inconexo (por ejemplo, como un miembro débilmente asociado). En ocasiones, puede ser incluso deseable que los nombres de estas entidades no se publiquen como miembros del entorno de gestación¹⁸.

Además de las empresas, un entorno de gestación puede incluir otras clases de organizaciones (institutos de consultoría/investigación, organizaciones gubernamentales, etc.) e incluso trabajadores autónomos que representan, a fin de cuentas, una organización pequeña compuesta por una única persona. Del mismo modo, puede contener algunas instituciones de apoyo (Oliveira y Camarinha-Matos, 2005) que proporcionen a los miembros del VBE algún servicio especializado: servicios legales, técnicas de comercialización, servicios de aseguramiento, cursos de formación, etc.

18 Este podría ser el caso, por ejemplo, de una organización externa que se asocie temporalmente con el VBE con la intención de formar parte de una única DVE concreta.

2.3.5 El ciclo de vida

El ciclo de vida del VBE representa todas las etapas por las que puede evolucionar. Al igual que en el ciclo de vida de una DVE, la existencia de un VBE comienza con la etapa de creación y finaliza con la de disolución, y entre ambas se encuentra la fase de operación y su posible evolución. No obstante, el ciclo de vida de un entorno de gestación tiene sus peculiaridades:

- La etapa de creación se puede dividir en dos fases, (i) iniciación y reclutamiento, que se ocupa de la planificación estratégica y de la incubación inicial del VBE, y (ii) la fase de fundación, que abarca la constitución e inicio de este tipo de asociación.
- Por otro lado, considerando que un VBE es una alianza a largo plazo y que su valiosa bolsa de activos se actualiza gradualmente, su disolución suele ser muy inusual. En su lugar, es mucho más probable que el entorno de gestación avance a través de otra etapa, que algunos autores denominan etapa de metamorfosis, donde puede cambiar no sólo su forma sino también su propósito.

Por tanto, el ciclo de vida de los entornos de gestación de empresas virtuales dinámicas incluye las siguientes fases: creación del entorno de gestación (integrada por la iniciación/reclutamiento y fundación), operación, evolución del entorno de gestación, metamorfosis y disolución. Las figuras 2.13 y 2.14 permiten apreciar, respectivamente, la vista conceptual y temporal de este conjunto de etapas.



Figura 2.13. Etapas del ciclo de vida de un entorno de gestación de DVE (Afsarmanesh y Camarinha-Matos, 2005).



Figura 2.14. Evolución temporal de un VBE. Adaptado de (Afsarmanesh, 2005).

Por último, y dado que se considera que los entornos de gestación son el contexto necesario para poder configurar y establecer empresas virtuales dinámicas de un modo eficiente, se puede afirmar que su creación (es decir, el primer paso del ciclo de vida de las VE dinámicas) se da dentro de la fase operativa de los entornos de gestación. Esta idea puede comprenderse mejor si se contempla la figura 2.15.

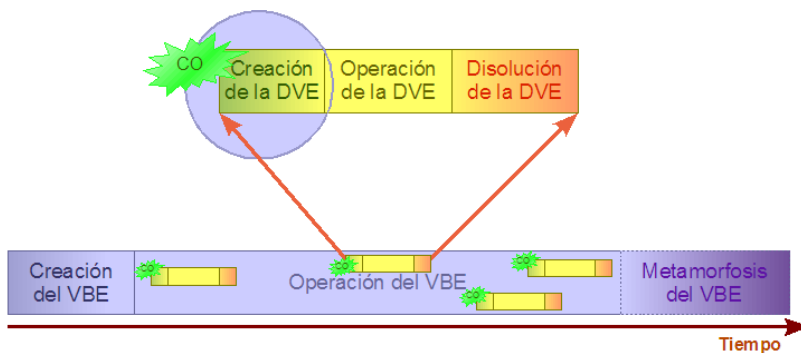


Figura 2.15. Las VE dinámicas surgen durante la operación de un VBE. Adaptada de (Camarinha-Matos *et al.*, 2005c).

2.3.6 Algunos ejemplos

En la literatura revisada (Afsarmanesh y Camarinha-Matos, 2005; Galeano *et al.*, 2005; Jarimo, 2006; Afsarmanesh, 2005; etc.) existen ejemplos de entornos de gestación exitosos, algunos de los cuales se han recogido en la tabla 2.1.

<i>VBE</i>	<i>Miembros</i>	<i>Ubicación</i>	<i>Dominio de aplicación</i>
<i>Virtuelle Fabrik</i>	100	Suiza, Alemania	Industria mecánica
<i>Virtec</i>	>9	Brasil	Fabricación
<i>CEFAMOL</i>	136	Portugal	Moldes de plástico
<i>VE Networks Yorkshire</i>	>25	Reino Unido	ICT, maquinaria, biotecnología.
<i>Virtual Biotech Company</i>	>150	Alemania	Biotecnología
<i>PVC</i>	45	Australia	Plásticos
<i>Fenix Cluster</i>	>250	México	Electrónica, metal y plástico
<i>Biotechnology cluster</i>	411	USA	Biotecnología
<i>Helsinki ICT cluster</i>	79	Finlandia	Telecomunicaciones
<i>CARPI</i>	2068	Italia	Textil

Tabla 2.1. Ejemplos significativos de entornos de gestación de VE a nivel internacional.

Tal y como se puede apreciar, los entornos de gestación de VE existentes sirven a sectores específicos, tienen propósitos particulares y están restringidos a unas regiones concretas. Por este motivo, es necesario que en el futuro los diferentes entornos de gestación, tanto regionales como globales, interactúen entre sí permitiendo la configuración de lo que hemos venido a denominar macroentornos de gestación (*MVBE, Macro VE Breeding Environment*) globales que soporten la creación eficiente de las futuras VE dinámicas. Sólo de esta forma se lograrán las condiciones ideales para la formación de VE dinámicas que estén constituidas por las empresas “mejor preparadas” a escala mundial.

2.4 ROLES DE LAS DVE

Para llegar a entender de forma correcta el proceso de creación y gestión de servicios dentro de una DVE, es preciso conocer los diferentes roles involucrados en su existencia. Pero antes, y dado que la formación y existencia de las VE dinámicas se ha enmarcado dentro del contexto de los VBE (lo que supone que los miembros de las diferentes VE dinámicas son, a su vez, miembros del entorno de gestación), también es necesario saber quiénes son los principales actores y roles que aparecen en estos entornos.

Los integrantes esenciales de un VBE son las empresas, es decir, entidades de negocio que suministran productos o servicios al mercado y que se involucran en una VE dinámica en busca de beneficio. Pero un entorno de gestación de DVE puede incluir, además, otros tipos de organizaciones tales como institutos de investigación, consultorías, organizaciones de soporte gubernamental, ministerios, etc.

También, y como ya se ha mencionado, pueden incluir instituciones de soporte que proporcionan servicios especializados relacionados con las competencias de los miembros del entorno de gestación, tales como proveedores de servicios legales y contractuales, cámaras de comercio, entidades de asesoramiento, etc.

Dentro de un entorno de gestación de las VE dinámicas se pueden identificar varios roles, de entre los que destacan claramente dos (Afsarmanesh y Camarinha-Matos, 2005):

- *Miembro del VBE.* Éste es el rol básico de aquellas organizaciones que se registran en el VBE y están listas para participar en sus actividades.
- *Administrador del VBE.* Este rol lo desempeña la organización responsable del funcionamiento y evolución del entorno de gestación. Se encarga de promover la cooperación entre los miembros y de corregir las carencias de competencias buscando y reclutando/invitando a nuevas organizaciones. También gestiona los procesos generales del VBE (como, por ejemplo, la asignación y reasignación de los derechos a los actores involucrados en el entorno de gestación en base a sus responsabilidades), resuelve los conflictos, prepara el *pool* de activos del VBE, elabora las políticas de actuación comunes, etc.

Aunque los roles anteriores son los principales no son los únicos. En ocasiones, puede ser interesante definir otras figuras como la de consejero, suministrador de servicios, suministrador de ontologías,... , y por último, aunque no menos importante, el rol de invitado del VBE desempeñado por una organización no perteneciente al VBE que pueda estar interesada en: encontrar información general del VBE, llegar a formar parte del VBE, contactar con el VBE en lo referente a una oportunidad de negocio, etc. Sin embargo, y para los propósitos de esta tesis, es suficiente con la utilización de las dos figuras iniciales.

De la unión temporal de varios miembros del VBE surgirán las VE dinámicas que intenten explotar las oportunidades de negocio que puedan surgir. Por tanto, los miembros del VBE serán, a su vez, potenciales miembros de las DVE, y podrán adquirir nuevas figuras:

- Dada la *naturaleza autónoma* de las organizaciones que componen el VBE, en diferentes momentos (o incluso simultáneamente) la misma entidad puede asumir diferentes papeles dentro de la DVE (por ejemplo, actuar como miembro de la DVE, coordinador, etc.), y cada uno de ellos le confiere diferentes derechos/responsabilidades dentro del entorno de gestación.
- Debido a la *naturaleza dinámica* de las DVE, los roles de las organizaciones miembro no son estáticos, por lo que con cada nueva figura adoptada estas entidades ven modificadas tanto sus responsabilidades, como sus derechos y permisos. Por ello, los miembros de las DVE precisan de herramientas de asistencia que les permitan realizar sus tareas de manera eficiente¹⁹.

Por tanto, y considerando los derechos y responsabilidades que deben atribuirse a cada rol de un actor en el VBE, es necesario que los miembros del entorno de gestación informen al administrador del VBE sobre cada nuevo rol que planeen asumir (comenzando por convertirse en una organización miembro del VBE) para solicitar los derechos y asumir las responsabilidades que le corresponden.

19 Supongamos, por ejemplo, que un miembro del VBE, actuando en el papel de *broker* acepta la responsabilidad de configurar y negociar una DVE. Para realizar esta misión necesita permisos de accesibilidad/visibilidad a la información correspondiente al rendimiento pasado de otras organizaciones y herramientas que le ayuden a buscar las mejores organizaciones de acuerdo con las habilidades requeridas.

Pero, ¿cuáles son los posibles roles que suelen aparecer durante la existencia de una VE dinámica?. En base a las definiciones, descripciones y los modelos de ciclo de vida presentados hasta el momento se llega a la conclusión de que en la creación y gestión de los servicios de una DVE pueden participar diferentes dominios administrativos (Katz y Dissel, 2001; Camarinha-Matos *et al.*, 2005; Ouzounis, 2001; Afsarmanesh, 2005; Petersen, 2003; etc.):

- *Socio potencial o candidato de la DVE*. Éste es el rol básico desempeñado por aquellos dominios que ofrecen un conjunto de procesos de negocio a la comunidad con el fin de lograr una cooperación efectiva. En nuestro caso, los socios potenciales proceden, principalmente, del entorno de gestación pero también pueden ser entidades externas (socios habituales de los miembros del VBE o incluso empresas procedentes del “universo abierto” de organizaciones). En cualquier caso, cuando un socio potencial es seleccionado después de un proceso de negociación se convierte en un socio o miembro de la VE dinámica.
- El rol de *socio o miembro de la DVE* lo ejercen las organizaciones que forman parte de la DVE. El socio es la entidad encargada de proporcionar al resto de dominios involucrados los procesos acordados. Obviamente, y por definición, es la entidad que mejor sabe desarrollar la función para la cual se ha incorporado, la que mejor satisface los requisitos y exigencias del proceso a realizar.
- *Cliente*. Es el beneficiario del servicio de la VE dinámica. Puede ser un *cliente interno* (un componente del VBE que requiera un determinado producto o servicio) o *externo*. El cliente puede comenzar un servicio, suspenderlo, reasumirlo o terminarlo.
- El *representante* es la entidad que representa a la VE dinámica en el mundo exterior. Es el domino al que accede el cliente para solicitar ciertos servicios o productos y a quien exigirá responsabilidades en el caso de que no se cumplan los requisitos acordados.
- *Iniciador, broker de la oportunidad* o simplemente *broker*. Esta función la realiza un actor del VBE que busca, identifica y evalúa nuevas oportunidades de colaboración mediante la comercialización de las competencias y activos del VBE, y la negociación con los clientes potenciales. Cuando la DVE se configura para prestar un servicio a un cliente, el iniciador es el representante de dicha DVE, pero si se forma para aprovechar una oportunidad de negocio el iniciador puede ser cualquier miembro (también existe la posibilidad de que esta función la realice una entidad no perteneciente al VBE).

- *Planificador o integrador del negocio.* Esta función la realiza un agente del VBE que, frente a una nueva oportunidad de colaboración (diseñada por un *broker*), identifica las competencias y las capacidades necesarias, selecciona un conjunto de socios apropiado, y estructura la nueva VE dinámica. En muchos casos, este rol y el de *broker* los ejerce el mismo actor.
- *Coordinador de la DVE.* Es el encargado de coordinar la DVE durante su ciclo vital: ejecuta y dirige los procesos de una manera transparente para el usuario empleando las capacidades y los activos del entorno de gestación. Si es el cliente el que recurre a la empresa el representante también suele ejercer de coordinador.

La tabla 2.2 relaciona las etapas del ciclo de vida de la DVE con los principales actores involucrados en su existencia.

	<i>Etapas</i>	<i>Principales roles involucrados</i>
<i>Creación</i>	<i>Identificación de la CO</i>	El <i>broker</i> /representante y el cliente.
	<i>Una primera planificación de la DVE</i>	El <i>broker</i> y el planificador.
	<i>Búsqueda y selección de los socios</i>	El planificador, los miembros del entorno de gestación y, en algunos casos, el <i>broker</i> .
	<i>Negociación</i>	El planificador de la DVE y los miembros del VBE.
	<i>Planificación detallada</i>	El planificador y los miembros.
	<i>Contratación</i>	El planificador y el coordinador de la DVE y los miembros del entorno de gestación.
	<i>Lanzamiento</i>	El coordinador y el administrador de la DVE y los miembros del entorno de gestación.
	<i>Operación</i>	Los socios de la DVE y su coordinador.
	<i>Disolución</i> ²⁰	Los socios de la DVE, el coordinador y el cliente.
	<i>Evolución</i>	Los mismos que en la fase de creación.

Tabla 2.2. Principales actores involucrados en las diferentes fases del ciclo de vida de la DVE.

20 Se observa que el cliente sólo aparece cuando solicita una solución y cuando ésta le es suministrada, lo que es muestra de la transparencia del modelo.

En un contexto como el actual el atributo clave que debe definir la cooperación entre estas entidades es la agilidad. Por ello, lo deseable sería tener la posibilidad de encontrar lo más rápidamente posible aquel conjunto de socios que mejor se adaptes a una oportunidad de negocio concreta y configurarlos, también con celeridad, constituyendo una red colaborativa capaz de explotar la oportunidad ágilmente.

Sin embargo, encontrar los socios adecuados y establecer las condiciones necesarias para iniciar el proceso de colaboración puede llegar a consumir mucho tiempo y recursos, convirtiéndose, de este modo, en una tarea inhibitoria de la deseada agilidad. La identificación y selección de socios en la VE dinámica se convierte, de este modo, en un tema tremendamente complejo y abierto, además, a múltiples soluciones.

En un intento por reducir esta complejidad, se ha establecido un marco conceptual común que sirva de punto de partida en el análisis del problema asociado a la identificación y selección de socios que se afrontará en breve. Ambos elementos, el marco conceptual y el análisis exhaustivo del problema de selección, constituirán el punto de partida en el desarrollo de nuestro modelo.

Un elemento clave de este marco conceptual son los entornos de gestación de VE, ya que a lo largo de la tesis hemos considerado que el proceso de creación de las DVE se da dentro del contexto de estos entornos. Los VBE son, sencillamente, asociaciones colaborativas a largo plazo compuestas por organizaciones preparadas para cooperar y, de esta forma, responder rápida y ágilmente a las oportunidades de colaboración que puedan ir surgiendo.

Dentro de los entornos de gestación nacen, viven y mueren las VE dinámicas. Por este motivo, hemos iniciado el presente capítulo describiendo cada una de las etapas del ciclo vital de las DVE, prestando una especial atención a aquellas fases más interesantes desde el punto de vista de la identificación y la selección de socios, es decir, las etapas de creación y evolución.

Sin embargo, la definición de este marco conceptual de partida no estaría completa sin una descripción de cada uno de los roles que pueden aparecer durante el ciclo de vida de la VE dinámica. Este es el objetivo de la tercera y última parte del capítulo, en la que se han analizado cada uno de los actores involucrados en la existencia de las VE dinámicas incluyendo, como no podía ser de otro modo, los más representativos de los entornos de gestación.

Llegados a este punto, ya estamos en condiciones de centrarnos en el objetivo fundamental de la presente tesis, que no es otro que la identificación y posterior selección de los socios más adecuados con los que configurar una empresa virtual dinámica hecha a medida de una necesidad de mercado u oportunidad de negocio.

CAPÍTULO

3

LA SELECCIÓN DE SOCIOS EN LA LITERATURA DE INVESTIGACIÓN

3.1 INTRODUCCIÓN

En las empresas tradicionales el proceso de selección de socios suele ser bastante tedioso y pesado. De hecho, cualquier entidad que desee encontrar nuevos “compañeros de viaje” debe, en primer lugar, hallar un conjunto de empresas candidatas. Para ello, suele recurrirse bien a las relaciones de negocio o bien a recomendaciones de terceros. En cualquier caso, a estas empresas candidatas se les solicita información básica (por medio de teléfono, fax, correo, e-mail, etc.) que servirá de filtro para realizar una primera selección.

El siguiente paso consiste en enviar a cada una de las empresas seleccionadas un grupo de ejecutivos y/o técnicos para que realicen una investigación *in situ* con la que recabar información más precisa sobre las mismas. Con esta información más detallada, y con las impresiones directas obtenidas durante

el periodo de recopilación, se inicia un proceso de negociación con aquellas entidades más prometedoras que dará como resultado la selección de la empresa más satisfactoria. El proceso de selección finaliza, obviamente, con la firma de un contrato.

Además de consumir mucho tiempo, recursos y esfuerzos, este proceso de selección presenta una serie de inconvenientes significativos. En primer lugar, a través de los medios y herramientas de comunicación tradicionales la empresa solamente puede conocer y acceder a una pequeña fracción de los potenciales socios disponibles, lo que no conduce a una solución óptima desde el punto de vista global. Además, el proceso de selección de socios suele separarse del proceso de diseño del producto lo que suele conducir, de manera habitual, a multiplicidad de diseños (producto/proceso).

También debemos tener en cuenta que debido a la ausencia de un enfoque de selección sistemático y a la falta de habilidad para cuantificar factores intangibles y subjetivos, las empresas suelen recurrir al coste como el elemento más seguro e incontrovertible para tomar sus decisiones sobre la selección de socios (Chu *et al.*, 2000). Con este comportamiento se ignoran una serie de factores que pueden llegar a ser muy importantes, e incluso decisivos, a la hora de determinar la idoneidad de los socios potenciales.

Sin embargo, al abordar toda esta problemática desde la perspectiva de la VE dinámica, se observa que las dos primeras dificultades dejan de ser tales dada la naturaleza de este nuevo modelo organizativo. En concreto, al considerar la selección de socios dentro del contexto de los entornos de gestación, que como se recordará son asociaciones de borde controlado pero no cerrado, permitimos soluciones óptimas desde un punto de vista global. Además, y como se ha mencionado, los procesos de selección de socios y diseño del producto están estrechamente vinculados tanto en la fases de creación como en la de reconfiguración.

Por tanto, el principal inconveniente reside en la falta de herramientas y de un proceso de selección sistemático que ayuden en la selección de los socios adecuados para formar y/o reconfigurar la VE dinámica, en base tanto a factores tangibles (objetivos, técnicos) como intangibles (subjetivos). Esta cuestión es, sin duda, uno de los asuntos clave para asegurar el éxito de este nuevo modelo organizativo, lo que explica la aparición de numerosos trabajos y proyectos de investigación en los que se aborda esta problemática desde perspectivas bien diversas.

El presente capítulo nace con el propósito de conocer parte de esta investigación y descubrir los modelos y herramientas que se están empleando para afrontar este reto. A partir de la literatura consultada se observa que para el objetivo que se persigue los sistemas multiagente son los que presentan un mayor potencial debido, sobre todo, a su capacidad para afrontar los problemas de naturaleza distribuida. Es por ello, y por la posibilidad de comprender mejor algunas de las decisiones y consideraciones que se realizarán en los próximos capítulos, que los proyectos y trabajos de investigación que giran en torno a esta tecnología tienen una especial importancia en el presente capítulo.

Pero antes de proceder con el análisis del estado del arte sobre la investigación vinculada a la “identificación y selección de socios en la empresa virtual dinámica” es preciso realizar algunas reflexiones iniciales que den a conocer la complejidad del problema al que nos enfrentamos y que, al mismo tiempo, nos permitan definir la estructura que regirá el resto del capítulo.

3.2 CONSIDERACIONES INICIALES

La agilidad es, en un entorno tan dinámico como el actual, una de las prioridades competitivas más importantes desde el punto de vista empresarial. La VE dinámica, caracterizada por el hecho de que entidades legalmente independientes sean capaces de unirse y configurarse ellas mismas formando una organización colaborativa orientada a la consecución de una meta u objetivo, constituye para la mayoría de las empresas el único medio de alcanzarla. Sin embargo, encontrar los socios adecuados y establecer las condiciones necesarias para iniciar el proceso de colaboración suele suponer, además de tiempo, una pérdida de agilidad que debemos intentar subsanar.

Entre los principales obstáculos que aparecen en este proceso se encuentran (Camarinha-Matos *et al.*, 2005) la falta de información (sobre los posibles socios, sus procesos, el desempeño de su actividad,...), la falta de preparación de las organizaciones para unirse y trabajar formando redes cooperativas, la heterogeneidad de los socios potenciales (diferentes infraestructuras, cultura corporativa, métodos de trabajo, prácticas de negocio,...), etc., aunque, sin duda alguna, el obstáculo por excelencia es la falta de confianza (Davidrajuh y Deng, 2000).

Por tanto, la selección de socios no puede limitarse a un simple problema de “optimización”, tal y como ocurre en el enfoque tradicional, donde la asignación de procesos es función del coste, el potencial y las habilidades (competencias nucleares) de los posibles candidatos, es decir, en función única y exclusivamente de factores técnicos y objetivos. Se han de tener presentes otros muchos factores entre los que se incluyen algunos de naturaleza subjetiva, como son las preferencias personales o la confianza resultante del conjunto de experiencias previas.

El enfoque de selección tradicional, es decir, aquel basado en el simple emparejamiento de competencias (*competency matching*), tiene su origen en las prácticas empleadas en la gestión tradicional de los recursos humanos (HR, *Human Resources*). Atendiendo a estas prácticas, primero se describe con precisión el conjunto de tareas, actividades y responsabilidades que caracterizan un determinado puesto de trabajo y, después, se encuentra a quien mejor se ajuste a dicha descripción.

En dicha gestión un elemento clave es el perfil, es decir, el conjunto de habilidades que se precisan de un posible candidato al puesto. En este contexto, una habilidad es lo que podemos hacer, la capacidad de realizar una tarea, ya sea ésta mental o física. Sin embargo, una habilidad no nos dice nada sobre cómo se aplica el conocimiento y las capacidades en el desempeño de una tarea, idea que sí está implícita en el término competencia. Por tanto, una competencia supone, además de la posesión de un conocimiento o habilidad, la capacidad de aplicarla y/o hacer que otros la apliquen.

Las competencias están estrechamente relacionadas con las situaciones de trabajo y, por tanto, representan el conjunto de factores “técnicos” y “objetivos” que pueden influir en la selección de los candidatos. Sin embargo, resulta evidente que existen otros factores, no directamente relacionados con las habilidades de los candidatos, que también deben influir en dicha selección. Esta idea es la que subyace en los enfoques de selección más modernos.

Entre los diferentes factores “no técnicos” destacan los relacionados con el comportamiento de los individuos: experiencia previa (cómo desempeñó el candidato su trabajo en el pasado, colaboró con el resto de sus compañeros, fue activo, cuál fue su contribución en proyectos anteriores,...), las competencias humanas (es constante en el trabajo, cómo influye en aquellos

que le rodean, cuál es su capacidad de liderazgo, presenta disposición a trabajar en equipo, ...), etcétera.

En definitiva, en la selección de los diferentes candidatos ya no basta con utilizar algoritmos de emparejamiento que solamente tengan en cuenta la adecuación de sus habilidades técnicas a las características del puesto de trabajo, sino que también es necesario filtrarles en base a otras competencias o factores, como son las relacionadas con su comportamiento, su relación con la dirección y los demás compañeros, etc.

Todas estas consideraciones que se hacen en el contexto de la gestión de los recursos humanos también pueden aplicarse en la selección de las entidades que entrarán a formar parte de una empresa virtual dinámica. Naturalmente, aquí debemos incluir aspectos como son la buena voluntad hacia la cooperación interorganizacional, el desarrollo de las relaciones pasadas, localización geográfica, costes, robustez de las competencias, etc.

En este punto, es necesario disponer tanto de un perfil de confianza como de información adecuada sobre el funcionamiento de aquellas entidades que desean formar parte de una DVE. También se precisa llevar a cabo un proceso de filtrado de los posibles candidatos, lo que supone una pérdida considerable de tiempo. Obviamente, si deseamos mantener los altos niveles de agilidad asociados a este paradigma organizativo necesitamos reducir, en la medida de lo posible, este tiempo.

Para lograr este objetivo, en los últimos años han surgido diferentes enfoques que van desde la simple automatización computacional del proceso hasta los recientes servicios web, pasando por los sistemas multiagente.

3.2.1 El enfoque computacional o asistido

El nacimiento del paradigma de empresa virtual vino acompañado de la utilización de un enfoque de búsqueda y selección de socios manual, caracterizado por una alta involucración de las personas, ausencia de automatización de los procesos, falta de uniformidad en la información y por el uso de grandes cantidades de papel (Afsarmanesh, 2006b). Sin embargo, la aparición y desarrollo de las ICT supuso la introducción de nuevas funcionalidades que favorecieron la automatización computacional, dando lugar al enfoque asistido.

Esta nueva orientación permite una reducción significativa de la participación humana, un alto nivel de automatización y una menor utilización de papel. Sin embargo, su correcta implantación exige la definición de procedimientos normalizados para anunciar públicamente las ofertas de negocio, así como para recibir y gestionar las posibles respuestas (Camarinha-Matos *et al.*, 2005). La estandarización se convierte, por tanto, en el principal obstáculo de la “automatización” electrónica.

Ambos enfoques, manual y asistido, se conocen con el nombre de “enfoques tradicionales”, ya que se derivan de los métodos que tradicionalmente se han estado empleando en la formación de los grupos de trabajo en las grandes empresas o empresas extendidas, basados principalmente en el emparejamiento de competencias (Afsarmanesh, 2006b). En consecuencia, no representan una solución acertada del problema de selección de socios al no constituir una visión plenamente operativa ni realista (al no considerar de forma correcta todos aquellos factores de naturaleza subjetiva) del mismo .

A pesar de ello, en la literatura existen bastantes proyectos I+D+i que han seguido este enfoque tradicional, tal y como se podrá apreciar en el apartado 3.3. Obviamente, la mayoría de estos proyectos se corresponden con las etapas iniciales en el desarrollo del paradigma. Por otro lado, también son numerosos los trabajos en los que se discuten diferentes algoritmos de optimización más o menos acertados (véase apartado 3.4).

3.2.2 El enfoque basado en los sistemas multiagente

En los últimos años se han publicado un gran número de trabajos en los que se utilizan los sistemas multiagente y los mecanismos de negociación orientados al mercado para la formación de empresas virtuales. Como veremos más adelante, en la mayoría de ellos (Li *et al.*, 2000; Rocha y Oliveira, 1999; Ouzounis, 2001;...) se asume la existencia de un mercado virtual donde las empresas, representadas por agentes geográficamente distribuidos, pueden unirse y cooperar para lograr una meta de negocio común.

Para modelar este mercado electrónico que soporta la formación de la DVE en estos trabajos suele proponerse una arquitectura MAS en la que, además de los agentes que representan a las empresas, existe un agente de mercado o *broker* encargado de coordinar el proceso de búsqueda, identificación,

negociación, etc. Estos sistemas, sin embargo, presentan algunos puntos débiles: no suelen considerar suficientemente aspectos de selección subjetivos como la confianza y el compromiso, muestran carencias en seguridad, no existen estándares y ontologías comunes, etc.

En el apartado 3.5 de este capítulo se discute la idoneidad de la tecnología multiagente para afrontar los retos y desafíos que plantean las VE dinámicas y se presentan algunos de los trabajos más relevantes en los que se emplea para identificar y seleccionar los socios de la DVE. Este análisis permitirá establecer algunos de los problemas habituales que presentan este tipo de soluciones.

3.2.3 El enfoque basado en los servicios web

Recientemente también han empezado a aparecer trabajos en los que se emplea el paradigma de los servicios web (*Web Services*). El apartado 3.6 del presente capítulo constituye una primera aproximación a este nuevo modelo. En él se describe brevemente qué son los servicios web, se discute su aplicación a la selección de socios en la VE y se presentan algunos de los trabajos en los que se hace un uso combinado de esta tecnología con la multiagente.

En particular, este nuevo planteamiento, denominado “enfoque basado en la federación de servicios” (Malucelli *et al.*, 2005), considera a los potenciales miembros como “proveedores de servicios” que publican sus ofertas en un catálogo conocido como “portal de servicios”. Por otro lado, los miembros de la DVE puede “consumir” los servicios de los otros dentro de un mercado electrónico (Petersen, 2003). Este enfoque presenta, por tanto, una selección de socios indirecta ya que se selecciona el servicio, no su suministrador: los socios son implícitamente escogidos a través de los servicios que han sido seleccionados.

Sin embargo, y a pesar de su reciente popularidad, existen multitud de limitaciones para aplicar eficientemente este paradigma a la selección de socios en las VE dinámicas (Camarinha-Matos *et al.*, 2005): ¿están los servicios siempre disponibles?, ¿cuál es nivel de conocimiento del proveedor del servicio?, ¿pueden las tareas ser representadas como servicios?, ¿tiene sentido aplicar servicios específicos para la selección de socios?, ¿y para el proceso de negociación?, etc. Además, la mayoría de los desarrollos hasta la

fecha se han realizado para el comercio electrónico y no para actividades colaborativas, existe una pobre integración de los mecanismos de seguridad y privacidad, y los mecanismos de búsqueda son todavía muy pobres (Camarinha-Matos, 2003).

De los tres enfoques considerados, creemos que la utilización de los MAS constituyen una forma acertada de abordar el problema de la selección de socios en entornos dinámicos. Con los MAS, no sólo es posible conseguir lo mismo que se logra con el enfoque federación sino que además, y gracias a los avances recientes, podemos integrar fácilmente ambas tecnologías, potenciando sus ventajas y atenuando sus carencias. Los trabajos descritos en (Hao *et al.*, 2005; Petrie y Bussler, 2003; Malucelli *et al.*, 2005; Matskin *et al.*, 2005; etc.) representan los primeros pasos hacia dicha posibilidad.

En consecuencia, y a la espera de que el enfoque “federación de servicios” muestre su verdadero potencial, estimamos que la orientación multiagente es la más adecuada para abordar la problemática de la selección de socios. Pero para lograr con éxito nuestro objetivo, primero debemos librarnos de algunos obstáculos que han venido lastrando la investigación asociada, entre los que se encuentran la falta de ontologías, la desconsideración de aspectos sociológicos, las carencias en seguridad, etc.

3.3 REVISIÓN DE PROYECTOS I+D+I RELACIONADOS

La noción de AVE ha despertado grandes expectativas en diferentes dominios de aplicación. Ya conocemos muchos de los beneficios potenciales de este tipo de redes colaborativas: acceso a nuevas oportunidades de negocio, se alcanzan metas de negocio que para una única organización serían difíciles (o imposibles) de afrontar, se comporten riesgos, se reducen los costes, etc. Por otro lado, el hecho de que ante la aparición de una oportunidad de negocio se pueda configurar rápidamente una DVE *ad hoc*, se reconoce como una expresión de agilidad, cualidad que garantiza la supervivencia en aquellos escenarios que, como los actuales, están caracterizados por una enorme turbulencia.

Precisamente, son estas expectativas las responsables de que durante las dos últimas décadas hayan surgido un gran número de proyectos de I+D+i en los que se intenta establecer tanto las bases tecnológicas como las prácticas operativas necesarias para soportar las DVE. Este esfuerzo es particularmente

visible en Europa a través de los múltiples programas financiados por la Unión Europea, pero también en EEUU, Brasil, Japón, China, o en otras regiones geográficas. Incluso algunos programas, como es el caso de IMS (*Intelligent Manufacturing Systems*) (IMS web, 2006), soportan proyectos en este área que involucran organizaciones de varios continentes.

Los primeros pasos significativos se dieron en Estados Unidos a principios de los 90. En concreto, a finales de 1993 comenzaba su andadura el proyecto NIIP (*Nacional Industrial Information Infrastructure Protocols*), posiblemente el primero de los más grandes y significativos en el área de la VE. Desde entonces, y hasta nuestros días, el número de proyectos y actividades de I+D+i mundiales relacionadas con los diferentes aspectos de las VE, y en particular con su ciclo de vida, ha crecido de un modo muy significativo. Sin embargo, en muchos de esos proyectos el grado de automatización del ciclo de vida de la VE es limitado y, por lo tanto, inadecuado para satisfacer las necesidades de las DVE (Ouzounis, 2001).

El proyecto NIIP fue liderado por el consorcio homónimo (NIIP web, 2003), una iniciativa formada por el gobierno y la industria de los EEUU, con el objeto de desarrollar tecnologías software que hiciesen viables las VE (Dang, 2004) a lo largo de todo su ciclo de vida. La meta no era otra que la de proporcionar los procesos y los protocolos que permitieran a las organizaciones colaborar entre sí, operando como empresas virtuales, sin importar las estructuras de datos, los procesos o los entornos de computación. En resumen, pretendía soportar la formación de VE industriales y proporcionar tecnologías que permitiesen a sus participantes colaborar dentro de un entorno de computación heterogéneo.

A pesar de su gran importancia, el proyecto NIIP planteó un enfoque muy genérico y en discordancia con la realidad de muchos de los sectores de negocio actuales (Ouzounis, 2001). En concreto, desarrolló conceptos y prototipos para las empresas virtuales estáticas y no logró un nivel de automatización adecuado en las diferentes fases del ciclo de vida. Por ejemplo, la selección de los socios se realizaba manualmente, sin ningún tipo de mecanismo de emparejamiento, lo que condicionaba significativamente tanto la creación como la evolución de la empresa virtual.

No obstante, y a pesar de no suponer un enfoque consistente para la creación, gestión y ejecución de procesos de negocio inter-organizacionales, sí se convirtió en un punto de partida para la aparición de un gran número de

proyectos con la VE como trasfondo. Uno de esos proyectos fue PRODNET II (PRODNET II web, 2003), *Production Planning and Management in Virtual Enterprises*, cuya meta principal consistió en el desarrollo de infraestructuras que soportasen las VE industriales a lo largo de su vida, centrándose, principalmente, en las fases de creación y operación (Camarinha-Matos y Afsarmanesh, 1999a). En particular, perseguía diseñar y desarrollar una infraestructura abierta y flexible que soportase el ciclo de vida de las VE de fabricación industrial, prestando una atención especial a las necesidades de las pequeñas y medianas empresas (SME, *Small and Medium Enterprises*).

Para la primera de estas fases, que es lo que nos interesa, el proyecto incluía un mecanismo de búsqueda y selección de socios dirigido por el usuario, sin una herramienta de negociación y basada en mercados virtuales públicos. La búsqueda de los socios potenciales se realiza a partir de algunos directorios públicos y una lista privada de suministradores donde se almacenaban perfiles de compañías relacionadas con ciertos productos o procesos. Además, la selección de los socios se realizaba *ad hoc*, sin ningún mecanismo de negociación.

Tanto el proyecto NIIP como PRODNET II representan dos de los grandes esfuerzos iniciales en afrontar la problemática asociada a la empresa virtual. Sin embargo, dichos esfuerzos se dirigían hacia las relaciones de negocio estáticas y altamente acopladas y descuidaban profundamente la selección dinámica de los socios. Estos problemas son comunes a muchos otros proyectos I+D+i orientados hacia topologías tipo cadena de suministro como e-MMEDIATE (e-MMEDIATE web, 2006), PRODCHAIN (PRODCHAIN web, 2006) y el propio PRODNET II, u orientados a la topología en estrella de las CSVE como e-COGNOS (e-COGNOS web, 2006), GLOBMEN (GLOBMEN web, 2006), ICCI (ICCI web, 2006), ISTforCE (ISTforCE web, 2006), etc.

Sin embargo, también podemos encontrar en la literatura múltiples ejemplos de proyectos enfocados a las relaciones dinámicas donde la topología es de tipo *peer-to-peer* como BIDSAVER (BIDSAVER web, 2006), EXTERNAL (EXTERNAL web, 2006), FETISH-ETF (FETISH-ETF web, 2006), GNOSIS (GNOSIS web, 2006), SYMPHONY (SYMPHONY web, 2006), MASSYVE (MASSYVE web, 2006), UEML (UEMLweb, 2006), etc., o proyectos de topología mixta como ALIVE (ALIVE web, 2006), THINKCreative (THINKCreative web, 2006), KM Forum (KM Forum web, 2006), CE-NET II (CE-NET II web, 2006), etc.

Por ejemplo, el proyecto BIDSAVER (*Business Integrator Dynamic Support Agents for Virtual Enterprise*) tiene por objeto la evaluación y desarrollo de un marco para la constitución y operación de empresas virtuales, en las que participan SME que se buscan en la web por medio de agentes de información. Además de definir una metodología, procesos de referencia y modelos de datos, este proyecto construye herramientas de infraestructura ICT para permitir la conducción del proyecto cooperativo y para la búsqueda dinámica de socios a través de la web, adecuada tanto para su rápida constitución como para una actualización dinámica y para las operaciones de la VE. También establece un marco legal para la gestión de las relaciones entre los socios de la VE y entre el mercado y la VE.

El proyecto MASSYVE (*Multiagent Manufacturing Agile Scheduling Systems for Virtual Enterprises*) (Rabelo *et al.*, 1998; Rabelo *et al.*, 2000) se centra en la secuenciación (*squeduling*) de las tareas que se han de ejecutar durante la fase de operación. En concreto, dirige la distribución de tareas usando agentes software que utilizan el protocolo Contract Net para asignar las tareas entre agentes. El procedimiento consiste en anunciar una tarea a través de la red de MAS y a continuación hacer que los agentes intercambien información entre sí hasta que uno de ellos sea seleccionado para realizar la tarea. Sin embargo, este enfoque sólo distribuye una tarea a la vez, por lo que es bastante lento si se necesitan asignar múltiples tareas (Dang, 2004). Además, esta asignación secuencial puede conducir a resultados no óptimos al ignorar las interdependencias que pueden darse entre las tareas.

Y ya que hemos empezado a hablar de la tecnología de agentes, es necesario destacar el proyecto AVE (*Agents in Virtual Enterprises*) (Fisher *et al.*, 1996), el primer proyecto en el que hace una utilización expresa del paradigma de agentes. AVE se centra en la utilización de agentes en la formación de la VE, aunque también aborda algunos aspectos de la operación, empleando mecanismos de subasta para formar la VE. Sin embargo, estos mecanismos son muy simples (el inglés, el alemán, puja sellada de primer precio y puja sellada de segundo precio), por lo que no permite a los socios potenciales variar su implicación potencial en la VE y, por tanto, no asegura la capacidad de seleccionar el conjunto de socios más adecuado en cada momento.

El actual proyecto *Electronic Institution including Electronic Contracting for Virtual Organizations* (Oliveira *et al.*, 2006) trabaja en el desarrollo de un marco software (una institución electrónica) donde los agentes, que

representan diferentes empresas, puedan interactuar de una manera regulada. El concepto de institución electrónica está relacionada con las instituciones del mundo real que definen las normas y reglas de la sociedad, regulando la actividad de sus individuos.

En este sentido, el proyecto intenta especificar e implantar un marco normativo que imponga tal gobierno a los agentes computacionales, incluyendo tanto las normas y reglas generales como aquellas directamente relacionadas con las actividades contractuales. Además de estas regulaciones existen una serie de servicios institucionales de vital importancia para la creación y operación de la empresa virtual: mediación en la negociación, plantillas de contratos, ontologías, validación de contratos, monitorización, ejecución, etc.

Entre los logros más destacados de este proyecto podemos destacar (Oliveira *et al.*, 2006): a) el desarrollo de un marco, basado en la plataforma JADE, para la creación de agentes de empresas (este marco incluye protocolos de negociación adaptativos para la interacción de los agentes), b) un primer prototipo de agente de servicios de ontologías, c) la definición de un formalismo de representación para las normas y los contratos, y d) una formalización completa del lenguaje de representación de contratos de la empresa virtual.

No podemos finalizar este apartado sin considerar brevemente los proyectos ECOLEAD, VOSTER y VE-Forum. El primero porque es donde se define con mayor precisión el concepto de entorno de gestación de VE que venimos empleando. El segundo porque se trata de un intento importante por capturar el estado del arte y los resultados en el área de las organizaciones virtuales; para ello recoge y sintetiza los resultados de numerosos proyectos de investigación europeos sobre los modelos de empresa virtuales. Finalmente, VE-Forum constituye un foro abierto para todos aquellos que están interesados en el campo de las redes y las organizaciones virtuales; supone un lugar de encuentro en el que intercambiar y desarrollar nuevas ideas relativas a este nuevo paradigma.

El proyecto integrado ECOLEAD (ECOLEAD web, 2006) es una iniciativa cuyo propósito es crear la base y los mecanismos necesarios para establecer una sociedad industrial avanzada basada en redes de cooperación (Camarinha-Matos *et al.*, 2005c). En la base de este proyecto se haya la idea de que una creación efectiva de DVE requiere la existencia de entornos de

gestación. Por ello, entre sus objetivos se marcan la determinación de los componentes y características clave de los VBE: su ciclo de vida y funcionalidades relacionadas, roles y derechos de los actores, ontologías, marcos de referencia, arquitecturas de sistema, etc.

Por su parte, VOSTER (VOSTER web, 2006) es un proyecto soportado por la Comisión Europea bajo el programa IST (*Information Society Technologies*) (IST web, 2006) cuyo propósito consiste en recoger, analizar y sintetizar los resultados de múltiples proyectos de investigación europeos líderes en el estudio de las organizaciones virtuales. La figura 3.1 muestra todos estos proyectos así como las relaciones existentes con otros proyectos dentro del ámbito europeo.

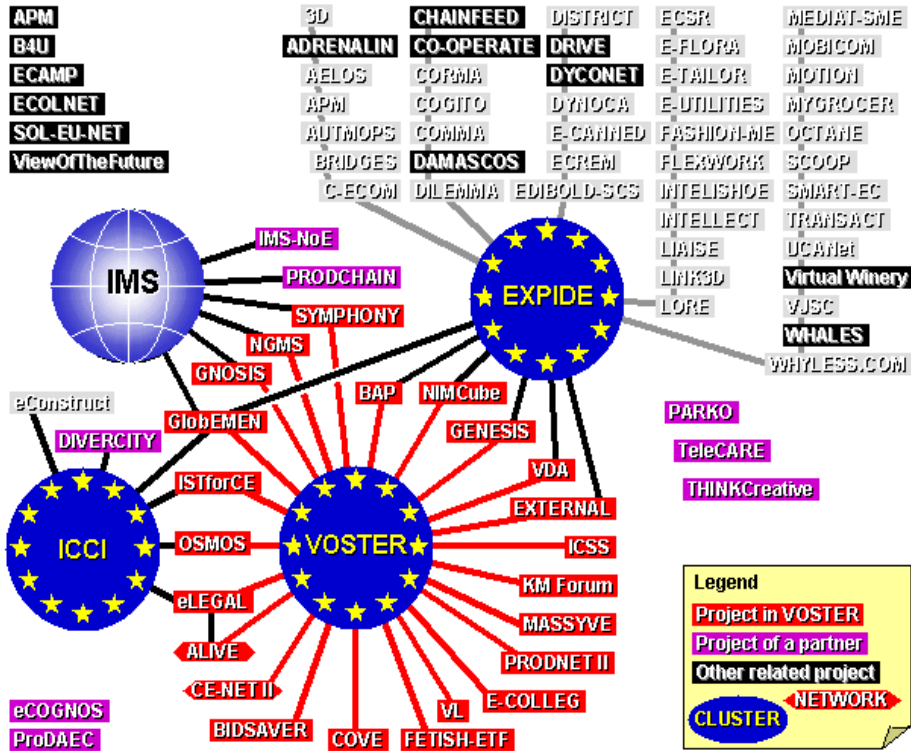


Figura 3.1. Proyectos enmarcados dentro de VOSTER y su relación con otros proyectos conducidos dentro del ámbito europeo (VOSTER web, 2006).

El equipo de VOSTER es un consorcio de 13 organizaciones europeas que representan 8 países diferentes. Todas ellas son participantes activas en la investigación relacionada con la VO, encontrándose entre los líderes de este campo de investigación²¹. Entre los objetivos científicos y tecnológicos que persigue el proyecto VOSTER encontramos los siguientes:

- Consolidar los conceptos relacionados con las VO y sus relaciones, tipos de VO, características e indicadores.
- Recomendar acercamientos útiles para el modelado de VO.
- Identificar tecnologías y estándares relevantes y determinar su potencial para su aplicación en las VO.
- Definir las funciones de las infraestructuras de VO y sugerir formas de implementarlas.
- Promover propuestas de VO en las industrias europeas.

Por lo que se refiere a la formación de las VE, el proyecto VOSTER se centra actualmente en la búsqueda y selección de socios basada en la negociación y en el papel de los mercados virtuales y los *brokers*. Además, se están dando los primeros pasos hacia el *e-contracting*.

Por último, encontramos el proyecto VE-Forum (VE-Forum web, 2006), el portal de la VE. Se trata del portal para todos aquellos investigadores y profesionales de la Unión Europea centrados en los desafíos y las oportunidades asociadas a las organizaciones conectadas en red y a las empresas virtuales. Además de proporcionar un fácil acceso a resultados conseguidos en el campo de las VE, VE-Forum soporta la comunicación y colaboración dentro de la comunidad.

Entre sus principales activos destacan (Hannus *et al.*, 2004): a) el acceso a proyectos, noticias, eventos y conferencias relacionados con los modelos de VE, b) un *pool* de conocimientos con los resultados de proyectos, una revista, acceso a los *abstracts* de las revistas más relevantes, *working papers*, y el directorio *who-is-who* donde se encuentran los datos más representativos de los miembros de la comunidad, y c) espacios de colaboración para foros de discusión, grupos de interés especial y cooperación entre proyectos²².

21 En (VOSTER web, 2006) se muestran todas estas organizaciones así como los proyectos asociados y sus direcciones web.

22 Entre los proyectos organizados por VE-Forum podemos encontrar Europe INNOVE, ECOLEAD, Equal-XLX, Satine Public Space, VOSTER, etc.

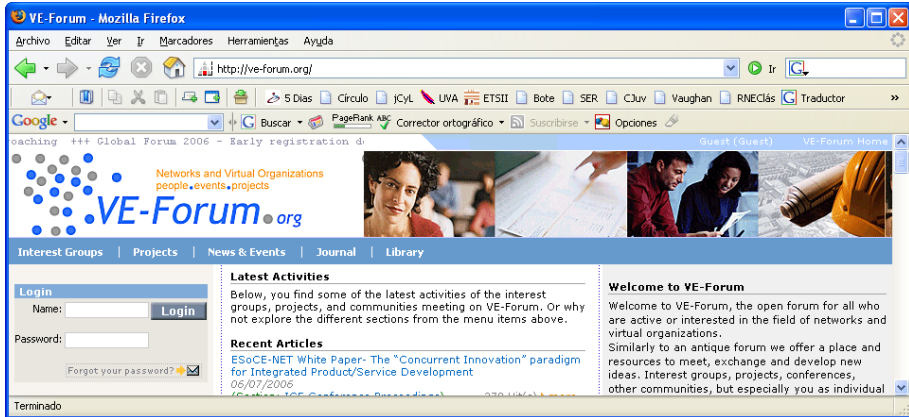


Figura 3.2. El portal VE-Forum.

3.4 LA SELECCIÓN DE SOCIOS DESDE UN ENFOQUE CUANTITATIVO

El proceso de selección de socios es una función importante para los sistemas de gestión de información de las empresas virtuales (Ip *et al.*, 2003) y ha atraído, como sabemos, la atención de una gran parte de la investigación reciente. Un alto porcentaje de esta investigación ha sido principalmente cualitativa y centrada en aspectos metodológicos (Meade *et al.*, 1997; Mikhailov, 2002), pero también se puede encontrar algunos trabajos basados en enfoques de toma de decisiones cuantitativos o matemáticos dentro del contexto de la empresa virtual.

Algunos de estos estudios han seguido el enfoque basado en la optimización del problema de selección de una manera similar a como se afronta en el contexto de la gestión de la cadena de suministro, donde la selección de los proveedores es una cuestión ampliamente tratada (Jarimo y Salo, 2006). A pesar de que esta selección de proveedores difiere de la selección de socios propia de la formación y reconfiguración de las empresas virtuales, algunos de los resultados obtenidos permiten una mejor comprensión del problema al que nos enfrentamos.

Así, por ejemplo, en 1997 Kasilingam y Lee (Kaslingam y Lee, 1997) presentan su trabajo, donde proponen un modelo de programación entera tanto para la selección de los socios como para la determinación de las

cantidades de las órdenes. Se trata de un modelo basado en los costes, en el que tanto los costes variables como los fijos forman parte de una función objetivo. Por tanto, el problema se reduce a una simple cuestión de optimización en el que se ha de minimizar una función de costes.

(Ko *et al.*, 2001), (Ip *et al.*, 2004) y (Wu y Su, 2005) son sólo algunos de los muchos trabajos en donde también se presentan modelos de programación entera donde el objetivo es minimizar los costes totales (incluidos los de producción, operación, transporte, etc.). Ko *et al.* (Ko *et al.*, 2001) resuelven el problema utilizando el método heurístico de la búsqueda tabú, mientras que Ip *et al.* (Ip *et al.*, 2004) emplean un algoritmo *branch and bound*. Por su parte, Wu y Su (Wu y Su, 2005) reformulan su modelo en términos de una representación gráfico-teórica y también emplean un algoritmo de aproximación.

Feng y Yamashiro (Feng y Yamashiro, 2003) minimizan una función de costes muy completa formada a través de la unión de los costes ocasionados por actividades de procesamiento, transporte, etc. En concreto, formulan un programa no lineal mixto que solucionan después de reducir el tamaño del modelo mediante una precualificación cuantitativa de los socios candidatos. Un modelo algo diferente es el presentado por Ip *et al.* (Ip *et al.*, 2003), en el que maximizan la probabilidad de éxito de la empresa virtual. Dado que su modelo no es lineal ni convexo, estos autores desarrollan un algoritmo genético para solucionarlo.

En todos estos modelos, así como en otros similares, sólo se tienen en cuenta criterios cuantitativos dado que los criterios cualitativos no pueden representarse fácilmente como funciones de costes. Sin embargo, la selección de socios requiere que en el proceso de toma de decisiones se consideren tanto criterios de evaluación cuantitativos, tales como la productividad anual y la estabilidad financiera (Lorange *et al.*, 1992), como índices cualitativos, tales como la reputación en el mercado o la sinceridad en la comunicación (Ip *et al.*, 2003). También es interesante incluir criterios subjetivos, tales como la coherencia cultural del negocio y los privilegios de los accionistas, e índices objetivos, tales como la imagen corporativa o el alcance geográfico (Mikhailov, 2002).

Tener que considerar todos estos atributos de decisión (cuantitativos y cualitativos, subjetivos y objetivos) hace que el proceso de evaluación se complique considerablemente (Chun-Wei y Hong-Yi, 2004) y que aumente

su dificultad. De la literatura consultada se puede concluir que el proceso de selección de socios debe ser considerado como un problema de *toma de decisiones multicriterio* más que un problema de modelado puramente matemático (Mikhailov, 2002).

El trabajo de Dean y Schniederjans (Dean y Schniederjans, 1993) introduce uno de los primeros modelos que tienen en cuenta esta consideración. Estos autores formulan el proceso de selección como un problema de toma de decisiones multicriterio en el que aplican un método de programación lineal por objetivos. Talluri *et al.* (Talluri *et al.*, 1999) también hacen uso de un modelo de programación por objetivos, pero tienen en cuenta algunas dependencias inter-organizacionales. Su modelo busca minimizar costes, distancias y el momento de inicio de las actividades, pero también maximizar la compatibilidad cultural. Para que su modelo sea computacionalmente tratable reducen el número de candidatos (y por tanto el tamaño del modelo), excluyendo a los más ineficientes.

Sin embargo, las ecuaciones matemáticas lineales empleadas en estos modelos no permiten representar adecuadamente el problema de selección. Dadas las características jerárquicas entre los atributos de evaluación, en la literatura se observa una amplia adopción de la técnica de análisis jerárquico AHP (*Analytical Hierarchy Process*). Entre las causas que explican esta elección se encuentra su capacidad inherente para manejar criterios tanto cuantitativos como cualitativos, que su procedimiento de decisión es simple y fácil de entender, y que el proceso de evaluación y selección es eficiente (Nydick y Hill, 1992; Partovi *et al.*, 1989).

3.4.1 AHP y su aplicación al problema de selección

El proceso de análisis jerárquico es una de las técnicas más usuales en el campo de los algoritmos multiatributo. Fue diseñado por Thomas L. Saaty a principios de la década de los 80 (Saaty, 1980) y resulta especialmente adecuado para decisiones complejas donde están involucradas comparaciones entre elementos con características difíciles de cuantificar.

El proceso de análisis jerárquico se basa en la descomposición, juicio comparativo y síntesis de las prioridades del problema de decisión (Gómez y Bosque, 2004). El problema de decisión ha de descomponerse en una jerarquía que capte los elementos esenciales del problema. El principio de

juicio comparativo se plasma mediante la comparación por pares de los elementos de cada nivel de la jerarquía. Finalmente, la síntesis de prioridades supone tomar cada una de las escalas de prioridades resultantes en cada nivel de jerarquía y construir un conjunto de prioridades global (compuesto) para el último nivel de la jerarquía, que será utilizado para evaluar las distintas alternativas. La aplicación de estos tres principios se lleva a cabo en varias fases (Saaty, 1995):

- El proceso comienza con la identificación de los criterios de decisión asociados al problema.
- Tras esta identificación, se estructuran los factores en diferentes niveles de una forma jerárquica, descendiendo desde los más generales a los más concretos, y de modo que los niveles adyacentes estén conectados entre sí.
- El tercer paso consiste en establecer la importancia relativa de los elementos de cada jerarquía a partir del método de comparación por pares.
- A continuación, se agregan los pesos de los niveles obtenidos en cada jerarquía, obteniendo así pesos compuestos o globales. Para ello, se multiplican los pesos relativos de la primera matriz (el primer nivel de jerarquía) por los pesos relativos de la segunda y así sucesivamente hasta llegar al último nivel de jerarquía. El vector de pesos compuestos final tendrá una dimensión de l por m , siendo m el número de alternativas de decisión en el último nivel de jerarquía. La puntuación total R_i para cada alternativa i se calcularía como sigue:

$$R_i = \sum_k w_k r_{ki} \quad (3.1)$$

donde w_k es el vector de prioridades (pesos) asociado a cada elemento k de la estructura jerárquica de criterios, la suma de w_k es igual a 1 y r_{ik} es el vector de prioridades obtenido al comparar las alternativas con cada criterio.

- Finalmente se ordenan las alternativas en función del valor R alcanzado, siendo la más adecuada aquella que obtiene el valor más alto.

Una de las primeras aplicaciones de esta técnica la encontramos en el trabajo presentado por Ossadnik (Ossadnik, 1996). Este autor propone el uso del método AHP para asignar las sinergias a los socios en una fusión, de acuerdo con el impacto de sus posibles rendimientos sobre el efecto sinérgico global.

Se proporcionan criterios tanto cuantitativos como cualitativos para medir este efecto sinérgico total, tales como el poder de innovación del departamento de I+D, las reservas de liquidez, el canal de distribución, el equipo de fabricación, la calidad del personal y de los gestores, etc. Se usa el método AHP estándar para deducir los pesos relativos de las sinergias de los componentes en relación a la cantidad total de efecto sinérgico.

En el trabajo presentado por Babic y Plazibat (Babic y Plazibat, 1998) se emplea el análisis multicriterio para establecer un *ranking* de las empresas candidatas de acuerdo con el nivel de eficiencia alcanzado en el desempeño de su actividad. Sin embargo, todos los criterios de eficiencia que definen son cuantitativos y equivalentes a los ya conocidos indicadores de eficiencia utilizados en el análisis financiero, tales como el margen de beneficio, el retorno de la inversión, el ratio de deuda, el nivel de *stock*, el ingreso por empleado, etc. En este estudio, se emplea el método AHP para determinar la importancia de los criterios con los que finalmente se determinará el *ranking*.

Meade *et al.* (Meade *et al.*, 1997) emplean el método ANP (*Analytic Networking Process*). ANP es una forma general de la teoría AHP que no requiere una estructura jerárquica ni relaciones unidireccionales entre los elementos de decisión. La importancia relativa del criterio y de las alternativas se determina por comparaciones análogas a las empleadas en AHP. Los autores identifican varias limitaciones a este enfoque, tales como que el proceso de adquisición de datos consume mucho tiempo, requiere datos precisos y exactos para las evaluaciones finales, etc., por lo que proponen algunas extensiones para hacer este enfoque más robusto.

Para superar la necesidad de datos precisos en los juicios de comparación, en (Mikhailov, 2002) se propone una extensión/generalización del método AHP. Mikhailov considera que un agente seleccionador normalmente no puede asignar valores precisos a los atributos de decisión, bien porque los criterios que se emplean son subjetivos y cualitativos, bien porque no se dispone de toda la información necesaria sobre los posibles socios y su desempeño (o ésta es incierta), etc. Por ello, propone la utilización de intervalos, en lugar de valores numéricos exactos, en los juicios de comparación²³.

23 Obviamente, la técnica AHP clásica es una particularización de esta extensión, ya que los números reales pueden ser representados como un intervalo en el que coinciden su límite inferior y superior.

A pesar de suponer un evidente mejora, el método propuesto por Mikhailov ignora los efectos propios de la interdependencia entre diferentes atributos. Para considerar dichos efectos podemos considerar trabajos como el de Efstathiou y Rajkovic (Efstathiou y Rajkovic, 1979) donde sugirieron el uso de reglas *fuzzy* para derivar la función de utilidad del decisor. Sin embargo, el enfoque de estos autores no puede ser aplicado a problemas complejos, tales como aquellos representados por una estructura jerárquica de los atributos de decisión (Chun-Wei y Hong-Yi, 2004).

En estos trabajos se aprecian dos de los requisitos básicos que se han de tener siempre presente en el problema de selección: por un lado, deben incluirse tanto atributos de decisión cualitativos como cuantitativos y, por otro, debe considerarse la interdependencia entre ellos. Sin embargo, en ninguno de estos trabajos se considera que los recursos disponibles para obtener la información sobre los atributos de evaluación (como pueden ser el tiempo y el dinero) son limitados. Algunos trabajos, como el presentado en (Chun-Wei y Hong-Yi, 2004) van en esta línea.

Adquirir toda la información sobre los atributos de evaluación es siempre complicado y costoso. Además, a medida que estos aumentan, su interdependencia también se ve incrementada (Chun-Wei y Hong-Yi, 2004). Por tanto, la cantidad de atributos y restricciones consideradas en el proceso de decisión afectarán a la calidad y eficiencia en la selección de los socios más adecuados. Por ello, estos autores presentan un marco de toma de decisiones *fuzzy* para seleccionar la alianza más favorable bajo recursos de evaluación limitados en el que consideran más de 180 criterios de evaluación.

El modelo presentado por Fischer *et al.* (Fischer *et al.*, 2004) tiene en cuenta las dependencias inter-organizacionales de un grupo de organizaciones. Para ello, introducen el concepto de células de competencia (CC, *Competence Cells*) o pequeñas unidades de funcionamiento que son seleccionadas en función de la oportunidad de negocio. Para encontrar la selección óptima de células se recurre a la técnica de optimización de la colonia de hormigas (*Ant-Colony Optimization*), mientras que para la computación de la función objetivo se utiliza el enfoque AHP.

Del análisis de estos modelos se observan dos aspectos relevantes. En primer lugar, son muy pocos los que tienen en cuenta consideraciones como el éxito en colaboraciones pasadas, homogeneidad cultural y de infraestructuras ICT, grado de confianza interorganizacional,... , que contribuyen al éxito esperado

de la futura colaboración (Camarinha-Matos y Afsarmanesh, 2003). Tales criterios deben ser considerados para un grupo de organizaciones y no para una sola, lo que conduce a modelos más complejos (Fischer *et al.*, 2004). En segundo lugar, pocos estudios tratan aspectos relacionados con la gestión del riesgo en la selección de socios, a pesar de que la minimización de los riesgos asociados, por ejemplo, a las fluctuaciones de capacidad y a los fallos de calidad son muy relevantes.

Por último, resulta significativo señalar que la teoría AHP ha recibido numerosas críticas procedentes, sobre todo, de su incompatibilidad con un gran número de principios ampliamente aceptados en la toma de decisiones racional (Dyer, 1990). Este hecho explica, en parte, la extensa proliferación de modificaciones, extensiones y generalizaciones que tratan de mejorar esta técnica.

3.5 LOS MAS Y LA SELECCIÓN DE SOCIOS EN LA DVE

Los agentes son entidades software autónomas diseñadas para operar en entornos distribuidos, cuyo comportamiento está orientado hacia la consecución de una serie de metas. Pueden llevar a cabo interacciones sofisticadas, gracias a sus habilidades sociales, y son capaces de responder oportunamente a los cambios en su entorno (Jennings *et al.*, 1996). Cuando varios agentes colaboran e interactúan entre sí para resolver un problema nos encontramos frente a un sistema multiagente. En un nivel abstracto, los MAS muestran muchas similitudes con las redes colaborativas (Afsarmanesh, 2006), lo que permite explicar, en parte, la fuerte tendencia a emplear estos sistemas como herramienta de modelado e implementación que soporte el proceso de creación de las VE dinámicas.

De hecho, a lo largo de estos últimos años se han publicado un gran número de trabajos en los que se aplican los sistemas multiagente y los mecanismos de negociación orientados al mercado en la formación de las empresas virtuales (Camarinha-Matos and Afsarmanesh, 2001). Estos trabajos siguen enfoques variados, lo que se traduce en arquitecturas e infraestructuras multiagente diferentes. Así, podemos encontrar ejemplos en los que la formación de la VE se considera dentro del contexto de un mercado electrónico, otros cuya solución gira en torno a la figura de un *broker*, también están los que consideran arquitecturas de agentes móviles, o incluso aquellos en los que aparece la noción de agentes de información, etc.

A lo largo de este apartado se van a mostrar, de forma resumida, algunos de los aspectos relacionados con los MAS que son potencialmente relevantes para la creación de DVE²⁴. También se expondrán los motivos que llevan a los investigadores a emplear esta tecnología en el ámbito de estudio que nos atañe, así como algunas de las carencias que han de subsanarse. Finalmente, se darán a conocer algunos de los trabajos más representativos en este área.

3.5.1 Breve reseña histórica

Los primeros esfuerzos relacionados con la conceptualización de los agentes se llevaron a cabo a principios de los ochenta. Desde entonces, el paradigma de los MAS se ha empleado en una gran variedad de dominios de aplicación. Dado que los agentes son entidades autónomas diseñadas para interactuar entre sí, el primer enfoque para crear un mecanismo que facilitase la interacción en la solución del problema fue el protocolo “Contract-Net”, originalmente desarrollado por Smith R.G. (Smith, 1980), que ha sido empleado en la creación de VE por varios grupos de investigadores (Rocha y Oliveira, 1999; Li *et al.*, 2000; Ouzounis, 2001; etc.).

Otros desarrollos potencialmente relevantes para la creación de la empresa virtual incluyen:

- *Agentes reactivos*. Son agentes capaces de tomar decisiones en respuesta a ciertos eventos, sin que exista ninguna interferencia externa procedente de los humanos.
- *Agentes móviles*. La movilidad representa la habilidad por la cual los agentes pueden viajar a través de redes heterogéneas para ejecutarse autónomamente y minimizar el intercambio de información. Los agentes móviles introducen una menor dependencia sobre la red dado que pueden operar de manera autónoma.
- *KQML*. Se trata de uno de los primeros lenguajes de comunicación de agentes. Es, además, un formato de mensaje y un protocolo de gestión de mensajes para soportar el intercambio de conocimiento entre agentes en tiempo de ejecución (Finin *et al.*, 1993).

24 En los capítulos siguientes se realizará un estudio más detallado de la tecnología multiagente enfocado tanto a justificar su adecuación al problema que nos ocupa como a seleccionar las herramientas, metodologías y estándares que permitirán construir la aplicación informática con la que validar y contrastar nuestro modelo.

- *FIPA*. Sus especificaciones representan un conjunto de estándares destinadas a promover interacciones de agentes heterogéneos y los servicios que pueden representar. El lenguaje *ACL* (*Agent Communication Language*) permite la comunicación entre los agentes a través de mensajes estructurados y mediante diferentes protocolos y convecciones suministrados por FIPA.
- *Subastas*. De manera similar a lo que sucede en los mercados mundiales, las subastas en los mercados multiagente representan el proceso de compra-venta de bienes y servicios en los que son ofrecidos para ser subastados (se reciben ofertas por ellos y finalmente se venden al mejor postor).

La figura 3.3 sintetiza estos y otros mecanismos relacionados con los agentes y sus aplicaciones tanto a la selección de socios como a la creación de empresas virtuales.

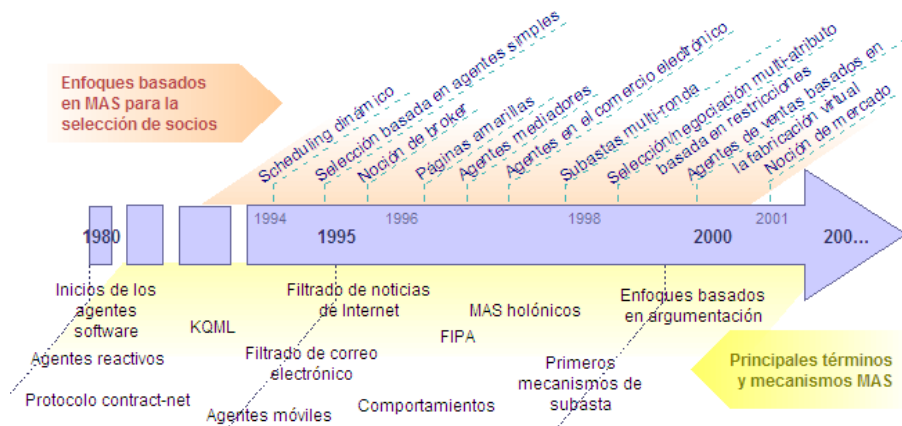


Figura 3.3. Enfoques basados en sistemas multiagente para la selección de socios frente a los principales mecanismos MAS (Afsarmanesh, 2006b).

3.5.2 Repaso a la utilización de los MAS en la selección de socios

Aunque fue en 1993 cuando Fox, Chionglo y Fadel propusieron por primera vez el uso de agentes en las cadenas de suministro (Fox *et al.*, 1993), el primer acercamiento de los MAS a la empresa virtual fue realizado en (Fisher

et al., 1996) en el proyecto AVE. Este proyecto pretendió suministrar una descripción sobre cómo se podían utilizar los agentes en la formación de una empresa virtual. Se centraba fundamentalmente en los aspectos relacionados con el establecimiento de la VE, aunque también abordaba algunos relacionados con la fase de operación. Sus autores describieron la selección de socios como un proceso de emparejamiento de las metas (o submetas) de la empresa virtual a los procesos parciales de las diferentes entidades que la componen.

A pesar de tratarse de un acercamiento muy genérico, sirvió para establecer algunas de las bases para la aplicación de los sistemas multiagente en estos entornos. Así, sus autores propusieron la necesidad de un mercado virtual electrónico donde las diferentes empresas pudiesen anunciar y obtener información variada (por ejemplo, atender las solicitudes de recuperación de información, responder a peticiones mediante ofertas adecuadas, etc.). También presentaron algunas consideraciones sobre los mecanismos de subasta necesarios en la formación de las empresas virtuales aunque, como ya se ha precisado anteriormente, se trataba de mecanismos muy simples.

En 1998 Ambroszkiewicz, Cetnarowicz y Radko (Ambroszkiewicz *et al.*, 1998) proponen un modelo abstracto de “mundo interconectado” constituido por una red de ordenadores donde las estaciones de trabajo desempeñan el papel de factorías de computación que poseen una serie de *inputs* y *outputs* específicos. Cada una de estas factorías están gestionadas por un agente denominado *factory agent*, que se encarga de crear los agentes móviles autónomos que realizarán el trabajo de formación y reconfiguración de la VE. Sin embargo, se trata de un modelo muy simplista en el que las decisiones se toman buscando minimizar una función de precios en la que se tiene en cuenta el precio de los *inputs*, los costes de producción y los beneficios netos.

También en 1998 podemos encontrar un primer ejemplo de la aplicación práctica de los sistemas multiagente a la creación de la VE en el contexto de los *clusters* industriales en (Rabelo *et al.*, 1998 y 2000). El *cluster*, constituido por doce compañías de la industria del molde, estaba representado legalmente mediante una entidad *broker* que ayudaba a un responsable humano experto a buscar y analizar oportunidades de negocio. Cualquier oportunidad de colaboración se transformaba, por medio de un agente *broker*, en un proceso de negocio distribuido que se repartía a las empresas potencialmente interesadas dentro del *cluster* cuyas competencias concordasen con la oferta.

Al final de todo el proceso, se formaba un conjunto de posibles equipos (es decir, empresas virtuales potenciales) que podían llevar a cabo tal oportunidad y se seleccionaba el más adecuado en base a las competencias y a los requerimientos de *scheduling*. La figura 3.4 ilustra la formación de un grupo de equipos de empresas dentro del *cluster* para atender un proceso de negocio distribuido concreto. En el ejemplo hay tres empresas virtuales (VE_1 , VE_2 , VE_3) capaces de llevar a cabo el proceso de negocio aunque, como se puede apreciar, es el equipo VE_1 el que finalmente se ha seleccionado.

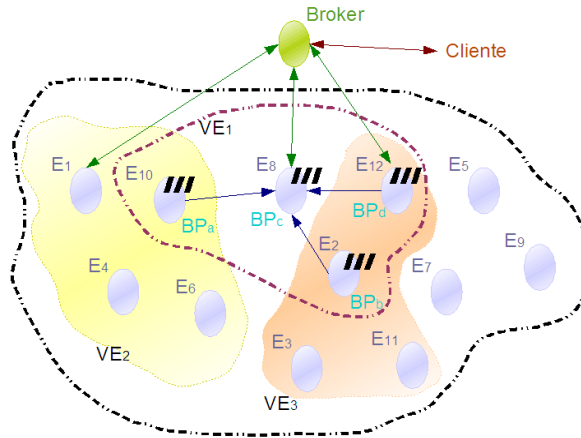


Figura 3.4. Múltiples VE dentro de un *cluster*.

Uno de los aspectos distintivos de este enfoque (MASSIVE) es su filosofía híbrida/semi-automática (Camarinha-Matos y Afsarmanesh, 2003) en la que los agentes se utilizan solamente como asistentes de los humanos. Dado que la decisión de selección última la van a tomar los expertos humanos, estos autores desarrollaron un sistema multiagente, el *MASSIVE Mould Broker System*, para ayudar al *broker* humano tanto en la gestión como en el proceso de selección de aquel conjunto de empresas que mejor responda a la oportunidad de negocio existente.

Cuando no se encontraban socios en el *cluster*, se utilizaba una herramienta de búsqueda y selección de socios que permitía encontrar candidatos externos en base a directorios de empresas disponibles en Internet. No obstante, el hecho de que el enfoque MASSIVE se centre en los *clusters* de empresas, es decir, en un universo multiagente controlado, la hace una solución bastante factible desde el punto de vista de la implantación.

Siguiendo con la arquitectura *broker*, Harbilas *et al.* (Harbilas *et al.*, 2000) emplean la noción de *broker* de información, que representa a una entidad encargada de almacenar información relativa a las empresas. De esta forma, aquellas empresas que buscan socios para constituir una VE pueden obtener información sobre los socios potenciales más adecuados a través del agente *broker*, ya que dicho agente es el que se encarga de comparar las metas (submetas) para encontrar los socios adecuados.

Por otro lado, también se ha sugerido el uso de enfoques federados (Shen y Norrie, 1999; Camarinha-Matos y Afsarmanesh, 2001) como medio para soportar arquitecturas de tipo *broker*. Se trata de un enfoque para la interoperación de los agentes donde éstos se organizan en los denominados sistemas federados. Existe un agente denominado “agente facilitador”, encargado de permitir la comunicación entre los agentes de modo que, en lugar de comunicarse directamente entre sí, los agentes se comunican a través de dicho facilitador.

Otro ejemplo significativo y bastante realista de la aplicación de los sistemas multiagente a la formación de empresas virtuales es el descrito en (Rocha y Oliveira, 1999; Rocha y Oliveira, 2001). Estos autores consideran un mercado virtual donde las empresas, representadas mediante agentes que están geográficamente distribuidos y que posiblemente carecen de un conocimiento previo del resto, pueden encontrarse y negociar para llegar a ser parte de la empresa virtual dinámica. Para ello, desarrollan un sistema en el que se hace uso de un protocolo Contract-Net (Davis y Smith, 1983) multironda y en el que los agentes tienen la capacidad de aprender de las experiencias pasadas.

Para modelar el mercado electrónico que ha de soportar la formación de la empresa virtual, Rocha y Oliveira proponen una arquitectura MAS en la que, además de los agentes que representan a las empresas que están interesadas en formar parte de la VE (*enterprise agent*), existe un agente de mercado (*market agent*) que es creado e insertado en la comunidad MAS siempre que aparece una oportunidad de negocio. Este agente desempeña el papel de coordinador en el mercado electrónico, siendo su principal objetivo la formación de la empresa virtual.

El agente de mercado se encarga de enviar invitaciones a los socios potenciales para cada una de las subtareas de la VE. Una vez recibidas las invitaciones, los agentes interesados formulan ofertas en base a sus propias

capacidades y se las envían al *broker*. Este agente de mercado evalúa las ofertas y selecciona las más favorables en base a mecanismos multi-criterio y a una negociación basada en restricciones. Por tanto, y dado que las ofertas se hacen para cada una de las subtareas, este enfoque no tiene en cuenta las relaciones e interdependencias que puedan existir entre ellas.

Entre los diferentes criterios que consideramos encontramos el de menor coste, mayor calidad, mayor disponibilidad, etc. A cada uno de estos criterios se les asignan valores de utilidad y se emplea una combinación lineal de los valores de los atributos ponderados por sus correspondientes valores de utilidad. Pueden llevarse a cabo múltiples rondas de negociación, aunque al final de cada ronda a los diferentes postores se les indica si sus ofertas/propuestas son ganadoras o perdedoras, junto con una justificación cualitativa y robusta que les permita reajustar sus propuestas.

Un trabajo similar se puede encontrar en (Li *et al.*, 2000) donde se realiza un análisis más detallado del problema de descomposición de los objetivos de la VE, lo que conduce a su jerarquización. Además de los agentes que representan a las empresas y del agente coordinador (*broker*), se introduce un agente servidor de información que mantiene información pública relacionada con las reglas operacionales y organizacionales comunes, el entorno de mercado, las empresas y productos/servicios suministrados, etc. En este trabajo se introduce explícitamente la necesidad de una ontología común que soporte la comunicación entre los agentes y se implementa un proceso de negociación multiatributo basado en restricciones.

En (Davidrajuh y Deng, 2000) se propone una arquitectura basada en agentes móviles para la formación de empresas virtuales²⁵. En concreto, estos agentes son enviados por el creador de la empresa virtual a las *web sites* de los socios potenciales para recoger datos, comprobar sus competencias y realizar una primera selección *in-situ* (un primer filtrado en base a un análisis cualitativo). Una vez vuelven al origen (al lugar desde donde fueron enviados) se realiza una evaluación/selección más precisa a partir de la información recogida por los agentes móviles.

25 Los agentes móviles ya habían sido utilizados anteriormente en otras aplicaciones, como es el caso de la formación de cadenas de suministro (Ambroszkiewicz *et al.*, 1998) o la integración de sistemas en las empresas virtuales (Clements *et al.*, 1997), etc.

Como parte del proceso de selección, también se realiza una valoración del desempeño cooperativo de los candidatos considerando su historial de cooperación. Sin embargo, de forma casi instantánea surge una pregunta crucial, ¿los socios potenciales van a mostrar su verdadera información privada a los agentes móviles del creador de la VE?

Ouzounis (Ouzounis, 2001) propone una arquitectura basada en tres capas: el entorno de procesamiento distribuido, la plataforma de agentes móviles y los servicios de soporte²⁶. La plataforma para la gestión de empresas virtuales dinámicas se construye en base a los conceptos de agentes móviles inteligentes (conformes con FIPA), una gestión del *workflow* para la ejecución y gestión de los procesos de negocio interorganizacionales basada en agentes, los mercados virtuales (*virtual market place*) y la negociación automatizada para la selección de socios.

El modelo de negocio (figura 3.5) para la plataforma está constituido por diferentes dominios administrativos que tienen relaciones específicas entre ellos. Los dominios involucrados son el *cliente* o entidad que precisa el servicio que la empresa virtual ofrece; el *representante*, que suministra el servicio al cliente y representa a la VE en el exterior; el *mercado virtual*, que proporciona los servicios de registro y selección a los candidatos; y los *socios* y *candidatos*, que son las entidades que registran sus ofertas de servicio en el mercado y negocian con otros socios bajo demanda para establecer relaciones de negocio.

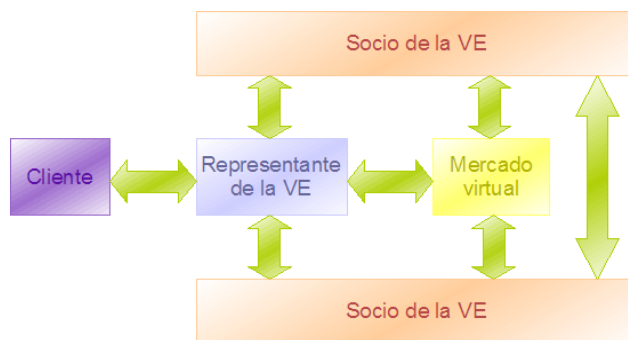


Figura 3.5. Modelo de negocio de Ouzounis (Ouzounis, 2001).

26 Propone una 4ª capa de especificación, registro y gestión de los BP y del mercado virtual.

El mercado virtual representa el dominio que proporciona los servicios de emparejamiento: permite a los candidatos registrar y administrar las ofertas de servicios, y al representante le permite buscar socios potenciales que puedan proveer los procesos de negocios particulares asociados a los tipos de servicios existentes. Para seleccionar al mejor socio capaz de suministrar un proceso remoto se recurre a un proceso de negociación basado en el protocolo Contract-Net, en el que los agentes intercambian mensajes ACL empleando una ontología de negociación específica.

Petersen (Petersen, 2003; Rao y Petersen, 2003; Petersen *et al.*, 2002) también propone un acercamiento basado en agentes para soportar la formación rápida y eficiente de empresas virtuales. Este acercamiento utiliza agentes software para representar a los socios de una empresa virtual y consta de tres elementos: un modelo de empresa virtual basado en agentes, el proceso de formación de la VE y una arquitectura multiagente para apoyar el proceso de formación. El modelo de VE permite definir los conceptos principales de una empresa virtual (figura 3.6): metas, actividades, roles, requisitos y agentes, y las relaciones entre ellos.

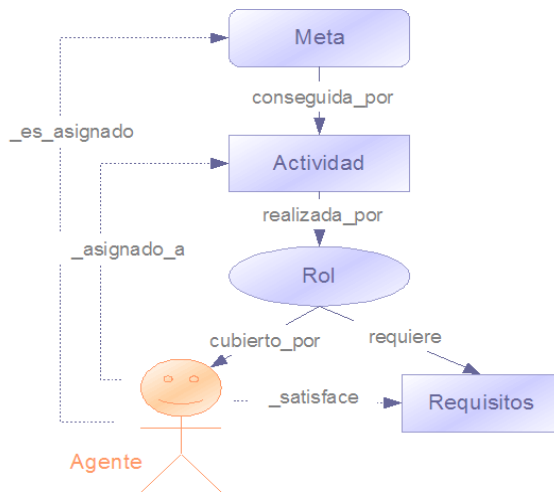


Figura 3.6. Vista del metamodelo definido por Petersen (Petersen, 2003).

En el proceso de formación, que se da dentro del contexto de un mercado electrónico, los socios evolucionan a través de tres etapas: socios interesados, socios potenciales y socios de la VE. El iniciador anuncia una VE y solicita

ofertas de aquellos agentes interesados en llegar a ser socios de la VE. Estos agentes proponen ofertas cuyo contenido corresponda a los requerimientos expresados en el anuncio, y el iniciador las evalúa en base a criterios de valoración previamente definidos.

Tras el proceso de evaluación, el agente iniciado informa a aquellos agentes cuyas ofertas han sido rechazadas y negocia con aquellos cuyas ofertas satisfacen los requerimientos (socios potenciales). Finalmente, se establece un contrato con el mejor socio potencial (o los mejores socios potenciales). Este proceso puede ser considerado como un conjunto de interacciones entre los agentes y, en consecuencia, puede ser representado como un protocolo de interacción de agentes²⁷.

Finalmente, se adopta la arquitectura multiagente AGORA para soportar el proceso de selección de socios y la formación de la VE. Se trata de una infraestructura basada en el mercado que permite a los agentes encontrar e intercambiar información. Además, AGORA facilita la colaboración entre los agentes soportando la negociación y coordinación. Entre los elementos principales encontramos el nodo Agora, encargado de facilitar la comunicación, coordinación y negociación entre los agentes, el gestor Agora, que realiza funciones de gestión generales, el coordinador, para soportar un comportamiento coherente entre el nodo y los agentes, y el negociador, responsable de la resolución de conflictos vía negociación.

Viet D. Dang considera en (Dang, 2004; Norman *et al.*, 2004) que el proceso de selección de socios se logra mejor empleado alguna forma de estructura de mercado, y más en concreto alguna forma de subasta, entendiendo por tal una institución de mercado con un conjunto explícito de reglas que determinan tanto la localización de los recursos como los precios en base a ofertas de los participantes en el mercado. Su enfoque consiste en utilizar mecanismos de subasta combinatoriales para automatizar la selección y desarrolla nuevos algoritmos que permiten a los agentes encontrar soluciones más rápidas y mejores que las desarrolladas hasta ese momento.

Aerts *et al.* (Aerts *et al.*, 2002) presentan una arquitectura basada en agentes móviles. La arquitectura ICT que se propone se compone de varios servicios generales (para comercializar, secuenciar y ordenar), agentes y componentes

27 La representación Agent UML (Bauer *et al.*, 2001) de dicho protocolo se puede contemplar en (Petersen, 2003).

de integración local (denominados *docks*). Esta arquitectura permite a las empresas participantes centrarse en sus competencias esenciales mientras toman parte en una DVE para una familia de productos donde se necesitan competencias adicionales. Además, son libres de elegir participar en iniciativas similares de otras empresas en la VE. Una vez que se ha realizado el esfuerzo inicial de mapear la ontología de la empresa con la ontología de la VE, el esfuerzo para unirse a una VE es exiguo.

En (Malucelli *et al.*, 2005; Cardoso *et al.*, 2005; Cardoso *et al.*, 2006; Rocha *et al.*, 2005) se propone el concepto de Institución Electrónica (EI, *Electronic Institution*), que representa el homólogo virtual de las instituciones del mundo real, y entre cuyos beneficios se encuentra la generación de un entorno fiable y regulado para hacer cumplir las normas y para proporcionar servicios institucionales específicos a los agentes. Estos servicios, basados en ontologías, favorecen la interacción entre los agentes, permitiendo que el establecimiento de acuerdos de negocio sea más eficiente y rápido, contribuyendo así a agilizar la respuesta de la empresa virtual.

El objetivo principal de estos autores consiste en desarrollar una EI basada en agentes que proporcione un entorno normativo virtual, que asista y regule la creación y operación de organizaciones virtuales a través de servicios relacionados con la contratación. Con este propósito definen dos servicios principales, que son los denominados “servicio de mediación en la negociación” y “servicio de seguimiento del contrato”. Además de estos, también podemos encontrar el servicio de conversión monetaria, el de conversión de unidades de medida, el servicio de emparejamiento de términos, etc. Para la implementación de todos estos servicios estos autores hacen uso de la plataforma JADE.

En general, los ejemplos anteriores de aplicación de MAS a la VE dinámica adolecen de una serie de limitaciones que condicionan su implantación práctica (Aerts *et al.*, 2002; Camarinha-Matos y Afsarmanesh, 2003; Malucelli *et al.*, 2005):

- En primer lugar, no existen estándares software comunes. A pesar de que entidades como FIPA trabajan en la construcción de dichos estándares, la realidad muestra que es necesaria la existencia de un suministrador de agentes común que permita preparar un entorno de agentes fiable y seguro en el que estas entidades puedan desarrollar su actividad.

- La ausencia de ontologías comunes²⁸ es otra limitación importante. Debe construirse una ontología de VE que contenga las definiciones tanto de los procesos de negocio como de las interfaces de componentes dentro de las familias de productos.
- Ninguna de las propuestas tiene en cuenta aspectos más subjetivos tales como la confianza, compromisos, historial de cooperación exitosos, etc., que, como sabemos, pueden llegar a jugar un papel crucial en la elección de los socios de la empresa virtual.
- Además, ponen poca atención en los aspectos de implantación y gestión del mercado y de las páginas amarillas.
- No se consideran aspectos de seguridad en el proceso de negociación, un punto crítico cuando los agentes (que representan a las empresas) son solamente parcialmente cooperativos (estos pueden ser auto interesados, competitivos e incluso exhibir un comportamiento antagónico).
- El intento de alcanzar un proceso de toma de decisiones totalmente automatizado, aunque es un ejercicio académico interesante, es bastante irrealista en este dominio de aplicación. En este sentido, la combinación de la tecnología multiagente con los sistemas expertos pueden constituir una aproximación adecuada.
- Por último, como los agentes se diseñan y desarrollan independientemente, es bastante difícil garantizar un elevado grado de confianza y coordinación a menos que se adopten reglas comunes.

3.6 EL ENFOQUE FEDERACIÓN DE SERVICIOS

3.6.1 ¿Qué es un servicio web?

Con el lanzamiento oficial de Visual Studio .NET, Microsoft presentó una gran innovación en la evolución de aplicaciones distribuidas para Internet, que denominó Web Service (Velásquez, 2002). Los Servicios Web representan, en opinión de Lizárraga (Lizárraga, 2002), una revolución al alejarse de las arquitecturas tradicionales tipo cliente-servidor a nuevas arquitecturas distribuidas tipo igual-a-igual (*peer-to-peer*).

28 Ambas limitaciones, la falta de ontologías y estándares comunes, representan una situación difícil de superar en un universo abierto de empresas. Sin embargo, no ocurre lo mismo si consideramos la creación de las DVE dentro del contexto de los VBE.

Pero, ¿qué es un *Web Service*?. De acuerdo con el W3C (*World Wide Web Consortium*) (W3C web, 2006), es un sistema software diseñado para soportar interacciones máquina a máquina sobre una red. Posee una interfaz descrita en un formato que la máquina puede procesar, WSDL, de tal manera que otros sistemas pueden interactuar con el *web service* de una manera preestablecida (en base a su descripción) usando mensajes SOAP que normalmente son transportados usando HTTP con una serialización XML en conjunción con otros estándares relacionados con la web.

De manera resumida, un servicio Web es una interfaz que describe una colección de operaciones a las que se puede acceder a través de una red (o de Internet) mediante mensajes XML estandarizados (Hao *et al.*, 2003). La interfaz oculta los detalles de implementación del servicio, lo que permite que pueda ser utilizado con independencia tanto de la plataforma hardware o software como del lenguaje de programación que se emplee. La figura 3.7 muestra una vista simplificada de la arquitectura *web service*.

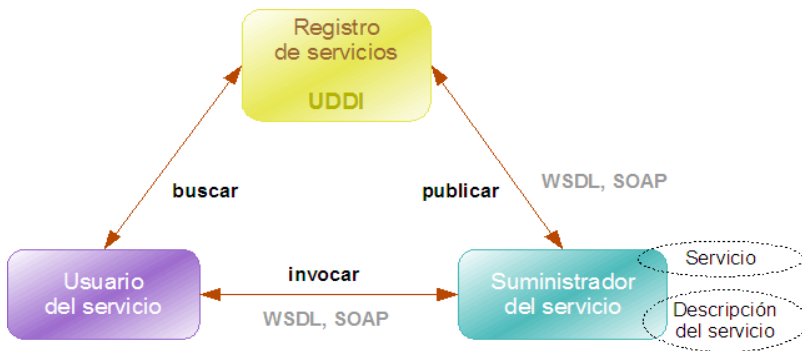


Figura 3.7. Arquitectura básica de los servicios web. Adaptada de (Hao *et al.*, 2003).

La interacción entre servicios web involucra a cuatro componentes. Para comenzar, los *suministradores de servicio* ofrecen *servicios*, los describen y los publican en el *registro de servicios*, que proporciona un repositorio de servicios junto con descripciones detalladas de los mismos. Finalmente, los *usuarios de los servicios* (aquellas aplicaciones que requieren que se satisfaga alguna función) usan motores de búsqueda para localizar los servicios demandados. El registro de servicios suministra al usuario descripciones de los servicios relevantes que posteriormente utilizará para invocar el servicio web más apropiado.

Un hecho significativo de los servicios web es que utilizan una comunicación basada en el intercambio de documentos XML (*eXtensible Markup Language*). En este sentido, existen tres iniciativas principales de estandarización para permitir interacciones entre servicios web (Sanz, 2004; Lizárraga, 2002):

- WSDL (*Web Services Description Language*) es un lenguaje basado en XML empleado para describir, de forma estructurada, la funcionalidad de un servicio web así como la forma en que dicha funcionalidad se hace disponible. Entre la información que incluye el archivo WSDL se encuentra el nombre del servicio web, los nombres de los métodos, sus argumentos, etc.
- SOAP (*Simple Object Access Protocol*) define un formato de documento XML que describe cómo invocar los métodos remotos del servicio web. Se trata, en definitiva, de un simple mecanismo de mensajería para el intercambio de datos estructurados. SOAP consta de una envoltura XML (que define un marco de referencia general para expresar qué hay en el mensaje, quién debe atenderlo, y si es opcional u obligatorio), las reglas de codificación, una convención que puede ser utilizada para representar las llamadas y respuestas de procedimientos remotos, y un cuerpo que incluye los datos a intercambiar.
- UDDI (*Universal Description, Discovery, and Integration*): es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que éstas ofrecen. A través de UDDI, se puede publicar y descubrir (a través de consultas) información de una empresa y de sus servicios.

3.6.2 Aplicación a la selección de socios en la DVE

La aplicación de este paradigma a la VE dinámica ha conducido a un nuevo planteamiento denominado “enfoque basado en la federación de servicios” (Malucelli *et al.*, 2005). De acuerdo con este modelo, las compañías (es decir, los miembros potenciales) son vistos como “proveedores de servicios”, lo que supone que el comportamiento colaborativo potencial de cada compañía se materialice a través de un conjunto de servicios. El enfoque asume la existencia de una entidad, conocida como “portal o nodo promotor de servicios”, que mantiene un catálogo de servicios donde las compañías suministradoras publican sus ofertas.

Los servicios se publican en el catálogo a través de un registro de “especificación de servicios” que incluye un *proxy* o representante del servicio para permitir una invocación remota y transparente del servicio. Las compañías clientes pueden usar la funcionalidad que proporciona el gestor del catálogo para “descubrir” los servicios existentes. Una vez se selecciona un servicio, el cliente obtiene una copia del *proxy* correspondiente y puede invocarlo remotamente de una manera transparente (sin preocuparse de su localización actual o incluso de su lenguaje de implementación). Se deduce, por tanto, que este enfoque conduce a una selección de socios indirecta ya que lo que se selecciona es el servicio, no su suministrador: los socios son elegidos implícitamente a través de los servicios seleccionados.

FETISH-ETF (Barata, 2005) es, posiblemente, el principal ejemplo de aplicación del enfoque federación de servicios al establecimiento de una infraestructura que soporte la creación y operación de empresas virtuales en el sector turístico. Este trabajo considera que el servicio de turismo se puede proporcionar a dos niveles. El nivel inferior contiene los servicios básicos, que representan los servicios elementales creados por un operador turístico. También es posible crear otro nivel, por encima del anterior, en el que los servicios básicos se usan como componentes de un servicio de valor añadido (VAS, *Value Added Service*).

Un ejemplo de VAS puede ser un paquete vacacional. Su materialización involucra un proceso de negocio distribuido que requiere (Barata, 2005) la invocación y coordinación de otros servicios que pueden ser suministrados por diferentes proveedores, lo que supone la formación de una empresa virtual. FETISH-ETF proporciona una herramienta para componer VAS, especificar el *workflow* correspondiente y seleccionar los proveedores de servicios de un catálogo. Una selección de un conjunto de proveedores de servicios que satisfacen las necesidades de un VAS particular conduce a la formación de una empresa virtual *ad hoc* para atender la oportunidad de negocio.

Sin embargo, y a pesar de su reciente popularidad, existen multitud de limitaciones para aplicar eficientemente este paradigma a la selección de socios en las VE dinámicas (Camarinha-Matos, 2003): ¿están los servicios siempre disponibles?, ¿cuál es nivel de conocimiento del proveedor del servicio?, ¿pueden las tareas ser representadas como servicios?, ¿tiene sentido aplicar servicios específicos para la selección de socios?, ¿y para el proceso de negociación?, etc. Además, la mayoría de los desarrollos hasta la

fecha se han realizado para el comercio electrónico y no para actividades colaborativas (Camarinha-Matos *et al.*, 2005), existe una pobre integración de los mecanismos de seguridad y privacidad, y los mecanismos de búsqueda son todavía muy escasos.

Con respecto a los mecanismos de búsqueda, el servicio de registro de UDDI utiliza una taxonomía no estandarizada que no clasifica el WSDL de acuerdo a sus operaciones (Petrie y Bussler, 2003). Por tanto, si se está buscando un servicio web específico, por ejemplo con un agente software automatizado, se tiene que consultar la descripción WSDL del servicio, lo que genera más tráfico de red y un mayor tiempo y coste de procesamiento. Claramente, se hace necesaria una ontología estándar de términos comunes que proporcione la semántica para las operaciones y los mensajes de servicio, y que permita la búsqueda automatizada de los servicios web, un requisito indispensable para favorecer su aplicabilidad en las VE dinámicas.

3.6.3 Unión MAS - web services

Recientemente han aparecido diferentes trabajos en los que se intentan integrar ambos paradigmas (Petrie y Bussler, 2003; Hao *et al.*, 2005; Malucelli *et al.*, 2005; Matskin *et al.*, 2005; Namin *et al.*, 2006; Svirskas *et al.*, 2007; etc.) en un intento de obtener mejores alternativas a algunos de los problemas que rodean a las organizaciones virtuales y, en particular, a las VE dinámicas. En todas ellas se considera que los servicios web representan un paradigma de computación prometedor para lograr la integración de aplicaciones a través de Internet y, por lo tanto, para facilitar la formación de empresas virtuales entre diferentes plataformas software y hardware.

Por ejemplo, Petrie y Bussler (Petrie y Bussler, 2003) además de justificar la necesidad de integrar ambas tecnologías, describen diferentes herramientas y tecnologías que permiten unir ambos paradigmas en la creación de soluciones adaptadas a las necesidades de las empresas virtuales, y en particular las DVE. Hao *et al.* (Hao *et al.*, 2005) proponen un marco para la gestión de la fabricación cooperativa basado en los servicios web y los agentes software, y posibilitado por Internet. Diseñan y desarrollan un entorno de integración formado por un sistema de coordinación basado en agentes, un sistema de control y monitorización basado en la Web, y los correspondientes servicios para la gestión, monitorización y control de un *shop floor*.

Malucelli *et al.* (Malucelli *et al.*, 2005) introducen un entorno normativo institucional basado en el concepto de realidad institucional en el que los sistemas multiagente se ven complementados por una serie de servicios institucionales suministrados a través de *web services*. Namin *et al.* (Namin *et al.*, 2006) proponen un modelo mixto para facilitar la colaboración entre empresas. En concreto presentan un modelo multiagente que se ajusta a los diferentes niveles de la arquitectura de sistemas de la empresa para lograr una adecuada selección de los servicios registrados, para comprobar el estatus de un proceso dado, para realizar consultas de los usuarios y para reaccionar a ellas de una forma colaborativa con otros servicios web basados en agentes. Además, proponen un modelo multiagente capaz de coordinar y monitorizar los procesos de *workflow*.

Resulta evidente que las propuestas y alternativas planteadas en estos trabajos todavía han de recorrer un largo camino antes de llegar a convertirse en realidades fehacientes. No obstante, no cabe duda de que representan el comienzo de una nueva línea de investigación encaminada a unir las ventajas de ambos paradigmas y a mitigar sus carencias. A pesar de esta realidad, todavía son múltiples y variadas las limitaciones que plantea la tecnología actual de los servicios web en relación con los procesos que se han de llevar a cabo en la formación y operación de las empresas virtuales dinámicas, lo que limita considerablemente su aplicabilidad en este dominio.

3.7 RESUMEN

La VE dinámica representa un modelo de negocio en el que un conjunto de entidades, legalmente independientes y geográficamente dispersas, se asocian para aprovechar oportunidades del mercado, permaneciendo unidas mientras dichas oportunidades persistan. Asociados a este paradigma existen una serie de beneficios: *agilidad* (habilidad para reconocer, reaccionar y hacer frente rápidamente a los cambios impredecibles que se dan en el entorno) que permite dar una mejor respuesta a las oportunidades, lograr un *menor time-to-market* y una *mayor calidad* pero con una menor inversión; *competitividad* (eficiencia en costes y en tiempo de respuesta); *optimización de los recursos*, mayor *innovación*, se logra *alcanzar dimensión sin ganar peso*, etc.

A pesar de que estas ventajas son bien conocidas a nivel conceptual, la implantación práctica de la DVE está todavía lejos de ser una realidad. Son muchos los obstáculos a superar en todas y cada unas de las fases del ciclo de

vida de las VE dinámicas, entre las que destaca la falta de herramientas e infraestructuras de soporte adecuadas para, entre otros propósitos, la búsqueda y selección de socios, negociación, formalización de contratos, gestión de recursos y competencias, distribución, monitorización y coordinación de las tareas, valoración del desempeño, etc.

Además, se carece de ontologías comunes entre las organizaciones cooperantes, no hay una cultura de cooperación, se precisa bastante tiempo para los procesos de construcción de la confianza, es necesario una reingeniería de los procesos de negocio, así como una formación adecuada de los trabajadores, etc. Se precisa, por tanto, un gran esfuerzo en investigación que permita encontrar soluciones apropiadas a estos y otros problemas y, en consecuencia, que permita hacer de este modelo cooperativo una realidad.

A lo largo de los últimos años Europa se ha convertido en un referente de dicha investigación motivado, principalmente, por dos aspectos inherentes a su realidad: por un lado, el actual proceso de integración que existe dentro de la Unión Europea, que representa un empuje hacia la “cultura de cooperación” y, por otro, el escenario económico europeo, constituido principalmente por empresas de pequeño y mediano tamaño que necesitan unir sus esfuerzos si quieren ser competitivos en unos mercados tan abiertos y turbulentos como los actuales. Sin embargo, el esfuerzo por establecer las bases tecnológicas como las prácticas operativas necesarias para soportar las empresas virtuales dinámicas no es sólo patrimonio del continente europeo: podemos encontrar actividades en este sentido en USA, Brasil, Canada, Japón, China, etc.

Durante este capítulo hemos pretendido mostrar lo más relevante de la investigación realizada hasta el momento entorno a la empresa virtual y, más en concreto, alrededor del proceso de identificación y selección de socios. Con este propósito hemos analizado algunos de los proyectos I+D+i más relevantes y hemos intentado mostrar los fundamentos básicos y los trabajos más representativos de los tres enfoques principales que se han utilizado para afrontar el problema.

De hecho, se han presentado trabajos en los que se discute la selección de socios desde un enfoque cuantitativo y en los que se proponen diferentes algoritmos de optimización, se ha realizado un análisis pormenorizado de la aplicación de la tecnología multiagente al problema que nos atañe y, finalmente, se ha explicado someramente el paradigma de los servicios web,

presentando algunos ejemplos de aplicación e introduciendo lo que sin duda son los primeros pasos hacia la integración de esta tecnología con la multiagente.

En conclusión, la investigación llevada a cabo nos permite confirmar que los sistemas multiagente constituyen, al menos en la actualidad, la forma más acertada de abordar el problema de la selección de socios en la VE dinámica.

CAPÍTULO

4

EL MODELO Y LA JUSTIFICACIÓN DE LA TECNOLOGÍA

4.1 INTRODUCCIÓN

De entre todos los modelos organizativos propuestos hasta la fecha, el arquetipo dinámico de empresa virtual es, con diferencia, el paradigma que mejor puede responder a las necesidades de supervivencia de las empresas actuales. Sin embargo, existen todavía un gran número de obstáculos que impiden su aplicación práctica a gran escala; entre esos problemas destacan, y muy significativamente, los derivados de la selección de socios. Esta es, sin duda, la razón principal que explica el porqué numerosos investigadores de diferentes áreas de conocimiento han dirigido sus esfuerzos en esta dirección a lo largo de los últimos años.

Al abrazar esta motivación como propia nos vemos obligados a ofrecer una solución innovadora que responda a las demandas de la VE dinámica de forma eficiente. Pero para poder alcanzar dicha solución primero es necesario

disponer de un modelo de negocio adecuado que tenga en cuenta las necesidades de la DVE y todas las consideraciones formuladas en los capítulos previos. Sólo así seremos capaces de desarrollar una herramienta informática que apoye la toma de decisiones de un modo realmente innovador y efectivo.

En consecuencia, nuestro primer objetivo a lo largo del presente capítulo no será otro que la definición de ese modelo de negocio (sección 4.3). A partir de él seremos capaces de construir una plataforma software con la que verificar si el modelo desarrollado constituye un punto de partida acertado para resolver el problema de la selección de socios. Sin embargo, antes de iniciar la confección de dicho modelo resulta adecuado realizar algunas consideraciones previas sobre los procesos de negocio y sobre las soluciones existentes en la literatura (sección 4.2), lo que nos permitirá comprender mejor el modelo guía.

Una vez definido el modelo operativo será el momento de empezar a conocer algunas de las generalidades propias de los paradigmas software que emplearemos: los sistemas multiagentes (sección 4.4) y los sistemas expertos (sección 4.5). No pretendemos suministrar un análisis exhaustivo de la historia y teoría de los agentes ni presentar las diferentes áreas de aplicación de este paradigma, ya que toda esta información puede encontrarse en multitud de publicaciones y trabajos²⁹. Más bien, buscamos proporcionar unas primeras pinceladas que guíen el estudio de las herramientas y metodologías que se llevará a cabo en el próximo capítulo, así como argumentar la idoneidad de aplicar esta tecnología al paradigma de la DVE.

Objetivos similares nos establecemos para abordar el estudio de los sistemas expertos o sistemas basados en el conocimiento. La aplicación de esta tecnología en combinación con la multiagente supone una innovación a la hora de afrontar la problemática asociada a la selección de socios en VE dinámicas. Además, constituye un medio acertado para representar de forma más adecuada los procesos de decisión que se dan en las empresas reales. A fin de cuentas, gracias a los sistemas expertos vamos a ser capaces de dotar al sistema multiagente de un módulo de decisión que simule el comportamiento de los decisores humanos, haciendo el proceso más real.

29 Como, por ejemplo, (Wooldridge y Jennings, 1995; Iglesias, 1997 y 1998; Green et al., 1997; Ferber, 1999; Corchado y Molina, 2002; Wooldridge, 2002; Molina et al., 2004; Mas, 2005; etc.)

4.2 CONSIDERACIONES PREVIAS

4.2.1 Los procesos de negocio y su descomposición

Un aspecto significativo en la creación de cualquier tipo de empresa, y muy especialmente en la creación de una VE dinámica, es la especificación de las tareas que son necesarias para alcanzar su objetivo (Camarinha-Matos y Afsarmanesh, 2001). Las actividades que realiza cualquier compañía normalmente se organizan en grupos de actividades interrelacionadas denominadas procesos de negocio (BP, *Business Processes*). Estos BP pueden ser vistos como un conjunto de actividades, reglas y restricciones que especifican los pasos que deben darse, además de las condiciones que deben satisfacerse, para conseguir un objetivo.

Pongamos, por ejemplo, la recepción de un pedido por parte de un distribuidor de libros. Por lo general, este acontecimiento exige la recogida de los datos del cliente y del pedido, la comprobación de existencias en el almacén, realizar un pedido a otro distribuidor o editorial si fuese necesario, cobro del pedido y, finalmente, su envío al cliente. Estas tareas deben coordinarse de acuerdo a una determinada lógica, que se representa como un proceso de negocio.

DEFINICIÓN Y MODELADO DE LOS PROCESOS DE NEGOCIO

Definir un proceso de negocio consiste, esencialmente, en construir un modelo inteligible y procesable por aplicaciones software (Arias, 2005), que represente el proceso como una red de actividades y sus relaciones, criterios para indicar el inicio y finalización del proceso, e información acerca de las actividades individuales, tales como sus participantes, aplicaciones y datos asociados, etc. En la tabla 4.1 se muestran los conceptos más relevantes utilizados habitualmente en el modelado de los procesos de negocio (WfMC web, 2006).

Estos modelos se representan mediante lenguajes de definición de procesos de negocio. Desde principios de los noventa se han definido y utilizado gran cantidad de lenguajes de definición de procesos de negocio, y aunque han existido múltiples intentos por establecer algunos como estándar, ninguno de ellos ha cuajado por el momento. En la actualidad, las redes de Petri y los

diagramas de actividad de UML son los formalismos más utilizados para el modelado de procesos de negocio, aunque no los únicos.

Caso	Por cada cliente que realiza una petición, sea éste interno o externo, se crea una nueva instancia del proceso para atenderle. Cada una de estas instancias recibe del nombre de <i>caso</i> .
Tarea	Representa una unidad de trabajo. Se describe como una ordenación total o parcial de operaciones, descripciones de actividades humanas y otras tareas.
Actividad	Realización de una tarea para un caso concreto. Los conceptos de actividad y tarea a menudo se emplean como sinónimos.
Actor	Entidad que realiza tareas e interactúa con otros actores durante la ejecución del proceso. Los actores pueden ser personas o no serlo (máquinas, programas de software).
Papel (rol)	Conjunto de actores cuyas características los capacita para realizar una tarea o conjunto de tareas. Cada actor está adscrito a uno o más roles, y cada tarea específica a qué rol debe pertenecer el actor que la realiza.
Objeto manipulado	Documento, datos, objeto material, etc., sobre el que se actúa en las tareas, y que intercambian los actores durante la ejecución de un proceso. También se le conoce como entidad.
Interacción	Modela cómo interactúan entre sí las distintas tareas (ordenación en la ejecución de tareas, sincronización, intercambio de objetos, etc.)
Evento	Suceso, normalmente externo al proceso, que puede forzar en éste un cambio de estado (por ejemplo, la llegada de un mensaje, el vencimiento de una temporización o una determinada acción de un actor).

Tabla 4.1. Definición de los principales conceptos aplicados al modelado de procesos en base a actividades (Arias, 2005).

LOS PROCESOS DE NEGOCIO EN LA DVE

Cualquier proceso de negocio se puede descomponer en una jerarquía de subprocesos y actividades elementales de empresa. Estas últimas representan el nivel más bajo en la descomposición, es decir, son los bloques de construcción básicos que la empresa debe realizar para ejecutar sus procesos

de negocio. La composición de cada proceso de negocio es diseñada de modo que se pueda alcanzar una meta específica (en definitiva, una meta parcial); así, la unión de todas las metas parciales permitirá satisfacer el objetivo global de la empresa.

Cuando un proceso de negocio es ejecutado mediante una VE³⁰, algunos de sus subprocesos pueden ser asignados a diferentes empresas, por lo que el BP se convierte en un proceso distribuido (DBP, *Distributed Business Process*) (Klen, 1999). Una perfecta combinación de varios de estos procesos en los diferentes miembros de la VE dinámica es lo que conduce a la consecución del objetivo global de la DVE. Obviamente, la supervisión y coordinación de un DBP se complica notablemente cuando se involucran varias entidades autónomas, distribuidas y heterogéneas.

El coordinador de la empresa virtual es responsable de coordinar y supervisar la ejecución de los procesos de negocio distribuidos, mientras que los diferentes miembros que componen la empresa virtual dinámica son solamente responsables de la coordinación de los subprocesos de negocio que les hayan sido asignados. Obviamente, el coordinador de la empresa virtual también puede ser un miembro de ella y, como tal, también puede ser responsable de cualquier subproceso de negocio particular que le haya sido asignado.

Además, un miembro de la VE puede convertirse en coordinador de una DVE dentro de la empresa virtual inicial (una subDVE); deberá, entonces, coordinar/supervisar las partes de su subBP asignadas a diferentes socios. Bajo este enfoque, los subconsorcios se constituyen con el único propósito de facilitar la coordinación de las actividades involucradas en estos subBP. Una vez finaliza el subproceso, el subconsorcio correspondiente se disuelve y sus miembros pueden volver a involucrarse en otros subconsorcios que se formen dinámicamente en esta VE a medida que evolucione la ejecución del DBP.

Dentro de la VE dinámica es necesario determinar quién es el responsable de crear los modelos e instancias del BP. En este sentido, diferentes organizaciones de VE pueden considerar distintos actores ante este proceso y diversas reglas de coordinación (Camarinha-Matos y Lima, 2000). Algunas posibilidades son las siguientes:

30 En este capítulo emplearemos indistintamente las siglas VE y DVE para referirnos a los modelos dinámicos de empresa virtual.

- *Planificación centralizada.* El coordinador planea todo el BP y se lo envía a los miembros. Es necesario, por tanto, tener en cuenta los derechos de visibilidad de los socios, es decir, decidir si un miembro puede ver todo el plan o sólo la parte de la que es responsable.
- *Planificación adaptable.* Consiste en establecer una definición abstracta del proceso de negocio e ir detallándola en función de las condiciones específicas de cada uno de los miembros potenciales.
- *Planificación cooperativa.* En este caso son varios miembros los que planifican cooperativamente el BP. Por tanto, debe existir un espacio de planificación compartido para el diseño del modelo de BP y unos mecanismos de negociación para la resolución de conflictos.
- *Planificación jerárquica.* El socio principal define un modelo abstracto del BP en el que se planifican solamente los primeros niveles de descomposición (este socio principal suele ser la empresa que identifica la oportunidad de negocio, es decir, el *broker* o coordinador de la DVE). El nivel de detalle de este modelo es el suficiente como para permitir la identificación de los socios y habilidades necesarias, los principales recursos y la distribución de los subprocesos de negocio entre estos socios. En consecuencia, cada miembro es responsable de refinar los subprocesos de negocio que se le han asignado de acuerdo con sus capacidades locales.

Tras definirse y secuenciarse el proceso de negocio global (el objetivo de la empresa virtual dinámica), y tras asignar a cada socio las responsabilidades correspondientes, alcanzar con éxito el objetivo común (el reparto del producto o servicio final al cliente) depende de que cada uno de los miembros de la VE desempeñe su labor de forma apropiada y en los plazos establecidos. Un retraso, un fallo, o incluso la anticipación de un fallo en uno de los nodos, si no se trata a tiempo y de forma adecuada puede hacer peligrar el objetivo común de la VE. Por tanto, es necesario gestionar y supervisar adecuadamente las interdependencias entre los distintos procesos de negocio distribuidos.

4.2.2 Los modelos de selección de socios en la literatura

Cuando se estudian los diferentes proyectos y trabajos vinculados a la identificación y selección de socios en empresas virtuales dinámicas, se observa la coexistencia de dos posiciones encontradas a la hora de afrontar su formación dentro de una plataforma:

- Una solución habitual consiste en seguir un acercamiento de arriba hacia abajo o *top-down* (conocido como enfoque planificador) en el que el planificador diseña la VE dinámica, decide los roles y selecciona los socios que mejor se ajusten a su plan.
- La alternativa opuesta se basa en la utilización de un enfoque de abajo a arriba o *bottom-up* (enfoque competidor). En este caso, el cliente o el *broker* anuncia la oportunidad de colaboración, espera hasta que se formen algunos consorcios espontáneamente (por iniciativa de algunos miembros), y después analiza sus ofertas globales.

Como se puede deducir, el primer enfoque (figura 4.1) se corresponde con la planificación centralizada que hemos comentado en el apartado anterior; un organismo de control central (el planificador/coordinador) se encarga de diseñar/descomponer todo el proceso de negocio global y distribuirlo adecuadamente según un proceso de negociación. Por tanto, se da una evolución desde un nivel de abstracción (*top*) que representa el diseño hasta el nivel de detalle (*down*) que supone seleccionar las entidades que van a llevar a cabo los subprocesos de negocio elementales.

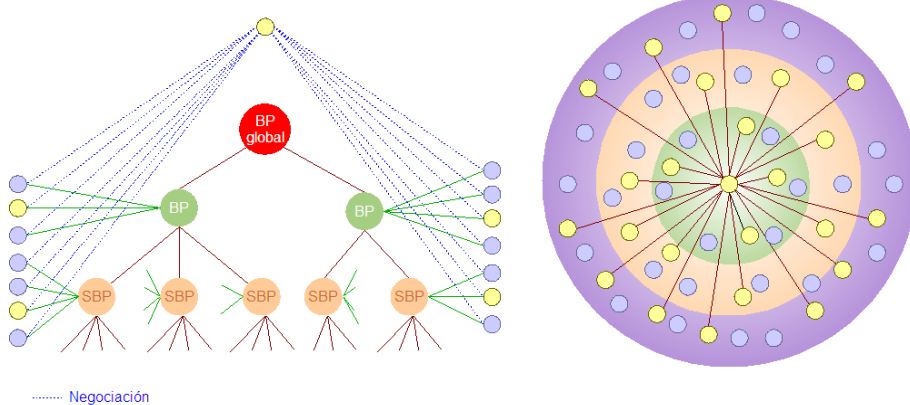


Figura 4.1. Enfoque planificador.

En el segundo enfoque considerado, *bottom-up* (figura 4.2), no existe un organismo de control que diseñe y controle la estructura de la DVE inicial; de hecho, se configuran varios consorcios de manera espontánea que presentan diseños y estructuras muy diferentes entre sí. En este caso, el *broker* no influye en el diseño final de las diferentes estructuras que se constituyen

aunque, evidentemente, deberá seleccionar una de las empresas virtuales creadas a través de un proceso de negociación.

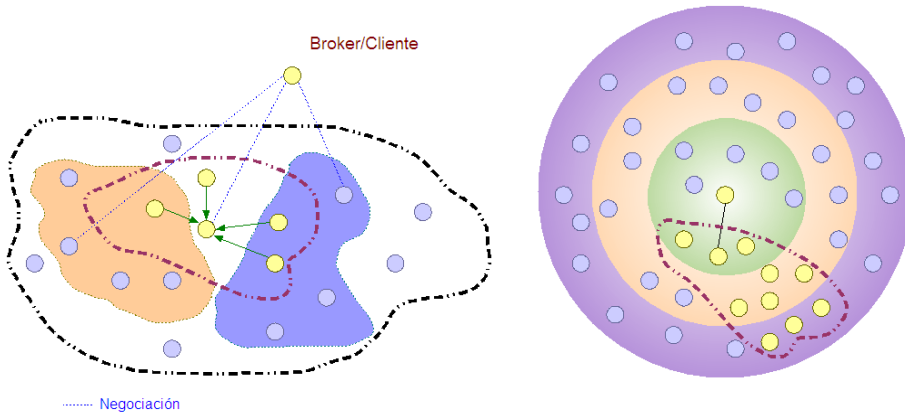


Figura 4.2. Enfoque competidor.

Entre estas dos alternativas extremas pueden definirse soluciones intermedias. Así, por ejemplo, se puede hacer uso de una planificación jerárquica, donde el *broker* o coordinador de la empresa virtual dinámica sólo define los procesos de negocio de más alto nivel (definición abstracta) y selecciona los socios más adecuados para cada uno de ellos. La figura 4.3 permite ejemplificar este enfoque.

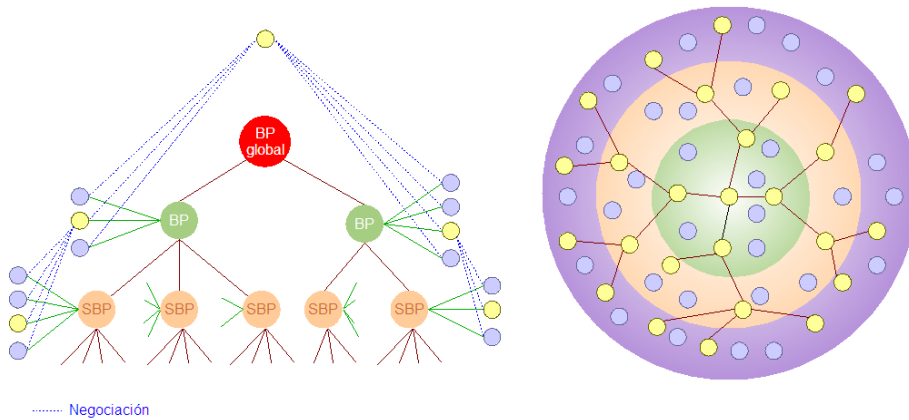


Figura 4.3. Enfoque jerárquico.

Cada uno de estos nuevos miembros de la DVE es responsable de refinar su diseño de acuerdo con sus capacidades locales; estos miembros tienen perfectamente definidos los procesos y subprocesos de negocio, estableciéndoles como locales (si los pueden ejecutar ellos mismos) o remotos (si necesitan de entidades externas para su realización). Como se observa, planificación y detalle se van alternando en el proceso³¹.

Cada enfoque considerado presenta sus ventajas e inconvenientes y exige de la plataforma funcionalidades diferentes. Así, por ejemplo, con el primer enfoque se pierde bastante tiempo en planificar la VE dinámica, aunque al final se obtiene una “DVE centralizada” a medida. También se requieren múltiples agentes (lo lógico es que exista uno por cada subproceso que se precise) para poder negociar con todas y cada una de las entidades que pueden suministrar los diferentes subprocesos requeridos. Es, por tanto, más ineficiente tanto desde el punto de vista temporal como desde el de los recursos.

Sin embargo, puede ser aconsejable en algunas situaciones en las que exista una mayor necesidad de control o solamente se precise la realización de unos pocos subprocesos. Además, el enfoque *top-down* es, sin duda, la opción más considerada en la literatura de investigación (Petersen, 2003; Rocha y Oliveira, 2001; etc.), incluida su variante jerárquica (Ozounis, 2001). Esta circunstancia se debe, posiblemente, a que es una solución más fácil de implementar y que no necesita considerar muchos de los problemas y requerimientos que acarrea el enfoque competidor.

En resumen, el enfoque planificador no garantiza la formación de las mejores DVE posibles. El enfoque *button-up* (Rabelo *et al.*, 2000) puede conducir a formas de DVE más eficientes y ágiles aunque su control suele ser más complejo. Con este enfoque se da la posibilidad a que se configure la mejor VE en su conjunto y sin los sesgos que pueden ocasionar los criterios propios de una única entidad; se da la posibilidad a que se configuren ofertas más realistas dado que se tiene en cuenta en cada momento lo que pueden ofertar aquellas empresas que complementen los procesos y no basa la negociación en simples estimaciones.

31 ¿Y qué sucedería si fuese el administrador quien estableciese la negociación con los posibles candidatos en función de las condiciones específicas de los nuevos miembros?. Obviamente tendríamos una planificación adaptable, en cierto modo intermedia entre la *top-down* y la jerárquica.

4.3 EL MODELO PARA LA SELECCIÓN DE SOCIOS

4.3.1 De los modelos basados en un único enfoque...

Imaginemos por un instante que un cliente, sea éste interno o externo al VBE, acude al entorno de gestación, o bien directamente a uno de sus miembros³², en busca de un producto o servicio (en otras palabras, en busca del proceso de negocio que dará lugar al nacimiento de una nueva VE dinámica) que le permita satisfacer una necesidad u oportunidad de negocio.

- Según el enfoque planificador, ante esta oportunidad de colaboración una entidad del VBE (que puede ser la misma empresa representante) toma el papel de planificador y diseña la VE dinámica, define los procesos y subprocesos que compondrán el proceso de negocio requerido, decide los roles y selecciona los socios que mejor se ajusten a su plan. Por ello, para cada uno de los procesos y subprocesos solicita una lista de todos los socios potenciales que pueden prestar el servicio. Tan pronto como se recibe la lista de socios candidatos para cada subproceso particular, da comienzo el proceso de selección. Dicho proceso se realiza normalmente por medio de negociación dando como resultado (Geppert *et al.*, 1998; Grefen *et al.*, 2000) un acuerdo o un contrato a corto plazo que regula las relaciones de negocio entre las entidades involucradas.
- Si seguimos un enfoque jerárquico el proceso es bastante similar. Ante una oportunidad de colaboración, una entidad del entorno de gestación adquiere el papel de planificador y diseña la VE dinámica, aunque en este caso se queda en un nivel de abstracción alto, es decir, define solamente los procesos de negocio de primer nivel (los que dependen directamente del negocio requerido), y selecciona los socios que mejor se ajusten a su plan. Para ello, y como en el caso anterior, solicita una lista de todos los socios potenciales para cada uno de esos procesos. Tan pronto como tiene a su disposición la lista de socios candidatos para cada subproceso comienza el proceso de selección, que normalmente se realiza por medio de una negociación. A continuación, cada nuevo miembro de la VE dinámica lleva a cabo

32 En este caso, resulta evidente que el miembro del entorno de gestación actuaría como representante de la nueva empresa virtual dinámica que se constituyese a partir de la necesidad.

un ejercicio similar para encontrar y seleccionar aquellas entidades que pueden suministrar los subprocesos de negocio remotos necesarios para llevar a cabo su misión. El proceso continúa hasta que se encuentra un proveedor de servicio para cada uno de los subprocesos de negocio y cada una de las actividades elementales que se precisen.

- Cuando se considera un enfoque competidor (*button-up*) el proceso cambia un poco. Tras aparecer la oportunidad de colaboración, el *broker* se encarga de anunciarla/publicarla en el entorno de gestación. Los agentes, entonces, intercambian información entre sí hasta que se constituyen, de manera espontánea, varias VE dinámicas capaces de responder a la necesidad que ha surgido. Al *broker* tan sólo le resta seleccionar, de entre todas esas DVE, la que mejor se ajusta a su plan de negocio en base a sus propios criterios y a un proceso de negociación.

4.3.2 ... a nuestro modelo multienfoque

Si pretendemos ser lo más realistas posibles, nuestro modelo ha de lograr que la formación de VE dinámicas pueda efectuarse siguiendo no sólo cada uno de los distintos enfoques analizados, sino también las múltiples y variadas combinaciones que puedan surgir a partir de ellos. Queremos crear un modelo que sea lo más genérico y abierto posible, con una orientación multienfoque que permita aunar las ventajas de las soluciones planteadas hasta el momento y mitigar sus carencias, aportando nuevas posibilidades que hagan de la formación de las empresas virtuales dinámicas un proceso más real y eficiente.

Para alcanzar tal propósito, cualquier aplicación software que se construya a partir de nuestro modelo deberá incorporar entre sus funcionalidades la posibilidad de que cada agente involucrado (perteneciente o no al entorno de gestación) pueda seleccionar a los proveedores de servicio de dos formas distintas. Como se puede apreciar en la figura 4.4, la primera de esas alternativas consiste en que los agentes puedan publicar sus necesidades en el VBE y responder, posteriormente, a las ofertas que vayan surgiendo; en otras palabras, los agentes podrán seleccionar aquella DVE o subDVE que más les interese. La segunda alternativa permite a los agentes solicitar al VBE la lista de los proveedores registrados para cada uno de los servicios que demanden, y elegir, de entre ellos, el o los proveedores que más les convengan.

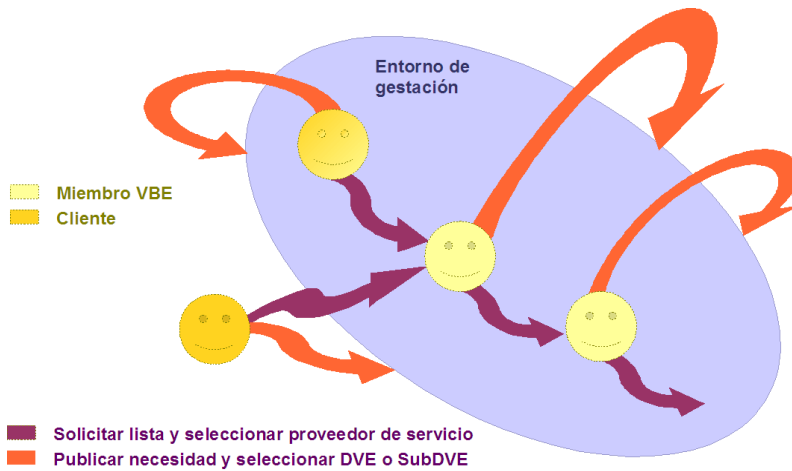


Figura 4.4. Capacidades de los agentes en la formación de VE dinámicas considerando nuestro modelo multienfoque.

Proporcionando esta capacidad de selección a los agentes estamos definiendo un modelo (figura 4.5) mucho más general que los presentados hasta la fecha y, al mismo tiempo, abrimos las puertas a un amplio abanico de nuevas posibilidades para poder formar VE dinámicas.

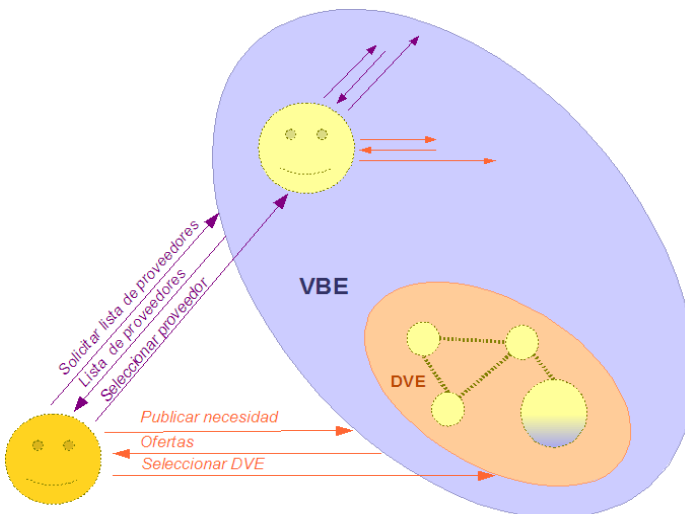


Figura 4.5. Modelo descriptivo del proceso de formación.

Cuando una entidad necesita la provisión de un servicio, puede solicitar al VBE (en concreto a su administrador) la lista de aquellos proveedores que lo suministren, y seleccionarlos mediante una negociación que tenga presente los múltiples criterios (objetivos o subjetivos) que se hayan podido establecer previamente. Pero también puede publicar su necesidad y esperar a que se formen espontáneamente empresas virtuales dinámicas que presten el servicio de forma conjunta y global. En ambos casos, la formación de las VE dinámicas se produce de forma transparente con respecto a la entidad que solicita el servicio.

Permitiendo esta simple elección nuestro modelo hace posible la formación de DVE que se ajusten a los distintos modelos y estructuras de planificación existentes en la literatura y, además, a cualquier otra combinación posible. El modelo presentado permite crear VE dinámicas que se ajusten a la definición pura de DVE, VE dinámicas controladas por un único miembro, DVE federadas, VE en las que se establezcan diferentes tipos de control en todos y cada uno de los procesos de negocio, o en unos pocos, etc. En resumidas cuentas, posibilita la creación de cualquier tipo de VE dinámica imaginable, y con el nivel de control que se precise en cada momento.

La figura 4.6 muestra un posible ejemplo de DVE construida a partir de nuestro modelo. Como se puede apreciar, la VE dinámica creada engloba los distintos enfoques existentes, lo que da lugar a la selección tanto de simples entidades como de subDVE.

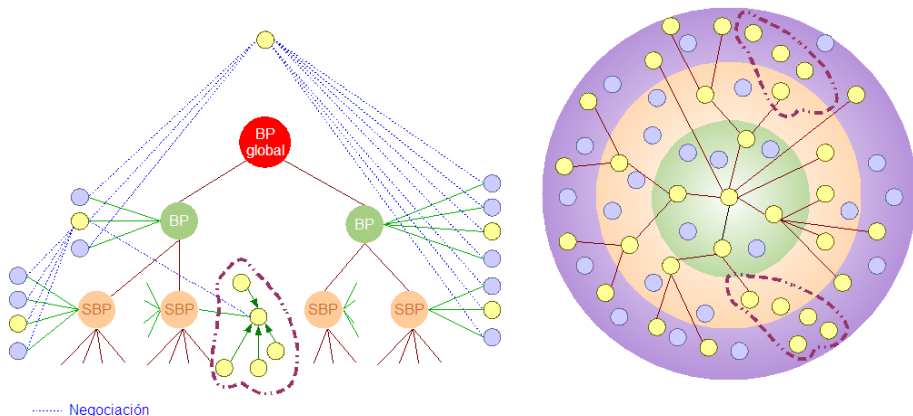


Figura 4.6. Posible ejemplo de VE dinámica constituida a partir de un modelo multienfoque.

En definitiva, hemos definido un modelo del proceso de selección innovador, multienfoque y global, que amplía las posibilidades de los modelos existentes en la literatura, pero que no deja de ser un modelo notablemente sencillo.

Una vez llegados a este punto, el siguiente paso a dar consiste en presentar y justificar las tecnologías que, en nuestra opinión, son las más adecuadas para implementar sus características fundamentales en una aplicación informática.

4.4 LOS SISTEMAS MULTIAGENTE

La última década del siglo XX fue testigo del nacimiento de un nuevo paradigma para el desarrollo de aplicaciones software, los sistemas multiagente (MAS, *Multi Agent Systems*). Este advenimiento supuso un cambio radical en la forma de abordar los problemas en multitud de áreas de investigación y aplicación, permitiendo la mejora significativa de muchas de las soluciones halladas hasta el momento y posibilitando la resolución de multitud de problemas que no habían podido afrontarse de manera satisfactoria, bien por la falta de una tecnología adecuada o bien porque la utilización de la existente suponía un gran esfuerzo.

Los MAS se concibieron sobre la idea de que las actividades complejas pueden dividirse en actividades más pequeñas y que éstas, a su vez, pueden volver a descomponerse en otras más simples, y así sucesivamente hasta que se encuentra un conjunto de actividades primitivas o elementales. Cada actividad elemental en este modelo puede ser realizada por una entidad de propósito especial llamada agente software. A su vez, diferentes agentes pueden interactuar entre sí y con el entorno para solucionar un problema complejo y/o específico o para proporcionar un servicio de una manera distribuida y coordinada constituyendo, así, un sistema multiagente.

Esta tecnología posee algunas características excepcionales que han propiciado su utilización en multitud de aplicaciones que van desde sistemas comparativamente pequeños, como pueden ser los filtros empleados en el correo electrónico, hasta sistemas complejos y con un impacto crítico en el funcionamiento, como son los sistemas de control del tráfico aéreo, pasando por el control de procesos de producción, gestión de la información, comercio electrónico, aplicaciones en Internet, juegos, aplicaciones médicas, etc. En particular, uno de los campos en los que la tecnología basada en sistemas multiagente está adquiriendo una relevancia notable es el de la VE.

4.4.1 El concepto de agente software

Los orígenes de la tecnología de agentes se enmarcan dentro del campo de la Inteligencia Artificial Distribuida (DAI, *Distributed Artificial Intelligence*), dedicada al estudio de las técnicas necesarias para la coordinación y distribución del conocimiento (Iglesias, 1998). Con el paso del tiempo, la proliferación de los ordenadores personales y el auge de las autopistas de información, la idea de “agente” se ha convertido en uno de los temas de actualidad (Corchado y Molina, 2002): hoy en día se habla de agentes inteligentes, agentes móviles, sistemas multiagente, etc.

Sin embargo, y aunque ya no se discuten las ventajas de los agentes software inteligentes, existen discrepancias para definirlos y delimitarlos. La diversidad de opiniones presentes en la comunidad científica sobre este tema tiene su origen, irónicamente, en sus virtudes: el campo de los agentes ha atraído a científicos procedentes de áreas muy dispares (psicología, sociología, ingeniería del software, inteligencia artificial, etc.) y cada uno de los miembros de estas comunidades tiende a ver el problema desde su propia perspectiva (Foner, 1993).

Ante la ausencia de una definición consensuada, numerosos autores han optado por delimitar el concepto de agente mediante un conjunto de propiedades o atributos, dando lugar a lo que se conoce como la *noción débil* de agente. La siguiente lista no pretende ser exhaustiva, pero recopila algunas de las propiedades que comúnmente se asocian a este concepto (Wooldridge y Jennings, 1995; Wooldridge, 1997; Iglesias, 1998; Franklin y Graesser, 1997; Genesereth *et al.*, 1994; Julián y Botti, 2000):

- *Autonomía*: los agentes pueden actuar sin la intervención de un ser humano o de otros agentes, y tienen algún tipo de control sobre sus acciones y su estado interno.
- *Habilidad social o sociabilidad*: los agentes interaccionan con otros agentes, humanos o no, para resolver sus propios problemas y ayudar a otros con los suyos. Esta interacción se realiza a través de algún tipo de lenguaje de comunicación.
- *Reactividad, sensibilidad*: los agentes son capaces de percibir estímulos de su entorno (bien del mundo real, de otros agentes, de Internet, etc.), y reaccionar consecuentemente a los cambios que se producen en él.

- *Iniciativa, proactividad*: los agentes también son capaces de elegir en cada momento las acciones a llevar a cabo para alcanzar sus objetivos; es decir, pueden “tomar la iniciativa” siempre y cuando sea apropiado y necesario.
- *Racionalidad*: asume que un agente actúa de forma racional, siendo capaz de emprender aquellas acciones que le permitan alcanzar sus objetivos de la forma más eficiente en función de los recursos disponibles.
- *Aprendizaje*: para que un agente pueda ser considerado inteligente ha de poder aprender del entorno que le rodea.
- *Movilidad*: capacidad de un agente de trasladarse a través de los nodos de una red para llevar a cabo sus tareas.
- *Veracidad*: un agente no comunica información falsa de forma deliberada.
- *Benevolencia*: un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos.

De entre todas ellas, autonomía, sociabilidad, reactividad e iniciativa parecen ser las propiedades fundamentales para lograr que un agente actúe de forma flexible en la consecución de los objetivos diseñados. En opinión de Jennings *et al.* (Jennings *et al.*, 1998) es la presencia de estos atributos en una única entidad lo que permite distinguir a los agentes de otros paradigmas software relacionados como son los sistemas orientados a objetos, los sistemas distribuidos o los sistemas expertos³³. Naturalmente, un agente puede poseer características adicionales, y para ciertos tipos de aplicaciones algunos atributos pueden ser más importantes que otros.

En base a estas consideraciones, podemos arriesgarnos a enunciar una definición operativa de agente³⁴: un agente es, simplemente, “un sistema informático (soportado por software y/o hardware) que actúa de forma flexible, dentro de un entorno, con el propósito de cumplir sus objetivos. Este sistema se caracteriza principalmente por ser autónomo, reactivo, proactivo y por estar dotado de habilidad social, lo que se traduce en una interacción con otros agentes y con su entorno gracias al intercambio de mensajes que hacen uso de un lenguaje de comunicación de agentes”.

33 En (Wooldridge, 1997) se realiza una comparativa de todos estos sistemas con aquellos basados en agentes.

34 En (Franklin y Graesser, 1997; Ingham, 1999) pueden encontrarse un amplio conjunto de definiciones alternativas.

Existe, además, una definición más restrictiva, la *noción fuerte*. Desde esta perspectiva el agente se contempla como un sistema que, además de poseer las propiedades ya enumeradas, se define o implementa usando conceptos que habitualmente aplicamos a los seres humanos. Esta postura, enunciada por Shoham (Shoham, 1993), considera a los agentes como sistemas intencionales, es decir, entidades cuyo comportamiento se puede describir en base a sus actitudes de información (conocimientos y creencias) o a través de sus actitudes de decisión (deseos, objetivos, intereses, compromisos, obligaciones, etc.)³⁵.

Una vez esbozada la definición de agente, podemos extenderla de manera natural a *sistema basado en agentes*, que será aquel sistema que utiliza agentes para llevar a cabo su función. Si estos agentes colaboran e interactúan entre sí para resolver un problema, estaremos ante un *sistema multiagente*.

4.4.2 Los sistemas multiagente

La característica de sociabilidad de los agentes software hace de ellos un concepto muy interesante para desarrollar sistemas informáticos distribuidos donde varios agentes interactúan entre sí y con el entorno. Cada uno de estos agentes tiene la capacidad de resolver una parte específica del problema, de forma que juntos pueden alcanzar la funcionalidad deseada operando de forma asíncrona y sin un sistema de control global. A estos sistemas se les denomina sistemas multiagente.

Al igual que ocurre para la noción de agente, tampoco se encuentra en la literatura una definición globalmente aceptada para el concepto de sistema multiagente. Para Jennings y Wooldridge (Jennings y Wooldridge, 1998) no es más que “un conjunto de agentes interrelacionados de forma que cada agente tiene capacidad para solucionar parcialmente el problema, no hay un sistema global de control, los datos no están centralizados y operan de forma asíncrona”. Otra definición interesante es la facilitada por Ferber (Ferber, 1999) que ve un MAS como un sistema que reúne los siguientes elementos:

35 Para representar estas intenciones, se han empleado diversos formalismos lógicos, de entre los que cabe destacar la teoría de la intención de Cohen y Levesque (Cohen y Levesque, 1990) y la lógica multi-modal BDI (Belief, Desire, Intention) (Haddadi y Sundermeyer, 1996; Rao y Georgeff, 1995).

- Un *entorno*.
- Un *conjunto de objetos*. Estos objetos se encuentran integrados con el entorno de forma que es posible, en un momento dado, asociar uno de estos objetos con un lugar en el entorno. Estos objetos son pasivos, pueden ser percibidos, creados, destruidos y modificados por los agentes.
- Un *conjunto de agentes*. Los agentes son objetos especiales que representan las entidades activas del sistema.
- Un *conjunto de relaciones* que unen objetos y, por tanto, a los distintos agentes.
- Un *conjunto de operaciones* que hacen posible que los agentes perciban, produzcan, consuman, transformen y manipulen objetos.
- *Operadores* que representan la aplicación de operaciones sobre el mundo y la reacción de éste al ser alterado. Pueden ser vistas como las *leyes del universo*.

Esta definición muestra que un agente influye en el resto no sólo a través de la comunicación explícita sino también como consecuencia de su actuación sobre el entorno. Este hecho aumenta enormemente la complejidad del desarrollo de los sistemas multiagente (GRASIA web, 2007), ya que obliga a estudiar el entorno con detalle para detectar qué acciones realizadas por un agente pueden afectar a otro agente.

Algunos investigadores emplean el término “sistema basado en agentes” como equivalente al de “sistema multiagente” (Jennings *et al.*, 1998) aunque, en realidad, presentan diferencias significativas. Por ello, conviene dedicar unas palabras a aclarar la diferencia entre ambos conceptos. Así, un sistema basado en agentes es aquel que utiliza el concepto de agente como mecanismo de abstracción, pero aunque esté modelado en términos de agentes (Wooldridge y Ciancarini, 2001) podría construirse sin las estructuras de software propias de este paradigma, y sin hacer uso de características tan interesantes como constituye, por ejemplo, la interacción entre agentes.

Por contra, un sistema multiagente es aquel que se diseña e implementa pensando en que estará compuesto por varios agentes que interactuarán entre sí, de forma que juntos permitan alcanzar la funcionalidad deseada (Bussman y Müller, 1993), funcionalidad que está fuera de las posibilidades de cada agente individual (Flores-Mendez, 2000). En este caso, hay que hacer un mayor esfuerzo de abstracción, identificar mecanismos de aprendizaje, coordinación, negociación, etc.

4.4.3 Interacción entre los agentes

La interacción entre los propios agentes es uno de los elementos clave en cualquier sistema multiagente dado que les proporciona la posibilidad de coordinarse, negociar y cooperar entre sí. Se pueden distinguir dos formas básicas de interacción: los sistemas de pizarra y el de paso de mensajes. La primera de estas formas se caracteriza por la existencia de una o varias zonas de trabajo común (pizarras) donde los agentes comparten todo tipo de información sin que se de una comunicación directa entre ellos.

La otra posibilidad es a través del mecanismo de *paso de mensajes*. Con este sistema, un agente emisor envía un mensaje solicitando un determinado servicio a uno o más agentes receptores, que a su vez podrán responder comunicando lo que consideren oportuno. A diferencia de la arquitectura de pizarra, los agentes deben mantener conocimiento sobre su entorno para saber a qué agentes deben dirigir sus mensajes. Debido a su flexibilidad, numerosos sistemas multiagente han adoptado esta forma de comunicación.

Para que el intercambio de información entre agentes se produzca de forma eficiente es necesario que los mensajes sean interpretables por los agentes que participan en la comunicación. Para lograr este objetivo es conveniente establecer un protocolo de comunicación, una ontología compartida y un formato del contenido de la comunicación común definido mediante un lenguaje de comunicación entre agentes (Bellifemine *et al.*, 2005).

Los protocolos de comunicación definen la estructura de la conversación (secuencia de mensajes) entre dos o más agentes, que dependerá de la intención con la que se inicie. Con ello, los agentes se pueden comunicar en la forma adecuada según el objetivo de la conversación. Así, se pueden establecer diferentes protocolos de conversación (Cost *et al.*, 2001): protocolos para solicitar información, para solicitar la ejecución de una acción, contratar un servicio, etc.

Aunque existen muchos protocolos de comunicación, uno de los más comentados en la literatura es el Protocolo de Red de Contratos (*Contract Net Protocol*) (Smith, 1980) que establece el proceso de negociación que lleva a cabo aquel agente que pretende acordar con otro la realización de una tarea. Por otro lado, los estándares actuales como FIPA (FIPA web, 2007) establecen diversos protocolos que se pueden usar para diseñar la interacción entre agentes.

Las ontologías se definen como esquemas que describen conceptos y las relaciones que hay entre ellos (Finin *et al.*, 1997). Es importante que las ontologías con las que los agentes conceptúan un determinado dominio sean similares, ya que de esta manera se garantiza que los agentes implicados en la comunicación realicen la misma interpretación del contenido del mensaje. Una vez que los agentes que interactúan se han puesto de acuerdo en una ontología común, se empleará para interpretar las comunicaciones.

Los mensajes que se envían los agentes no sólo deben portar el contenido que se desea comunicar. Para asegurar una comunicación eficiente el mensaje debe incluir otros datos como el nombre del agente emisor, el del receptor, el protocolo de comunicación empleado, la ontología con la que se ha especificado los contenidos, etc. Además, para que los agentes puedan interpretar el mensaje su estructura debe ajustarse a un formato. Estos formatos, a su vez, están definidos por los lenguajes de comunicación entre agentes (Molina *et al.*, 2004). Uno de los lenguajes más referenciados es KQML (Labrou y Finin, 1998), del que se ha inspirado el ACL de FIPA.

Además de especificar el formato de los mensajes, también debe definirse el lenguaje en el que se comunican los contenidos; en otras palabras, es necesario precisar cómo se pueden formar expresiones haciendo uso de los conceptos establecidos en la ontología. A este lenguaje se le conoce con el nombre de lenguaje de contenidos. Algunos ejemplos de lenguajes de contenido son XML, FIPA-SL y KIF (*Knowledge Interchange Format*).

4.4.4 Utilidad de los sistemas multiagente

Al revisar la literatura relacionada con los agentes, se puede observar un cambio paulatino en su motivación. Si en un principio se orientaba más hacia la definición de los aspectos básicos de la tecnología, en la actualidad, y gracias a la existencia de estándares, metodologías y plataformas de desarrollo adecuados, los investigadores se centran más en la resolución de problemas que en las cuestiones técnicas. Esta situación hace pensar que los MAS han alcanzado la madurez suficiente como para aplicarse con éxito a problemas que tenían difícil solución con las tecnologías anteriores.

Pero cualquier tecnología que pretenda tener éxito en su aplicación debe ser, ante todo, útil. Para Jennings y Wooldridge (Jennings y Wooldridge, 1998) una tecnología informática será útil siempre y cuando resuelva alguno de los

dos tipos de problemas siguientes: aquellos que hasta su aparición no se habían resuelto bien por su dificultad o porque conllevaban un coste excesivo, o aquellos problemas que hasta ese instante se habían resuelto con cierta dificultad y para los que la nueva tecnología aporta soluciones más rápidas, menos costosas y de mayor calidad. La cuestión fundamental es si los agentes software son capaces de resolver alguno de esos dos tipos de problemas.

Con respecto a la primera categoría, Jennings y Wooldridge proponen tres tipos de sistemas para los cuales la tecnología multiagente se presenta como una alternativa viable: los sistemas abiertos (*Open Systems*), los complejos (*Complex System*) y los sistemas obicuos (*Ubiquitous Computing Systems*). Cada uno de estos sistemas presentan características y necesidades propias.

- Los *sistemas abiertos* son sistemas en los que sus componentes pueden cambiar a lo largo del tiempo de forma aleatoria e inesperada. Un ejemplo característico de este tipo de sistemas es Internet, donde la información es heterogénea y está distribuida en diferentes nodos diseñados y desarrollados por diferentes personas. Un sistema informático que opere en estos entornos debe ser capaz de coordinarse con todos los elementos que lo constituyen.
- Los *sistemas complejos*. Conceptualmente son aquellos sistemas en cuyo comportamiento temporal se producen bifurcaciones, de modo que la posibilidad de predecir su evolución es prácticamente nula. De forma general, un sistema se reconoce tanto más complejo cuanto mayor es el número de componentes y, por consiguiente, mayor es el número de interacciones entre estos (Delgado, 2002). Una forma lógica de tratar estos problemas es mediante la descomposición del sistema en subsistemas más simples que se puedan tratar más fácilmente, de forma que el objetivo final pueda conseguirse a través de la interacción entre todos los módulos.
- A diferencia de los sistemas informáticos clásicos donde un componente proporciona servicios a uno o más usuarios, en los *sistemas ubicuos* es un número potencialmente grande de componentes pequeños los que proporcionan estos servicios. A pesar de los avances realizados en las últimas dos décadas en las interfaces de usuario, la utilización de muchos programas sigue siendo problemática, ya que es el usuario quien debe guiar, paso a paso, las tareas que han de realizarse para resolver un determinado problema. Por tanto, lo ideal sería que el computador fuese capaz de cooperar con el usuario en la resolución de los problemas planteados.

Además de dar solución a problemas no resueltos, la tecnología basada en agentes también proporciona una forma de mejorar el diseño e implementación del software ya existente. Existen problemas para los que la utilización de agentes inteligentes suele proporcionar una forma natural de modelar el sistema. Dichos problemas se caracterizan, fundamentalmente, porque a) la información, el control y/o los recursos se encuentran distribuidos, b) el sistema se organiza de forma natural como una sociedad de componentes autónomos y cooperativos, y c) porque el sistema posee elementos que deben interactuar con componentes de otros sistemas.

4.4.5 ¿Por qué son adecuados los MAS en las DVE?

Existe muchas características en el dominio de las VE dinámicas que hacen de estos modelos organizacionales un área de aplicación adecuado para los MAS. Algunas de ellas son (Ambroszkiewicz *et al.*, 1998; Camarinha-Matos y Afsarmanesh, 2001; Petersen *et al.*, 2002; Dignum y Dignum, 2002; Sanz, 2004; Afsarmanesh, 2005b):

- Los agentes software autónomos, adaptativos y cooperativos son adecuados para aquellos dominios que requieren una constante adaptación a entornos distribuidos y/o a demandas cambiantes, como es el caso de la DVE.
- Una VE dinámica está formada por componentes autónomos, heterogéneos y distribuidos, una situación que se puede trasladar fácilmente a un sistema multiagente.
- Los MAS abordan, entre otras cuestiones, la coordinación y resolución de problemas distribuidos, aspectos críticos en la gestión de las empresas virtuales dinámicas.
- La toma de decisiones con información incompleta y la implicación de los miembros de la red como entidades autónomas, que aunque tienen la voluntad de cooperar para alcanzar una meta común podrían llegar a ser competidores si se consideran otras metas de negocio, es otro punto en común.
- La ejecución y supervisión eficaz de los procesos de negocio distribuidos requiere de rápidas reacciones por parte de los miembros de la VE dinámica. Existe, por tanto, la necesidad de que cada compañía tenga un representante (un observador) en la red. Al tener agentes que representan a cada organización, es posible asignar recursos (agentes) a supervisar la red.

- Los recientes progresos en el área de las VE hacen que el modelado e intercambio de información pierda importancia frente al modelado de los roles, considerando aspectos relacionados con la distribución de las responsabilidades, capacidades y conocimientos.
- La fase de formación de la VE dinámica, en la que es necesaria seleccionar socios y distribuir tareas, muestra características de mercado y necesidades de negociación que han sido cuestiones ampliamente estudiadas en los sistemas multiagente.
- Una empresa virtual ágil es una organización dinámica que puede requerir, en un momento dado, una reconfiguración (reemplazo de algún socio, replanificación de la colaboración, cambios en los roles, etcétera.). En esta situación se necesita un paradigma de modelado flexible y que pueda adaptarse rápidamente a las circunstancias.
- Las funcionalidades que soportan las VE necesitan interactuar con el entorno local (aplicaciones y personas). Precisamente, la interacción con el entorno es uno de los atributos que definen a los agentes.
- Se requiere más flexibilidad que la suministrada por un modelo cliente-servidor para soportar el cambio dinámico de roles de los miembros de la DVE. En un sistema multiagente se puede lograr fácilmente este dinamismo mediante, por ejemplo, la utilización de comportamientos (*behaviors*).
- La propiedad de escalabilidad de los MAS parece particularmente adecuada para soportar VE dinámicas en la que se pueden establecer diferentes niveles de cooperación con distintos conjuntos de socios. Por otro lado, cada empresa puede verse a sí misma como constituida por una red de entidades (departamentos) semi-autónomas.
- La evolución continua de los modelos de negocio, tecnologías, paradigmas organizacionales y las condiciones de mercado requieren un soporte efectivo para la evolución y un alto nivel de modularidad de las infraestructuras.
- En el contexto de las VE dinámicas están emergiendo nuevas formas de *teamwork*, concretamente la ingeniería concurrente cooperativa. Los agentes pueden jugar un papel muy importante como asistentes de los actores humanos en dichos entornos.
- Existe la necesidad de gestionar los requisitos de autonomía frente al comportamiento cooperativo: los enfoques de MAS federados pueden proporcionar una solución equilibrada a esta necesidad.
- Es necesario una terminología que sea precisa, nada ambigua y que pueda ser comprendida de forma colectiva. Estos requisitos pueden ser satisfechos usando ontologías y lenguajes comunes.

- Finalmente, como los agentes suelen ser diseñados y desarrollados de forma independiente, es bastante difícil garantizar la coordinación a menos que se adopten reglas comunes (“leyes sociales” o estándares). En este sentido, pueden combinarse los fundamentos teóricos sobre sociabilidad de los agentes con los desarrollos actuales en el marco de las VE dinámicas.

Sin embargo, los sistemas multiagente también presentan varias limitaciones que deben ser subsanadas. Algunas de ellas son las que se muestran a continuación (Camarinha-Matos, 2003):

- Aunque existen muchas plataformas de desarrollo para los agentes, la mayoría conforme a los estándares FIPA (como JADE o ZEUS), no son lo suficientemente robustas cuando operan sobre Internet.
- Los mecanismos de seguridad y persistencia no están bien integrados con los sistemas multiagente.
- Existe una falta de integración entre las comunidades de inteligencia artificial y la de procesos de negocio.
- Hace falta integrar los lenguajes de contenido (como ACL) y los lenguajes de definición de procesos de negocio.
- La mayoría de los desarrollos son a nivel de prototipo y existen muy pocos casos de demostración reales.

4.5 LOS SISTEMAS EXPERTOS

Antes de la aparición del ordenador, el hombre ya se preguntaba si alguna vez se le arrebataría el privilegio de razonar y pensar. En la actualidad, existe un campo dentro de la inteligencia artificial al que se le atribuye esa facultad: el de los sistemas expertos³⁶ (ES, *Expert System*). Pero, ¿qué es un ES?, ¿qué elementos componen este tipo de sistemas?, ¿cuándo se emplean?. En los próximos apartados intentaremos dar respuesta a estas y otras cuestiones relacionadas, de forma que podamos comprender mejor la decisión de incorporar un ES al problema particular que nos atañe integrándolo, de este modo, con el sistema multiagente.

36 Los sistemas expertos son un subconjunto especial dentro de los sistemas basados en el conocimiento (KBS, *Knowledge-Based System*), aunque en ocasiones ambos conceptos son considerados como sinónimos.

4.5.1 ¿Qué es un sistema experto?

Al igual que ocurre con otros paradigmas y conceptos tratados en esta memoria, no resulta nada sencillo dar una respuesta clara e intuitiva a esta cuestión. Se trata de un concepto en continua evolución, fruto del rápido desarrollo de la tecnología (Durking, 1994) y de la paulatina ampliación de sus funciones, lo que ha dado lugar a que en la literatura existente convivan multitud de definiciones que incluso pueden llegar a parecer bastante dispares entre sí.

Entre ellas podríamos destacar la proporcionada por Edward Feigenbaum, también conocido como el “padre” de los sistemas expertos, que los definió como “un programa de computador inteligente que usa el conocimiento y los procedimientos de inferencia para resolver problemas que son lo suficientemente difíciles como para requerir la intervención de un experto humano para su resolución” (Feigenbaum y Feldman, 1963), o la definición de Stevens (Stevens, 1984) que considera a los sistemas expertos como “máquinas que piensan y razonan como un experto lo haría en una cierta especialidad o campo”.

Estas y otras definiciones comparten una misma esencia: consideran que un ES es un sistema informático (hardware y software) capaz de emular las prestaciones de los expertos humanos en un área concreta de conocimiento especializado (el llamado dominio de conocimiento). En otras palabras, “son sistemas informáticos que simulan el proceso de aprendizaje, memorización, razonamiento, comunicación, y, en consecuencia, de acción de un experto humano en cualquier rama de la ciencia” (III web, 2007), lo que les permiten almacenar datos y conocimiento, sacar conclusiones lógicas, tomar decisiones, comunicarse con expertos humanos, explicar el porqué de las decisiones tomadas y realizar acciones.

El especialista humano es aquella persona que tiene experiencia desarrollada en cierta área, lo que le permite tener conocimientos o habilidades especiales que la mayoría no conoce o de las que no dispone, y que puede resolver problemas que la mayoría no podría resolver o los resuelve con mucha mayor eficiencia (y a un menor coste). En su dominio de conocimiento el sistema experto razona o hace inferencias de la misma forma que un especialista humano inferiría la solución de un problema (dados unos hechos se infiere una conclusión).

Los problemas que los sistemas expertos pueden tratar son muy diversos, como lo demuestra el rápido desarrollo en las últimas décadas de un gran número de aplicaciones de estos sistemas en prácticamente todas las ramas del conocimiento. Así, podemos encontrar ejemplos de aplicación de los sistemas expertos en ámbitos que van desde la medicina hasta la ingeniería, pasando por la economía, psicología, finanzas, derecho, educación, informática, telecomunicaciones, ciencias aplicadas, transporte, aeronáutica, investigación militar, geología, agricultura, etc.

Si particularizamos aún más, podremos apreciar que se ha hecho uso de los sistemas expertos en la meteorología, en las transacciones bancarias, en el control de tráfico en una ciudad, en el diagnóstico médico de distintas enfermedades, para el enfoque automático de imágenes fotográficas, en el diagnóstico de problemas en automóviles, en las interfaces de ordenador en lenguaje hablado, en la gestión distribuida de redes de ordenador, para el urbanismo y la gestión del territorio, en la navegación terrestre y marítima, en la gestión de recursos energéticos, etc.

4.5.2 Estructura de un sistema experto

Los elementos principales de un ES son (Castillo *et al.*, 1998):

- *La componente humana.* Un sistema experto es generalmente el resultado de la colaboración de uno o varios *expertos humanos* especialistas en el tema de estudio y los *ingenieros del conocimiento*, con los *usuarios* siempre en mente. Los expertos humanos suministran el conocimiento básico en el tema de interés, de forma ordenada y estructurada, y un conjunto de relaciones bien definidas y explicadas, y los ingenieros trasladan este conocimiento a un lenguaje que el sistema experto pueda entender.
- *Base de conocimientos.* Es la parte del sistema experto que contiene el conocimiento sobre el dominio. Primero hay que obtener el conocimiento del experto y posteriormente codificarlo en la base de conocimientos. Una forma clásica de representar el conocimiento en un sistema experto son las reglas, pero también se emplean otros mecanismos como, por ejemplo, las distribuciones de probabilidad.
- *Base de hechos o memoria de trabajo.* Contiene los hechos sobre un problema que se han descubierto durante su análisis. El usuario introduce la información del problema actual en la base de hechos, y

el sistema empareja esta información con el conocimiento disponible en la base de conocimientos para deducir nuevos hechos. Esta base es temporal (contiene información que no se conserva) y depende exclusivamente de la situación estudiada.

- *Motor de inferencia.* Es el corazón de todo sistema experto, ya que se encarga de modelar el proceso de razonamiento humano. Contrasta los datos particulares almacenados en la memoria de trabajo con el conocimiento contenido en la base de conocimientos para así obtener conclusiones acerca del problema.

Además de estos componentes básicos, en ocasiones pueden aparecer otros módulos como los que se describen a continuación:

- *Subsistema de adquisición del conocimiento.* Controla el flujo de nuevos datos que fluye del experto humano a la base de conocimiento, valorando si se precisa más conocimiento o si el recibido realmente ya existe en el subsistema. Cuando lo estima oportuno, incorpora la nueva información a la base correspondiente.
- *Subsistema de adquisición de la información.* Si la información inicial sobre el problema es muy limitada y no se pueden sacar conclusiones, el motor de inferencia utiliza este subsistema para obtener información con la que continuar el proceso de inferencia. En algunos casos, el usuario puede suministrar la información requerida para éste y otros propósitos; de aquí se deriva tanto la necesidad de una interfaz de usuario como la obligación de comprobar la consistencia de la información suministrada antes de introducirla en la memoria de trabajo.
- *Interfaz de usuario.* La interfaz de usuario es el enlace entre el sistema experto y el usuario: permite que el usuario pueda describir el problema al ES, interpreta sus preguntas y la información ofrecida; formula la información generada por el sistema incluyendo respuestas a las preguntas, explicaciones y justificaciones, etc., posibilita, en definitiva, que la respuesta suministrada por el sistema sea inteligible para el interesado. En consecuencia, una implementación inadecuada de la interfaz minaría notablemente la calidad del sistema experto. Por ello, para que sea una herramienta efectiva, debe incorporar mecanismos eficientes para mostrar y obtener información de forma fácil y agradable, las preguntas y explicaciones deben ser comprensibles, el aprendizaje de su manejo debe ser rápido y debe evitarse la entrada de datos errónea.

- *Control de la coherencia.* El subsistema de control de la coherencia ha aparecido en los sistemas expertos muy recientemente. Sin embargo, es un componente esencial. Este subsistema controla la consistencia de la base de conocimientos y evita que unidades de conocimiento inconsistentes entren en ella. En situaciones complejas incluso un experto humano puede formular afirmaciones inconsistentes. Por ello, sin un subsistema de control de la coherencia, unidades de conocimiento contradictorio pueden formar parte de la base de conocimiento, dando lugar a un comportamiento insatisfactorio del sistema. Por otra parte, cuando se solicita información de los expertos humanos, este subsistema informa sobre las restricciones que ésta debe cumplir para que sea coherente con la ya presente en la base de conocimiento; de esta forma, ayuda a los expertos humanos a dar información fiable.
- *El subsistema de ejecución de órdenes.* Es el componente que permite al sistema experto iniciar acciones en base a las conclusiones que deduce el motor de inferencia. Como ejemplos, un sistema experto diseñado para analizar el tráfico ferroviario puede decidir retrasar o incluso detener ciertos trenes para optimizar el tráfico global, o un sistema para el control de una central nuclear puede abrir o cerrar ciertas válvulas, mover barras, etc., para evitar un accidente. La explicación de las razones por las que se inician estas acciones pueden darse al usuario mediante el subsistema de explicación.
- *Subsistema de explicación.* Una característica de los sistemas expertos es su habilidad para explicar su razonamiento. Usando este módulo, un sistema experto puede proporcionar una explicación del proceso que sigue el motor de inferencia. Este módulo proporciona beneficios tanto al diseñador del sistema como a los usuarios: el diseñador puede usarlo para detectar errores y depurar el sistema experto durante su desarrollo, mientras que el usuario se beneficia de la transparencia del sistema lo que le permite tomar decisiones importantes amparándose en el consejo del sistema experto y, además, convertirse en especialista en la materia al asimilar el proceso de razonamiento seguido por el sistema.
- *El subsistema de aprendizaje.* Una característica esencial de un ES es su capacidad para aprender. Diferenciaremos entre aprendizaje estructural y aprendizaje paramétrico. El aprendizaje estructural hace referencia a algunos aspectos relacionados con la estructura del conocimiento: el descubrimiento de nuevos síntomas relevantes para una enfermedad o la inclusión de una nueva regla en la base de

conocimiento son ejemplos de aprendizaje estructural. Con el aprendizaje paramétrico nos referimos a la determinación de los parámetros necesarios para construir la base de conocimiento: la estimación de frecuencias o probabilidades asociadas a síntomas o enfermedades es un ejemplo de aprendizaje paramétrico. Otra característica de los ES es su habilidad para obtener experiencia a partir de los datos disponibles; estos datos los pueden obtener tanto expertos como no expertos, y pueden ser utilizados por el subsistema de adquisición del conocimiento y por el subsistema de aprendizaje.

La figura 4.7 muestra cada uno de los elementos descritos en los párrafos anteriores, así como los distintos flujos de información que existen entre ellos (representados mediante flechas).

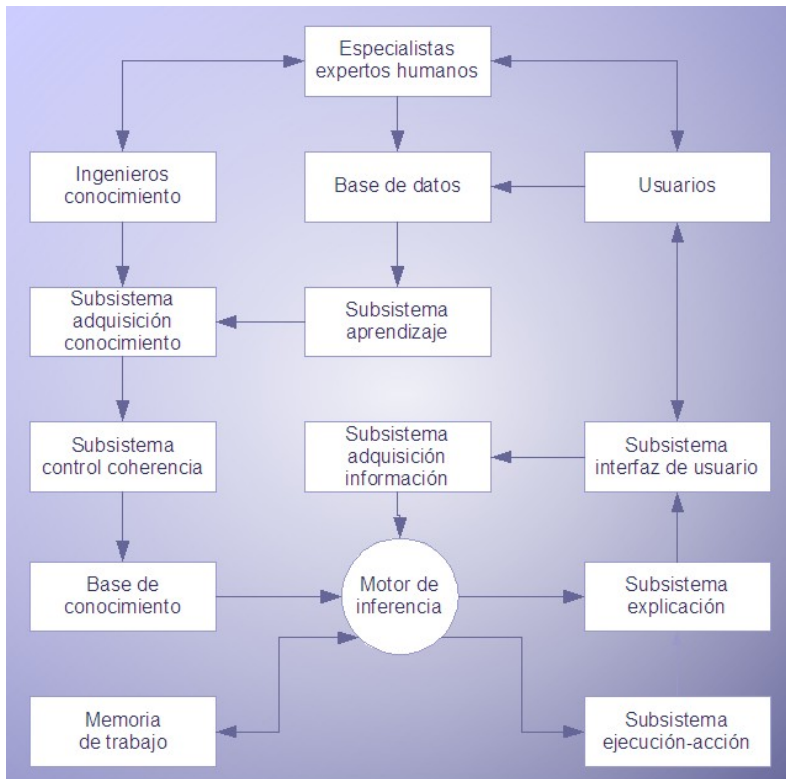


Figura 4.7. Componentes típicos de un sistema experto y sus conexiones (Castillo *et al.*, 1998).

A través del estudio de estos componentes se pueden deducir algunas de las tareas que realizan los sistemas expertos:

- Adquirir conocimiento y verificar su coherencia (puede ayudar a los expertos a proporcionar un conocimiento coherente).
- Almacenar (memorizar) conocimiento.
- Preguntar cuándo se requiere nuevo conocimiento.
- Aprender de la base de conocimiento y de los datos disponibles.
- Realizar inferencia y razonamiento en situaciones deterministas y de incertidumbre.
- Explicar conclusiones o acciones tomadas.
- Comunicarse con los expertos humanos, y no tan expertos, y con otros ES.

4.5.3 Ventajas y limitaciones

A partir del estudio de los sistemas expertos y de sus aplicaciones se pueden entrever una serie de ventajas y limitaciones de estos sistemas. Dentro del primer grupo, podemos destacar las siguientes (Justo, 2004; Aznar, 2004):

- Mayor disponibilidad. La experiencia esta disponible en cualquier ordenador sobre el que funcione nuestro sistema.
- Permanencia. A diferencia de un experto humano un ES no envejece y, por tanto, no sufre pérdida de facultades con el paso del tiempo.
- Fiabilidad. Los ES no se ven afectados por condiciones externas, funcionan a plena capacidad y no se ven influenciados por sus emociones. Un ser humano, por contra, sí puede verse afectado significativamente (cansancio, presión, fatiga, sentimientos, etc.).
- Rapidez. Un ES obtiene información y realiza cálculos numéricos mucho más rápido que cualquier ser humano. Un sistema experto es la solución ideal cuando se requiere una respuesta en tiempo real.
- Un ES permite mayor accesibilidad a la experiencia gracias a la posibilidad de duplicación y transferencia. Esto es especialmente útil cuando el número de expertos en la empresa sea escaso o nulo.
- Bajo coste a medio y largo plazo. A pesar de que el coste de desarrollo inicial puede ser elevado, gracias a la capacidad de duplicación el coste finalmente es bajo.
- Experiencia múltiple. El conocimiento de varios especialistas puede estar disponible de manera simultánea.

- Mayor confiabilidad. Al proporcionar una segunda opinión los sistemas expertos incrementan la confianza en que un especialista haya tomado la decisión correcta.
- Un sistema experto puede trabajar en entornos peligrosos o dañinos para el ser humano.
- Explicación. El ES puede explicar de forma clara y detallada el razonamiento que conduce a una conclusión, aumentando la confianza y brindando la confiabilidad necesaria para su utilización en casos futuros.
- Los ES se pueden emplear en la formación de nuevo personal, lo que posibilita que el número de expertos crezca de manera incremental.
- Los sistemas expertos también pueden ser utilizados por personas no especializadas para resolver problemas.
- El conocimiento puede ajustarse o reexaminarse, lo que aumenta su calidad. El conocimiento de los especialistas humanos debe disponerse clara y formalmente para introducirlo en el sistema. Por ello, y dado que se dispone explícitamente del conocimiento, en lugar de tenerlo implícito en la mente del especialista, puede examinarse para corregirlo, darle más consistencia y completarlo.
- La separación entre base de conocimiento y motor de inferencia proporciona una gran flexibilidad al ES. Además, esta separación del conocimiento y del procedimiento lógico permite un mejor y más fácil mantenimiento.
- La mayoría de los ES han llegado a las mismas conclusiones que los expertos humanos en una gran variedad de casos, verificando que los ES son casi tan buenos como sus equivalentes humanos.
- El sistema experto, al dedicarse a casos en donde no necesita asistencia o supervisión de un experto humano, permite el ahorro de tiempo a este último para que dedique todo su esfuerzo y tiempo a casos que excedan la capacidad del propio sistema experto.
- En definitiva, permite aumentar la productividad.

Sin embargo, también poseen algunas limitaciones importantes (Justo, 2004; Berliaga, 2004):

- Con un experto humano podemos mantener una conversación informal mientras que con un ES esto no es posible.
- Capacidad de aprendizaje. Cualquier persona aprende con relativa facilidad de sus errores y de errores ajenos; dotar a un ES de esta capacidad es muy complicado.

- Un sistema experto es incapaz de aplicar el sentido común³⁷ para resolver situaciones complejas ni controlar situaciones ambiguas.
- Perspectiva global. Un experto humano es capaz de distinguir cuáles son las cuestiones relevantes de un problema y separarlas de cuestiones secundarias.
- Un sistema experto carece de sentidos y, por tanto, de capacidad sensorial. Además, por ser máquinas o sistemas no poseen “conciencia” de su propio alcance y limitaciones.
- Estos programas son poco flexibles a los cambios. Frente a esto, una persona es sumamente flexible en la captación de datos para la resolución de un problema.
- Un sistema experto no es capaz de manejar conocimiento poco estructurado.
- Las expectativas creadas alrededor de esta técnica seducen a muchas personas pero en numerosas ocasiones crean aspiraciones imposibles de satisfacer.
- La extracción del conocimiento es el problema más complejo que se les plantea a los ingenieros de conocimiento, sobre todo porque todavía existen multitud de problemas ligados a su representación.
- Los sistemas expertos son incapaces de reconocer y solucionar un problema para el que su propio conocimiento es inaplicable o insuficiente. El ser humano posee la capacidad de aplicar todos los conocimientos adquiridos para solucionar los problemas que se le presenten (aún estando fuera del campo de su especialidad) consiguiendo un éxito que disminuye conforme se aleja de su campo de especialización. Esto no sucede con los sistemas expertos, ya que al alejarse de su dominio de aplicación fracasan catastróficamente y no de forma gradual.
- Los motores de inferencia poseen algunos límites.
- Las interfaces de usuario no son lo suficientemente amigables.
- Elevado coste a corto plazo en dinero y tiempo.
- Por lo general, la explicación que ofrece un ES se limita a la simple enumeración de las reglas que utilizó para llegar a una determinada conclusión o acción.

37 Para un sistema experto no hay nada obvio. Por ejemplo, un sistema experto sobre medicina podría admitir que un hombre llevase 40 meses embarazado, a no ser que se le especificase que ese hecho no es posible.

4.5.4 Estudio de la viabilidad de un sistema experto

El desarrollo o adquisición de un sistema experto suele ser, por lo general, bastante costoso. Sin embargo, tanto el mantenimiento como el coste marginal asociado a su uso repetido son relativamente bajos. Además, la ganancia en términos monetarios, de tiempo y de precisión como consecuencia del uso de los sistemas expertos es muy alta y la amortización muy rápida. Resulta evidente, por tanto, que antes de decidirse a desarrollar o adquirir un sistema experto primero conviene realizar un análisis de su factibilidad así como un estudio de los posibles costes y beneficios asociados a su implantación.

A la hora de determinar si se debe o no abordar la resolución de un problema mediante la tecnología de los sistemas expertos, Waterman (Waterman, 1986) considera que son tres los aspectos fundamentales a tener en cuenta:

- En primer lugar, se debe determinar si la construcción del sistema experto es posible, para lo que se deben cumplir una serie de requisitos relativos tanto a los expertos humanos como al problema. En concreto, han de existir verdaderos expertos en el dominio que estén disponibles e interesados en ofrecer su conocimiento y experiencia. Además, estos expertos han de estar de acuerdo en las soluciones a los problemas así como ser capaces de estructurar y explicar los métodos que utilizan para resolver problemas. Por último, el problema no ha de ser excesivamente difícil ni requerir el uso de sentido común.
- Una vez estimada la posibilidad de construcción del sistema experto se debe considerar si está justificada la automatización. Para lograr esta justificación se deben dar algunas de las siguientes razones que aconsejen automatizar la tarea: que se prevea una alta rentabilidad, que los expertos escaseen, que se precise el conocimiento en diversos lugares o que se precise en situaciones peligrosas o en entornos hostiles.
- Una vez justificado, hay que determinar si es adecuado hacer un sistema experto. Para la adecuación deben darse una serie de características: que el procesamiento que se requiera sea más simbólico que numérico, que el conocimiento se base más en reglas de comportamiento que en algoritmos y que se deba manejar conocimiento inexacto e incompleto.

4.5.5 Tipos de sistemas expertos

Existen tres tipos principales de sistemas expertos: a) basados en reglas, b) basados en redes bayesianas y c) basados en casos o CBR (*Case Based Reasoning*). En cada uno de ellos, la solución a un problema planteado se obtiene aplicando bien a) reglas heurísticas, apoyadas generalmente en lógica difusa para su evaluación y aplicación, bien b) redes bayesianas, basadas en la estadística y el teorema de Bayes, o bien c) el razonamiento basado en casos, donde la solución a un problema similar planteado con anterioridad se adapta al nuevo problema.

BASADOS EN REGLAS

En nuestra vida cotidiana podemos encontrar multitud de situaciones complejas gobernadas por reglas deterministas, en las que el estado actual del sistema depende de su estado anterior y de las acciones sobre el entorno: este es el caso de los sistemas de control de tráfico, sistemas de seguridad, transacciones bancarias, etc. Los sistemas expertos basados en reglas constituyen una herramienta eficiente para tratar estos problemas.

Las reglas deterministas constituyen la más sencilla de las metodologías utilizadas en sistemas expertos. Los datos hacen referencia a las verdades o realidades irrefutables de una situación particular (por ejemplo, en un sistema de diagnóstico médico serían los síntomas que posee el paciente). Estos datos, al ser variables de una situación a otra, se almacenan en la memoria de trabajo. La base de conocimiento contiene las variables y el conjunto de reglas³⁸ que definen el problema; los conocimientos almacenados en este módulo son estáticos y permanentes de una aplicación a otra, excepto que se incorporen elementos de aprendizaje, por ejemplo a través del subsistema de aprendizaje. El motor de inferencia obtiene las conclusiones aplicando la lógica clásica³⁹ a los datos de la memoria de trabajo y las reglas de la base de conocimientos.

38 Por regla se entiende una proposición lógica que relaciona dos o más objetos (datos) e incluye dos partes, la premisa y la conclusión. Cada una de estas partes consiste en una expresión lógica con una o más afirmaciones objeto-valor conectadas mediante los operadores lógicos *y*, *o*, o *no*. Una regla se escribe normalmente como “Si premisa, entonces conclusión”.

39 En la lógica clásica una proposición sólo admite dos valores: puede ser verdadera o falsa. Por eso se dice que la lógica usual es binaria.

Pero los sistemas basados en reglas sólo pueden aplicarse a situaciones deterministas, por lo que multitud de casos prácticos que implican incertidumbre quedan fuera del alcance de estos sistemas. Por ejemplo, hay situaciones en la vida cotidiana en las que los problemas que se presentan no poseen una solución muy específica en términos exactos. Contestar a preguntas del tipo ¿cuántas personas son muy bajas?, o ¿en qué ciudades hace mucho frío?, es relativamente fácil para un ser humano, pero el problema se complica cuando pretendemos trasladar estas cuestiones a un sistema informático para que las resuelva.

Resulta necesario, por tanto, extender la lógica clásica para incorporar dicha incertidumbre. Una de las soluciones que empezaron a utilizarse en la década de los setenta consistió en el uso de una nueva teoría denominada lógica difusa⁴⁰ o lógica borrosa (FL, *Fuzzy Logic*), en la que se usan modelos matemáticos para trasladar nociones subjetivas, como caliente/tibio/frío o alto/normal/bajo, a valores concretos que puedan ser manipulados por los ordenadores. El uso de esta teoría en los sistemas expertos basados en reglas dio lugar al nacimiento de los sistemas expertos basados en la lógica difusa o, simplemente, sistemas expertos difusos. El núcleo de estos sistemas es el que aparece en la figura 4.8.

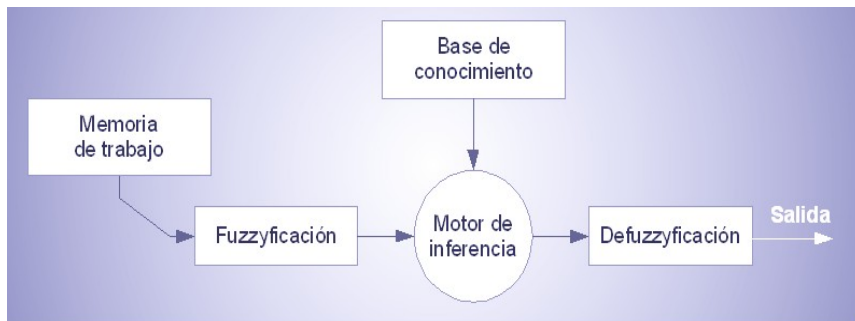


Figura 4.8. El núcleo de un sistema experto de lógica difusa.

Un sistema de lógica difusa o lógica borrosa convierte variables de entrada (cuantitativas y cualitativas) en variables lingüísticas a través de funciones de pertenencia o conjuntos difusos (*fuzzyfication*), que son evaluados mediante reglas difusas del tipo si-entonces a través del motor de inferencia. A

40 Esta teoría fue propuesta por L.A. Zadeh en el año 1965 (Zadeh, 1965).

continuación, las salidas del sistema se convierten en valores nítidos (*crisp*) mediante un proceso de concreción (*defuzzyfication*), que permite brindar información para la toma de decisiones. Por su parte, la base de conocimientos se encarga de contener tanto el conjunto de reglas del sistema como la base de datos donde se definen las funciones de pertenencia a los conjuntos difusos.

Un sistema de lógica difusa utiliza cualquier tipo de información y la procesa de manera similar a como lo haría un ser humano; por ello, los sistemas de lógica difusa son adecuados para operar con información cualitativa, inexacta e incierta (en definitiva, para tratar procesos complejos), lo que la hace una alternativa interesante para modelar problemas de toma de decisiones.

BASADO EN REDES BAYESIANAS

La incertidumbre es uno de los inconvenientes con que se encuentra cualquier persona en el momento de resolver un problema o cuando debe tomar una decisión. En general, los conocimientos que posee una persona en un momento dado poseen un cierto grado de incertidumbre. Las razones que permiten explicar la existencia de incertidumbre en el conocimiento proceden de tres fuentes distintas: la información de partida, el propio conocimiento y, finalmente, la representación de dicho conocimiento.

Con respecto a la información, ésta puede ser incompleta (falta de análisis en medicina, falta de variables de campo en sistemas de control), poco confiable (medidores poco fiables, instrumentos imprecisos, análisis poco confiables) o puede verse afectada por la existencia de ruido y distorsión durante su recopilación. El conocimiento, por su parte, puede ser impreciso (si tiene dolor de cabeza posiblemente tiene gripe) o contradictorio (si tiene dolor de cabeza es probable que tenga gripe, pero también es posible que no). Además, la representación puede no ser adecuada (no se seleccionó la representación idónea para la aplicación) o puede que carezca de poder descriptivo (las representaciones no permiten representar el conocimiento del dominio como lo expresaría el experto).

Como se ha mencionado, los sistemas expertos basados en reglas muestran una evidente incapacidad a la hora de tratar situaciones que posean un cierto grado de incertidumbre, ya que las reglas con las que trabajan son deterministas. En los primeros sistemas expertos, se eligió la probabilidad

como medida para tratar la incertidumbre⁴¹ pero, desgraciadamente, muy pronto se encontraron algunos problemas debidos, en su mayor parte, al uso incorrecto de algunas hipótesis de independencia utilizadas para reducir la complejidad de los cálculos. Como resultado, en las primeras etapas de los sistemas expertos la probabilidad fue considerada como una medida de incertidumbre poco práctica. Esta pobre valoración propició que surgieran, en la literatura de la época, algunas medidas de incertidumbre alternativas, como es el caso de la lógica difusa descrita en el apartado anterior.

Sin embargo, con la aparición de las redes probabilísticas (principalmente las redes Bayesianas y Markovianas), la probabilidad ha resurgido de forma espectacular y es, hoy en día, la más intuitiva y la más aceptada de las medidas de incertidumbre (Luger y Stubblefield, 1989). Lindley (Lindley, 1987), por ejemplo, dice: “La única descripción satisfactoria de la incertidumbre es la probabilidad. Esto quiere decir que toda información incierta debe estar en forma de una probabilidad, que varias incertidumbres deben ser combinadas usando reglas de probabilidad, y que el cálculo de probabilidades es adecuado para manejar situaciones con incertidumbre. Las descripciones alternativas de la incertidumbre son innecesarias”.

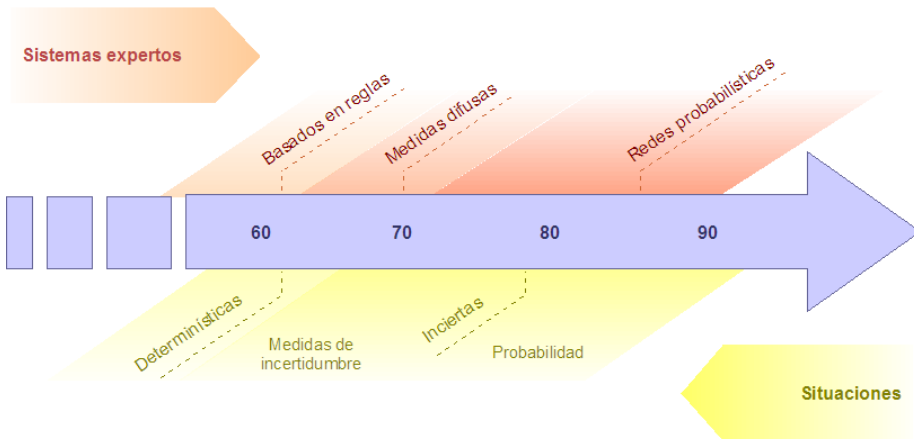


Figura 4.9. Diferenciación gráfica de los ES basados en reglas, difusos y de redes probabilísticas en base a su desarrollo temporal y a la naturaleza de las situaciones que manejan.

41 Véase (Cheeseman, 1985) o (Castillo y Álvarez, 1991).

Si el núcleo de los sistemas expertos basados en reglas es el conjunto de reglas que describen las relaciones entre los objetos (las variables), en los sistemas expertos probabilísticos las relaciones entre las variables se expresan mediante su función de probabilidad conjunta. Por ello, la función de probabilidad conjunta forma parte de lo que se denomina conocimiento.

Para comprender mejor estos sistemas no hay nada mejor que establecer una breve comparativa de los sistemas expertos basados en reglas con aquellos basados en la probabilidad. Por ello, en los próximos párrafos se discuten algunas de las analogías y diferencias, ventajas y desventajas de cada tipo de sistema experto y de la estructura (lógica o probabilística) en la que se basan (Castillo *et al.*, 1997):

1. *Base de conocimiento.* El conocimiento de un ES basado en reglas está constituido por un conjunto de hechos y reglas. El conocimiento de un sistema experto basado en probabilidad consiste en el espacio de probabilidad, que incluye las variables, sus posibles valores, y su función de probabilidad conjunta (FPC). Por otra parte, los datos de ambos sistemas constituyen la evidencia asociada a los casos a analizar. La base de conocimiento en los sistemas expertos basados en reglas es fácil de implementar, puesto que sólo es necesario utilizar elementos simples (conjuntos de valores, premisas, conclusiones, reglas, etc.). Sin embargo, el conocimiento que puede almacenarse es limitado cuando se compara con el de los sistemas expertos basados en probabilidad. Un inconveniente de los ES probabilísticos es el alto número de parámetros que manejan, lo que hace que sea difícil su especificación y definición.
2. *Motor de inferencia.* En los sistemas expertos basados en reglas las conclusiones se obtienen de los hechos aplicándoles diferentes estrategias de inferencia, tales como *Modus Ponens*, *Modus Tollens* y encadenamiento de reglas. Por ello, el motor de inferencia es rápido y fácil de implementar. En los sistemas expertos basados en probabilidad, el motor de inferencia es más complicado que en el caso de los sistemas expertos basados en reglas. El motor de inferencia de un sistema experto probabilístico se basa en la evaluación de las probabilidades condicionales utilizando uno o varios métodos propuestos por los diferentes tipos de sistemas expertos probabilísticos. El grado de dificultad depende del modelo seleccionado y varía desde baja, para los modelos de independencia, a alta, para los modelos de dependencia generales.

3. *Subsistema de explicación.* La explicación es fácil en el caso de los sistemas expertos basados en reglas, ya que en todo momento se sabe qué reglas se han empleado para obtener la conclusión.: el motor de inferencia sabe qué reglas se han utilizado en el encadenamiento y han contribuido a obtener conclusiones y qué reglas se han utilizado sin éxito. En el caso de los sistemas expertos basados en la probabilidad, la información sobre qué variables influyen en otras está codificada en la función de probabilidad conjunta. Por ello, la explicación se basa en los valores relativos de las probabilidades condicionales que miden los grados de dependencia. Una comparación de las probabilidades condicionales para diferentes conjuntos de evidencia permite analizar sus efectos en las conclusiones.
4. *Subsistema de aprendizaje.* En los sistemas expertos basados en reglas el aprendizaje consiste en incorporar nuevos objetos (nuevas reglas y/o nuevos hechos) o modificar los objetos existentes. En los sistemas expertos probabilísticos, el aprendizaje conlleva incorporar o modificar la estructura del espacio de probabilidad: variables, conjunto de posibles valores, o los parámetros (valores de las probabilidades).

Para comprender mejor las diferencias entre ambos tipos de sistemas, la tabla 4.2 resume todas y cada una de las consideraciones formuladas en los párrafos anteriores.

	Basados en reglas	Probabilísticos
<i>Base de conocimiento</i>	Reglas, hechos	Variables, FPC, hechos
<i>Motor de inferencia</i>	Estrategias de inferencia, encadenamiento de reglas	Probabilidad condicional, métodos de evaluación
<i>Subsistema de explicación</i>	Basado en reglas activas	Basado en probabilidad condicional
<i>Subsistema de aprendizaje</i>	Incorporar nuevos objetos o modificar los existentes	Cambio en modelo probabilístico

Tabla 4.2. Una breve comparación entre los sistemas expertos basados en reglas y aquellos basados en la probabilidad. Adaptado de (Castillo *et al.*, 1997).

BASADO EN CASOS

El enfoque tradicional de la AI se basa en construir un sistema de la nada, tratando de resolver los problemas mediante la construcción de soluciones basadas en principios generales (o reglas) y en un conocimiento profundo de los expertos del tema. Los peligros de la adquisición de los conocimientos de un experto humano han sido bien documentados y han dado lugar a la expresión “la adquisición de conocimientos es el cuello de botella” acuñada por Feigenbaum (Feigenbaum, 1977). En contraste, el razonamiento basado en casos (CBR, *Case Based Reasoning*) recuerda casos “pasados” similares al problema actual y transforma la solución correspondiente para resolver el nuevo problema.

El CBR tiene su origen en los trabajos de Roger Shank sobre la dinámica de la memoria (Schank, 1982), en los que describió un enfoque de razonamiento basado en la misma. Según J. Kolodner (Kolodner, 1993) el razonamiento basado en casos puede significar “adaptar viejas soluciones para cubrir nuevas demandas, usar viejos casos para explicar o criticar nuevas soluciones, razonar a partir de precedentes para interpretar una nueva situación, o crear una solución equitativa a un nuevo problema”. En síntesis, esta línea de investigación en AI destaca la importancia de utilizar, al igual que lo hacemos los seres humanos, experiencias anteriores para enfrentarnos a una situación nueva.

El paradigma de razonamiento que existe detrás del CBR es muy simple: se recuperan (se recuerdan) los casos “pasados” más próximos (en características) a la situación del problema actual, y la solución correspondiente al caso que más se aproxime se transforma para resolver el nuevo problema. La ventaja aquí es que se ahorra mucho tiempo y esfuerzo ya que se pueden volver a extraer lecciones, explicaciones, planes o soluciones (por ejemplo, el régimen de tratamiento para un nuevo paciente) simplemente reutilizando los resultados de casos anteriores (de los pacientes), lo que se aproxima suficientemente a como un experto humano podría llevar a cabo la resolución de problemas.

Obviamente, el inconveniente de utilizar este tipo de razonamiento radica en que es necesario que exista una gran “base de datos de casos” de la que extraer los casos antiguos a partir de los cuales construir la nueva solución; sin embargo, existen razones éticas y prácticas por las que a veces esa base de datos puede ser bastante difícil de encontrar. Por otra parte, también debe

existir una buena comprensión del dominio por lo que no es adecuado, por ejemplo, cuando abordamos nuevas áreas en las que hay información incompleta o existe un conocimiento incorrecto o controvertido.

A continuación se detallan los principales pasos involucrados en el razonamiento basado en casos (Ong y Narasimhalu, 1998):

1. Se analiza un nuevo problema/caso y se representa de forma tal que el sistema CBR pueda recuperar los últimos casos relevantes. El objetivo es recuperar casos útiles, es decir, aquellos que tienen el potencial de aportar algo en la resolución del nuevo problema.
2. Una vez se han recuperado los casos relevantes, estos se clasifican (según la similitud del conocimiento) y se selecciona el mejor subconjunto, el caso o casos más prometedores.
3. A menudo, un caso anterior no encaja exactamente en el nuevo, por lo que es necesario modificar y hacer cambios a la solución anterior para adaptarla a la nueva situación. Este proceso, conocido como adaptación, puede variar desde una pequeña sustitución de los valores a cambios estructurales. Qué adaptación y cómo se ha de hacer depende del dominio de conocimiento.
4. A continuación se propone al usuario la solución inicial de este nuevo caso.
5. La solución propuesta es probada y evaluada. Los resultados obtenidos se analizan y si no se alcanzan las expectativas establecidas se suministra una explicación de los resultados anómalos. Tanto las explicaciones de los fracasos como los intentos de repararlos se almacenan para lograr que en el futuro los fracasos se puedan predecir y evitar.
6. El nuevo caso se actualiza en la biblioteca de casos para usarse en el futuro. Al añadir nuevas situaciones y casos en la biblioteca de casos, el sistema está realizando, en realidad, un proceso de aprendizaje gradual. Esto es especialmente útil para dominios dinámicos que necesitan “mantenerse en el tiempo”.

El CBR es, por tanto, un paradigma de resolución de problemas y de aprendizaje, que encara la reutilización y adaptación de soluciones almacenadas de problemas similares, donde las soluciones son almacenadas en una base de casos cuidadosamente indexada (Watson, 1997). Puede verse como un proceso cíclico que combina resolución de un problema y aprendizaje de su experiencia (figura 4.10).

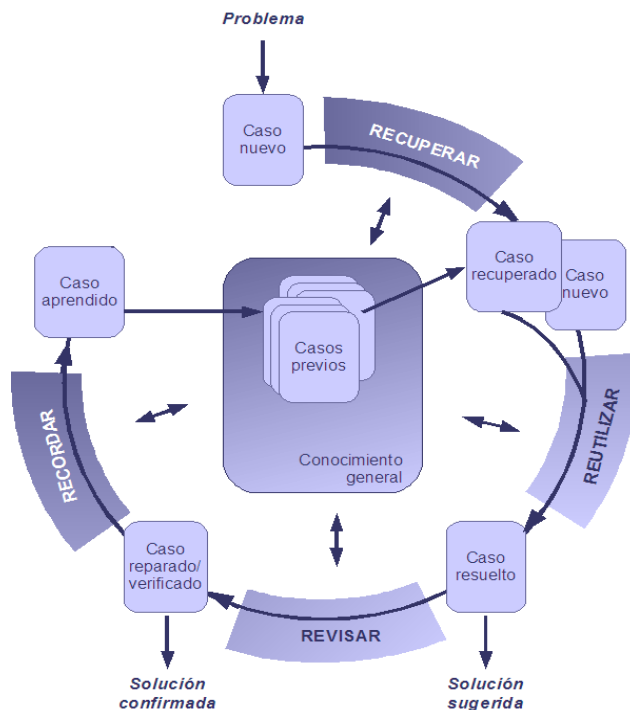


Figura 4.10. El ciclo del CBR. Adaptado de (Aamodt y Plaza, 1994; Watson, 1997)

Deben destacarse dos puntos fundamentales que caracterizan a los sistemas CBR. Por un lado, el aprendizaje no es un módulo que se agrega a un sistema basado en conocimiento, sino que constituye una parte básica del desarrollo, integrándose en la metodología con la que se aborda la resolución de problemas. Por otro, el papel dinámico y central que adquiere la memoria, que no es ya un simple receptáculo de hechos establecidos en el curso de una consulta, sino que debe organizarse para permitir la memorización de episodios, y su correcto indexado⁴², para poder evocarlos cuando resulten útiles en el análisis de casos futuros.

42 Puede afirmarse que estos sistemas aprenden fundamentalmente de dos formas: almacenando nuevos casos y asignando o adaptando los índices correspondientes en la base de casos. El sistema se volverá más eficiente en su razonamiento y mejorará su actuación al incrementar su biblioteca de casos disponibles, pero también como consecuencia de una mejor clasificación de estos nuevos casos en la memoria.

4.5.6 Utilidad de un sistema experto

Existen múltiples razones por las que los investigadores y profesionales utilizan la tecnología de los sistemas expertos. Entre las más importantes se encuentran las que se comentan a continuación (Castillo *et al.*, 1998):

- Con la ayuda de un sistema experto, personal con poca experiencia puede resolver problemas que requieren un conocimiento de experto. Esto es también importante en casos en los que hay pocos expertos humanos. Además, el número de personas con acceso al conocimiento aumenta con el uso de sistemas expertos.
- El conocimiento de varios expertos humanos puede combinarse, lo que da lugar a ES más fiables, ya que se obtiene un sistema experto que combina la sabiduría colectiva de varios expertos humanos en lugar de la de uno solo.
- Los sistemas expertos pueden responder a preguntas y resolver problemas mucho más rápidamente que un experto humano. Por ello, estos sistemas son muy valiosos en aquellas situaciones en las que el tiempo de respuesta es crítico.
- En algunos casos, la complejidad del problema impide al experto humano resolverlo. En otros, la solución de los expertos humanos no es fiable. Debido a la capacidad de los ordenadores de procesar un elevado número de operaciones complejas de forma rápida y aproximada, los ES suministran respuestas rápidas y fiables en situaciones en las que los expertos humanos no pueden.
- Pueden utilizarse para realizar operaciones monótonas, aburridas e incómodas para los humanos. En verdad, los sistemas expertos pueden ser la única solución viable en una situación en la que la tarea a realizar desborda al ser humano (por ejemplo, un avión o una cápsula espacial dirigida por un sistema experto).
- Se pueden obtener enormes ahorros mediante el uso de sistemas expertos.

A modo de resumen, el uso de los sistemas expertos se recomienda especialmente cuando el conocimiento es difícil de adquirir o se basa en reglas que sólo pueden ser aprendidas de la experiencia; cuando la mejora continua del conocimiento es esencial; cuando el problema está sujeto a reglas o códigos cambiantes; cuando los expertos humanos son caros o difíciles de encontrar; y cuando el conocimiento de los usuarios sobre el tema es limitado.

4.5.7 ¿Por qué son adecuados los ES en nuestra solución?

Para comprender esta decisión conviene detenerse un instante a analizar cómo toman sus decisiones las empresas. En casi todas ellas, y sobre todo en las de pequeño y mediano tamaño, las decisiones no se toman sólo en base a enrevesadas funciones y algoritmos matemáticos ni a complejas hipótesis estadísticas; más bien, los altos ejecutivos recurren a su “experiencia”, “conocimiento” e “intuición”. En ocasiones, este conocimiento puede llegar a relegar a un segundo plano toda la información recopilada para ese instante.

Esta realidad también es aplicable a muchas de las decisiones concernientes a la selección de socios: como hemos visto, el proceso de selección no puede limitarse a un simple emparejamiento de habilidades-necesidades, y se han de tener en cuenta otras variables de tipo subjetivo y cualitativo que tienen mucho que ver con el conocimiento y la experiencia adquirida por los directivos a lo largo de los años. Pero, ¿cómo podemos representar ese “conocimiento”, “experiencia” e “intuición” en una aplicación informática?

Está claro que para poder hacerlo necesitamos un sistema que “piense”, que “razone” como lo haría un especialista humano; necesitamos, en definitiva, un ES capaz de simular el proceso de memorización, razonamiento, comunicación, etc., y de acción de un experto humano. Gracias a estos sistemas podremos tomar decisiones basadas no sólo en mecanismos de emparejamiento matemáticos sino también en base a la experiencia y el conocimiento acumulado por los directivos y expertos en el área.

Por tanto, con un sistema experto seremos capaces de dotar al sistema multiagente de un módulo de decisión adecuado a las necesidades que plantea la selección de socios en las VE dinámicas. La pregunta que surge entonces es, ¿qué tipo de ES debemos emplear?. Está claro que para nuestros propósitos iniciales nos sobra con un sistema basado en reglas que sea capaz de reflejar el conocimiento de los expertos. A fin de cuentas, carecemos de la información necesaria que nos pudiese dar la posibilidad a incorporar cualquier otro tipo de sistema experto.

Sin embargo, y es una de las ventajas de incluir el sistema experto como un módulo de decisión dentro del sistema multiagente, en el futuro siempre sería posible emplear otro tipo de sistemas: podríamos incorporar, por ejemplo, la lógica difusa o sustituir el ES basado en reglas por un sistema experto CBR (siempre y cuando se disponga de una base de casos adecuada).

4.6 RESUMEN

La selección de socios constituye uno de los principales retos a superar para conseguir que el paradigma de VE dinámica alcance una aplicación práctica a gran escala. Si pretendemos dirigir nuestros esfuerzos en esta dirección nos vemos obligados a ofrecer una solución innovadora que responda a las necesidades de la VE dinámica de un modo eficiente. Para lograr tal propósito, es imprescindible disponer primero de un modelo de negocio adecuado que tenga en cuenta las consideraciones realizadas en los capítulos previos. Sólo entonces seremos capaces de desarrollar una solución informática que apoye la toma de decisiones de las empresas de un modo innovador y efectivo.

Con este objetivo en mente, a lo largo del capítulo hemos podido conocer los distintos enfoques planteados en la literatura para abordar la formación de VE dinámicas. Cada uno de estos enfoques presenta sus ventajas e inconvenientes, por lo que resultaría útil disponer de una nueva alternativa que lograra aunar las primeras y eliminar, o al menos mitigar, los últimos; con el modelo multienfoque presentado en el presente capítulo creemos haber logrado esta meta. En consecuencia, partimos de un modelo que nos permite construir cualquier tipo de VE dinámica, proporcionándonos un mayor control sobre el proceso de formación.

Una vez establecido el modelo de partida llega el momento de iniciar la construcción de la plataforma software que permita conocer realmente su eficiencia y operatividad con respecto al problema que nos ocupa. En otras palabras, debe comenzar la planificación de la solución informática, cuyo primer paso consiste en la justificación de la tecnología software que emplearemos en su construcción.

En este sentido, hemos comenzado presentado y justificando el uso de los sistemas multiagente. Esta tecnología parece especialmente prometedora para desarrollar sistemas software que se enfrenten a problemas que hasta el momento no han sido resueltos de forma satisfactoria (sistemas abiertos, complejos y ubicuos) y para simplificar el tratamiento de los sistemas distribuidos (con elementos dispersos por naturaleza que están fuertemente interrelacionados). Además, es la solución que mejores resultados está dando en la literatura de investigación relacionada. En definitiva, se trata de una tecnología totalmente adecuada al problema que nos atañe.

Por otro lado, si pretendemos acercarnos lo más posible al proceso de decisión que se da en las empresas, está claro que necesitamos un sistema que “piense” y que “razone” como lo haría un experto humano; necesitamos, en definitiva, un sistema experto. Gracias a esta tecnología podremos tomar decisiones basadas no sólo en mecanismos de emparejamiento matemáticos sino también en base a la experiencia y el conocimiento acumulado por los directivos y expertos en el área.

A lo largo del capítulo siguiente continuaremos con la planificación centrándonos, en este caso, en la elección de las herramientas y metodologías de desarrollo con las que construir la plataforma software.

ELECCIÓN DE LAS
HERRAMIENTAS Y
METODOLOGÍAS DE
DESARROLLO

5.1 INTRODUCCIÓN

Los primeros desarrolladores de sistemas multiagente se encontraron ante una tecnología difusa, sin una teoría ampliamente aceptada, sin especificaciones que unificaran las soluciones y sin herramientas de apoyo que facilitaran su labor. Con el paso del tiempo esta situación ha cambiado significativamente: la teoría parece clarificarse en muchos aspectos, existen algunos estándares bastante aceptados y se dispone de un conjunto bastante numeroso de herramientas de desarrollo. En esta nueva situación no parece lógico, por tanto, desarrollar un MAS desde cero, utilizando metodologías y herramientas genéricas o concebidas para otros propósitos.

Aún así, el desarrollo actual de los sistemas multiagente está marcado por la coexistencia de dos filosofías diferenciadas, principalmente, por su carga de componente práctica (GRASIA web, 2007). La primera ve estos sistemas como el resultado de utilizar un lenguaje de especificación de agentes y un conjunto de principios establecidos en modelos operacionales y formales de los MAS. La segunda estudia el MAS como un sistema software que hay que

construir: el desarrollo no parte de cero, sino que utiliza plataformas de desarrollo que proporcionan servicios básicos de comunicación, gestión de agentes y una arquitectura de agente. En ambos casos, y sobre todo cuando el sistema a construir es grande, se necesitan metodologías que estructuren el desarrollo de acuerdo con las prácticas de ingeniería del software.

Los desarrolladores de aplicaciones MAS parecen inclinarse hacia el uso de plataformas de desarrollo debido, en gran parte, al elevado nivel de conocimientos que requiere programar con un lenguaje de agentes. Este hecho ha conducido a la proliferación de multitud de plataformas, algunas de las cuales son implementaciones de estándares de agentes existentes⁴³: JADE, FIPA OS, Grasshopper, Aglet, ABLE, ZEUS, AgentBuilder, etc.

Teniendo presentes estas consideraciones, y con el objeto de seleccionar aquellas técnicas que mejor se adapten a las necesidades del presente trabajo, comenzaremos este capítulo analizando tres aspectos básicos en el desarrollo de MAS: estándares, plataformas y metodologías de desarrollo.

- Un *estándar* es un conjunto de especificaciones que, en el caso de los sistemas multiagente, describen cómo abordar aspectos relacionados con las arquitecturas, movilidad, formato de los mensajes, definición de ontologías, etc. Si los sistemas se elaboran de acuerdo a estas especificaciones se facilita la interoperabilidad entre los mismos, la reutilización de sus componentes y, sobre todo, su comprensión por parte de otros desarrolladores.
- Además de los estándares, los investigadores de sistemas multiagente han elaborado *plataformas* que permiten desarrollar software adaptado a este paradigma. En algunos casos incluyen todos los aspectos del desarrollo, que van desde el análisis hasta su implementación, y en otros sólo esta última etapa.
- Las *metodologías* pueden definirse como un conjunto de métodos o técnicas que ayudan en el desarrollo de un producto software. Pueden abarcar todas las etapas de la ingeniería de software, aunque en la mayoría de los casos se quedan en el análisis y el diseño. Las principales actividades de una metodología son la definición y descripción del problema, construcción de la solución y supervisión de su funcionamiento.

43 JADE (y su versión Leap para PDA) y FIPA OS son implementaciones del estándar FIPA, mientras que Grasshopper y Aglet(IBM) lo son de MASIF.

Al contrario de lo que pudiera suponerse, estándares, metodologías y plataformas de desarrollo no son independientes sino que están estrechamente relacionados; de esta forma, podemos encontrar en el mercado plataformas que son conformes a alguna de las especificaciones existentes y, al mismo tiempo, metodologías que están soportadas por plataformas de desarrollo. La elección de estos tres elementos ha de realizarse, por tanto, de manera conjunta si se quiere sacar el máximo partido a los MAS.

Otra actividad importante en el desarrollo actual de los MAS es la definición de ontologías. Básicamente, una ontología es un colección de conceptos organizados jerárquicamente que permiten definir el conjunto de entidades relevantes en un campo de aplicación determinado, así como las interacciones entre las mismas; para un agente una ontología define, esencialmente, su entorno. A pesar de que la definición de ontologías no es estrictamente necesaria para el desarrollo de MAS, su uso es recomendable dado que facilita el intercambio y la reutilización del conocimiento y permite, además, separar el dominio de conocimiento de los aspectos operacionales.

Por último, también dedicaremos algo de tiempo a tratar los aspectos más interesantes relativos a la elección de la herramienta de desarrollo del sistema experto; a fin de cuentas, de su elección dependerá significativamente el módulo de decisión de nuestro MAS. En concreto, conoceremos los distintos tipos existentes y describiremos algunas de las herramientas más importantes que existen en el mercado. Con toda esta información, y teniendo presente tanto nuestras necesidades como las decisiones adoptadas en los apartados previos (sobre todo las relativas a la elección de la plataforma de desarrollo multiagente), estaremos en disposición de realizar una elección acertada.

5.2 ESTÁNDARES DE DESARROLLO MULTIAGENTE

Como en todas las tecnologías incipientes, las primeras aplicaciones basadas en agentes se desarrollaron partiendo de cero. Cada grupo de investigación, partiendo de conceptos y herramientas diferentes llegaba a soluciones con capacidades muy dispares. Esta situación presentaba una serie de problemas que limitaba la calidad de las soluciones propuestas (García y Pavón, 2000):

- Ausencia de una definición estándar: cada equipo desarrollaba su MAS de acuerdo con las necesidades particulares de cada proyecto, dando lugar a la aparición de múltiples sistemas heterogéneos.

- Duplicación de esfuerzos: cada sistema empezaba desde cero debido a la imposibilidad de intercambiar conocimientos entre distintos equipos de trabajo. No era posible reutilizar arquitecturas, diseños ni componentes para esos sistemas.
- Incapacidad para satisfacer los importantes requisitos demandados por la industria: los agentes se deben integrar con el software y la infraestructura existente para poder afrontar un uso real en la industria y no sólo con carácter investigador. Además, son muy importantes los conceptos de seguridad y escalabilidad.
- Incompatibilidad y pérdida de portabilidad: los sistemas de agentes deberían poder interactuar y cooperar entre sí. Para ello, es necesario que los esfuerzos de desarrollo se coordinen en esta dirección.

La solución a estos inconvenientes pasa por la elaboración de un estándar ampliamente aceptado que permita unificar criterios y, en consecuencia, desarrollar sistemas más prácticos. Ante esta necesidad, diferentes organismos se han implicado en la elaboración de estándares, sobre todo relacionados con el intercambio de conocimiento y la interoperabilidad (García y Pavón, 2000; Flores-Mendez, 2000). Entre estos organismos destacan OMG (*Object Management Group*), W3C (*World Wide Web Consortium*), DARPA-KSE (*Knowledge Sharing Effort*) y FIPA (*Foundation for Intelligent Physical Agents*).

5.2.1 MASIF y CORBA

El estándar MASIF (*Mobile Agent System Interoperabilities Facility*), también conocido por las siglas MAF (*Mobile Agent Facility*), fue desarrollado por el grupo OMG (OMG web, 2006) con la pretensión de garantizar la interoperabilidad entre plataformas de agentes de diferentes vendedores escritas en un mismo lenguaje (Milojicic *et al.*, 1998). Es, ante todo, un estándar concebido para facilitar la creación de sistemas de agentes móviles.

Con este objetivo, MASIF trata de estandarizar tres aspectos decisivos de los MAS como son el control de los agentes (crear el agente, suspender su hilo de ejecución y finalizarlo), la transferencia de agentes entre diferentes nodos de la red y el sistema de identificación de los agentes en función de su localización. Sin embargo, no aborda los aspectos relacionados con la comunicación entre agentes, ya que ésta se puede conseguir mediante la

especificación CORBA (*Common Object Request Broker Architecture*), ni establece cómo deben estructurarse los mensajes y su contenido para garantizar el correcto intercambio de conocimiento.

En el estándar MASIF, el sistema de agentes es la plataforma que permite gestionar los agentes sobre los que tiene control. Para garantizar la interoperabilidad, esta plataforma debe ser capaz de crear agentes, transferirlos, gestionar las localizaciones y garantizar la seguridad del sistema. Esto se consigue gracias a dos interfaces de objetos CORBA (OMG web, 2006): la interfaz *MAFAgentSystem*, que define las operaciones de control de los agentes y de su ciclo de vida, y *MAFFinder*, que establece las operaciones para registrar y localizar agentes y sistemas de agentes.

La especificación MAF está basada, por tanto, en la arquitectura CORBA, un conjunto de especificaciones destinadas a facilitar la interoperabilidad entre objetos o componentes software con independencia del lenguaje en el que estén desarrollados, de la plataforma hardware y del sistema operativo sobre el que se ejecuten⁴⁴. Para conseguir estos objetivos CORBA se apoya en tres pilares fundamentales (Charte, 1999): el lenguaje de definición de interfaces IDL (*Interface Definition Language*), los ORB (*Object Request Brokers*) y el protocolo GIOP (*General Inter-ORB Protocol*) y sus derivados, entre los que destaca IIOP (*Internet Inter-ORB Protocol*).

5.2.2 KQML y KIF

KSE (*Knowledge Sharing Effort*) (KSE web, 2006) es un consorcio centrado en el desarrollo de normas para la realización de sistemas basados en el conocimiento. Sus aportaciones más destacadas han sido las especificaciones del lenguaje KQML (*Knowledge Querying and Manipulation Language*) (Labroue y Finin, 1998), el formato KIF (*Knowledge Interchange Format*) y el desarrollo de Ontolingua, todas ellas orientadas a facilitar el intercambio de conocimiento entre agentes (Finin *et al.*, 1997).

El lenguaje KQML contempla tres niveles de comunicación entre agentes: nivel de contenido, nivel de intención y nivel de coordinación. El *nivel de contenido* se refiere a la información que se pretende comunicar. En

44 Esto se realiza de un modo transparente, es decir, sin que sea necesario conocer la ubicación física del objeto para utilizarlo (Charte, 1999).

particular, KQML no concreta nada sobre este nivel de contenido, por lo que la forma en que se represente la información es una elección de los programadores. Por este motivo, KQML puede transportar tanto una cadena ASCII como una expresión en Lips, Prolog, español, inglés o cualquier otro tipo de representación de la información.

El *nivel de intención o de mensaje* es la parte más importante del estándar y define la intención del mensaje o performativa (*performative*), es decir, el objetivo con el que se envía dicho mensaje (solicitar una acción, informar, etc.). KQML está basado en la teoría de la comunicación humana (*speech act theory*), de modo que el conjunto de intenciones que se establecen se basan en ella. Algunos ejemplos de performativas son los siguientes: *evaluate, ask-if, ask-one, ask-all, reply, tell, achive, cancel, next*, etc.

Finalmente, el *nivel de comunicación* establece la forma en que los agentes dialogan entre sí, lo que permite concretar su conducta frente a la llegada de nuevos mensajes. Básicamente, determina un conjunto de parámetros necesarios para conseguir la comunicación como son el código de la conversación, la identidad del agente que envía el mensaje y la del agente o agentes que lo reciben, el tiempo de espera de la respuesta, el lenguaje y la ontología empleada, etcétera.

5.2.3 FIPA y ACL

El estándar FIPA (FIPA web, 2007), creado y desarrollado por la fundación homónima, se ha convertido en estos últimos años en el principal referente para la creación de sistemas multiagente. La razón principal de este éxito se haya en que abarca todos los aspectos relacionados con la tecnología de agentes (gestión, seguridad, movilidad, comunicación,...) propuestos hasta el momento.

La fundación FIPA (*Foundation for Intelligent Physical Agents*) es un consorcio internacional constituido por organizaciones procedentes tanto del mundo universitario como de la industria. Desde su creación, en 1995, FIPA ha buscado promover el uso de los agentes inteligentes a través del desarrollo de unas especificaciones, de carácter público, que facilitasen la interoperabilidad entre agentes y entre sistemas de agentes heterogéneos en el comercio y la industria. Este trabajo le ha llevado a convertirse, el 8 de junio de 2005, en el undécimo comité de estándares del IEEE.

El estándar FIPA, al igual que MASIF, resuelve todos los aspectos relacionados con la interoperabilidad entre plataformas de agentes móviles con implementaciones diferentes. Además, y a diferencia de este último, incluye especificaciones sobre el intercambio de conocimiento entre agentes; el lenguaje definido para este propósito es ACL (*Agent Communication Language*), que puede considerarse como una evolución del KQML. FIPA también incluye especificaciones sobre la definición de ontologías y lenguajes de contenidos, aspectos no contemplados en los otros estándares.

Se compone de dos conjuntos de especificaciones de diferente contenido: las referencias tecnológicas y las referencias a aplicaciones. Las *referencias a aplicaciones* ofrecen pistas sobre cómo introducir la tecnología de agentes en aplicaciones comerciales (tabla 5.1).

Application	FIPA Nomadic Application Support Specification	SI00014
	FIPA Quality of Service Specification	SC00094
	FIPA Personal Travel Assistance Specification	XI00080
	FIPA Audio-Visual Entertainment and Broadcasting Spec.	XI00081
	FIPA Network Management and provisioning Specification	XI00082
	FIPA Personal Assistant Specification	XI00083
	FIPA Message Buffering Service Specification	XC00092

Tabla 5.1. Referencias a Aplicaciones de FIPA⁴⁵.

Las *referencias tecnológicas*, por su lado, están destinadas a la especificación del conjunto de estándares que deben seguirse para desarrollar sistemas multiagente que puedan interactuar con otros sistemas desarrollados bajo las mismas especificaciones. Éstas se agrupan en cuatro categorías (tabla 5.2): arquitectura abstracta (*Abstract Architecture*), comunicación entre agentes (*Agent Communication*), gestión de agentes (*Agent Management*) y transporte de mensajes (*Agent Message Transport*).

Todo el conjunto de especificaciones tecnológicas tienen como base una arquitectura abstracta (FIPA00001 *FIPA Abstract Architecture Specification*) que establece los elementos abstractos y las relaciones necesarias para elaborar agentes que interactúen en un entorno determinado. A partir de esta especificación se despliega el resto del estándar. Así, bajo la categoría *Agent*

⁴⁵ En la columna derecha de la tabla se ha incluido la denominación informal de la especificación. Se ha optado por esta denominación debido a que incorpora información sobre el estado de la especificación: experimental (X), preliminar (P) o estándar (S).

Management se engloban un conjunto de estándares orientados a definir los aspectos relacionados con la gestión de agentes. De las tres especificaciones agrupadas en esta categoría, sólo la FIPA00023 ha llegado a ser considerada estándar, manteniéndose el resto en estado preliminar.

Abstract Architec	FIPA Abstract Architecture Specification		SC00001	
	FIPA Domains and Policies Specification		PC00089	
Agent Communication	FIPA ACL Message Structure Specification		SC00061	
	FIPA Ontology Service Specification		XC00086	
	Interaction Protocols	FIPA Request Interaction Protocol Specification		SC00026
		FIPA Query Interaction Protocol Specification		SC00027
		FIPA Request When Interaction Protocol Spec.		SC00028
		FIPA Contract Net Interaction Protocol Spec.		SC00029
		FIPA Iterated C. Net Interaction Protocol Spec.		SC00030
		FIPA Brokering Interaction Protocol Spec.		SC00033
		FIPA Recruiting Interaction Protocol Spec.		SC00034
		FIPA Subscribe Interaction Protocol Spec.		SC00035
		FIPA Propose Interaction Protocol Spec.		SC00036
FIPA English Auction Interaction Protocol Spe.		XC00031		
FIPA Dutch Auction Interaction Protocol Spec.		XC00032		
Communi- cative Act.	FIPA Communicative Act Library		SC00037	
Content Language	FIPA SL Content Language Specification		SC00008	
	FIPA CCL Content Language Specification		XC00009	
	FIPA KIF Content Language Specification		XC00010	
	FIPA RDF Content Language Specification		XC00011	
Agent Management	FIPA Agent Management Specification		SC00023	
	FIPA Agent Discovery Service Specification		PC00095	
	FIPA JXTA Discovery Middleware Specification		PC00096	
Agent Message Transport	FIPA Agent Message Transport Service Specification		SC00067	
	FIPA Messaging Interoperability Service Specification		XC00093	
	ACL Representation	FIPA ACL MR ⁴⁶ in Bit-Efficient Spec.		SC00069
		FIPA ACL MR in String Spec.		SC00070
		FIPA ACL MR in XML Spec.		SC00071
	Envelope Representation	FIPA AMT ⁴⁷ Envelope Representation in XML		SC00085
		FIPA AMT Envelope Represent. in Bit Efficient		SC00088
Transport Protocols	FIPA AMT Protocol for IIOP Spec.		SC00075	
	FIPA AMT Protocol for HTTP Spec.		SC00084	
	FIPA AMT Protocol for WAP Spec.		XC00076	

Tabla 5.2. Referencias tecnológicas de FIPA.

46 MR: *Message Representation*.

47 AMT: *Agent Message Transport*.

El conjunto de especificaciones englobadas dentro de la categoría *Agent Communication* proporcionan las características de alto nivel de la comunicación entre agentes. Para ello se especifica un lenguaje de comunicación (FIPA00061: *Agent Communication Language ACL*) basado en la teoría de actos del habla. Además del lenguaje ACL, esta categoría contiene un amplio conjunto de especificaciones agrupadas en tres subcategorías: *Communicative Act* donde se establecen las posibles intenciones con las que se pueden emitir mensajes, *Content Language* donde se proponen diferentes lenguajes para expresar el contenido de los mensajes (sólo FIPA-SL es considerado como estándar: FIPA00008), e *Interacción Protocols* donde se proponen diferentes esquemas de conversación (secuencia de mensajes) entre agentes.

Las especificaciones referentes a la codificación y transporte de mensajes se agrupan bajo la categoría de *Agent Message Transport*. Este conjunto de especificaciones también se clasifican según tres subcategorías (*ACL Representation*, *Envelope Representation*, y *Transport Protocols*) donde se describen diferentes aspectos de bajo nivel referentes al envío de mensajes.

5.2.4 XML

W3C (W3C web, 2006) es una organización que produce estándares para la World Wide Web. Dentro de sus aportaciones, la que más influencia ha tenido en el contexto multiagente es el lenguaje XML⁴⁸ (*Extensible Markup Language*) diseñado, principalmente, para el intercambio de documentos estructurados a través de Internet (Cuenca *et al.*, 2002). Dadas sus características, este lenguaje se ha convertido en la actualidad en un estándar que todos entienden y usan.

XML permite representar documentos que contienen información estructurada, para que ésta puede ser almacenada, transmitida o procesada por algún sistema de gestión de la información. En este sentido su aportación principal a los MAS radica en su posible uso como lenguaje de comunicación entre agentes o como lenguaje de contenidos; de hecho, tanto FIPA-ACL como KQML ya tienen especificaciones en XML. Otra característica interesante de XML es su fácil integración con el lenguaje de programación Java, gracias a los parsers DOM, SAX y JDOM.

48 XML es un subconjunto del SGML (*Standard Generalized Markup Language*).

5.2.5 Discusión

El valor de un estándar está directamente relacionado con su aceptación y adaptación. Si los desarrolladores de software no lo utilizan de poco servirán todos los esfuerzos realizados. Desde este punto de vista, la especificación FIPA destaca significativamente frente al resto de estándares. De hecho, desde su aparición se ha convertido en un referente para la creación y desarrollo de MAS al incluir todos los aspectos de la tecnología multiagente que la comunidad investigadora ha ido proponiendo hasta el momento.

FIPA, al igual que MASIF, resuelve todos los aspectos relacionados con la interoperabilidad entre plataformas de agentes móviles con implementaciones diferentes pero, y a diferencia de este último, también incluye especificaciones sobre el intercambio de conocimiento entre agentes. El lenguaje definido para este propósito es ACL que, como ya se ha comentado, se puede considerar como una evolución del KQML. Además, FIPA incluye especificaciones sobre la definición de ontologías y lenguajes de contenidos, aspectos no contemplados en los otros estándares.

5.3 ENTORNOS DE DESARROLLO MULTIAGENTE

Desde la aparición de la tecnología de agentes, se han desarrollado gran cantidad de herramientas, algunas de ellas generales y otras destinadas a fines específicos como pueden ser la simulación o la movilidad de los agentes. La mayoría de plataformas están basadas en lenguajes ya conocidos como C++, LISP, PROLOG o Java. Este último posee ciertas cualidades que propician que la mayoría de herramientas se basen en él (Ayes, 2003).

Java está compuesto básicamente por tres elementos: un lenguaje orientado a objetos, un conjunto de bibliotecas y una máquina virtual JVM (*Java Virtual Machine*). El código fuente es compilado en un código intermedio llamado *bytecode* que es el que será interpretado por cualquier implementación de la máquina virtual. De esta forma, los programas compilados en *bytecode* pueden ser ejecutados por cualquier plataforma que posea una implementación de la máquina virtual.

Java posee, además, ciertos mecanismos integrados como son la posibilidad de múltiples hilos de ejecución (*threads*), comunicación entre objetos, modelos de objetos distribuidos, tratamiento de eventos, etc. También se

caracteriza por hacer uso de una sintaxis bastante sencilla y por soportar “estrictamente” el paradigma de orientación a objetos. Todo ello hace de Java un lenguaje muy apreciado para el desarrollo de sistemas de objetos distribuidos en entornos heterogéneos y, más concretamente, de los sistemas multiagente.

Sin embargo, el uso o no de java como lenguaje de programación no es la única característica que ha de tenerse presente para seleccionar la plataforma de desarrollo. La elección de un entorno frente a otros es una decisión que marcará significativamente la evolución del sistema a construir, por lo que han de considerarse un gran número de variables. La complejidad de la decisión se ve incrementada, además, por la existencia en el mercado de un amplio abanico de plataformas de desarrollo.

Intentar analizar todas y cada una de estas plataformas supondría un esfuerzo considerable a la par que innecesario. Por todo ello, en este apartado se van a describir cuatro herramientas de desarrollo que según la literatura consultada están bien valoradas: AgentTool, AgentBuilder, ZEUS y JADE.

5.3.1 AgentTool

AgentTool (AgentTool web, 2006) es un entorno gráfico, basado en Java, que pretende ayudar a analizar, diseñar e implementar sistemas multiagente. El diseñador define inicialmente las características del sistema usando una descripción de alto nivel en la que se concreta tanto los tipos de agentes existentes como las comunicaciones que puede haber entre ellos. Una vez realizada esta especificación en el ámbito del sistema, se concretan las características de cada tipo de agente, para lo que el diseñador selecciona o crea una arquitectura de agente y después detalla el comportamiento de cada componente de esa arquitectura (Araújo, 2003).

Tras especificar totalmente el sistema, llega el momento en el que el diseñador debe concretar los detalles como son, por ejemplo, los protocolos específicos de la comunicación o los elementos de seguridad. Una vez definidos todos estos aspectos, el sistema genera código java automáticamente. AgentTool posee una biblioteca de componentes de Inteligencia Artificial, que se pueden reutilizar para construir los agentes que el programador estime necesarios. De esta forma sólo será necesario desarrollar componentes que no existan en la biblioteca.

Las principales ventajas de este entorno derivan del hecho de que soporta todas las fases del desarrollo del sistema, partiendo de una especificación de alto nivel. Además, la metodología que soporta (*MaSE*) es una de las más reconocidas entre los investigadores de este campo.

5.3.2 AgentBuilder

AgentBuilder (AgentBuilder web, 2006) es un entorno de desarrollo bastante completo desarrollado por Reticular Systems, Inc. (Reticular web, 2006). El modelado, que constituye la base de concepción del sistema, se basa en las técnicas de orientación a objetos, mientras que la elaboración de los agentes se realiza a partir de una arquitectura BDI. Como lenguaje de comunicación entre agentes se utiliza KQML, aunque permite al programador construir mensajes según sus propios criterios.

El entorno posee dos componentes implementados en Java: un conjunto de herramientas de desarrollo y un entorno de ejecución. Los agentes desarrollados mediante AgentBuilder son programas Java, por lo que pueden ser ejecutados sobre una JVM. Las herramientas de desarrollo permiten el control del proyecto, la definición de ontologías, agencias (dos o mas agentes que se comunican y cooperan) y agentes, la construcción del aprendizaje, la planificación de los agentes y la depuración.

AgentBuilder soporta una metodología propia desde el análisis a la implementación y posee buenas cualidades en cuanto a la gestión de agentes e intercambio de mensajes. Como contrapartida, el entorno es bastante complejo por lo que los desarrolladores necesitan cierto esfuerzo de aprendizaje. Además, tiene ciertas limitaciones en cuanto a la extensibilidad, la reutilización de código y la implementación de sistemas distribuidos en varias máquinas (Garneau y Delisle, 2002).

5.3.3 ZEUS

ZEUS (Nwana *et al.*, 1999; ZEUS web, 2006) es una plataforma de desarrollo de sistemas multiagente que tiene como objetivo el desarrollo fácil y rápido de aplicaciones basadas en este paradigma. Para ello, presenta un entorno visual de desarrollo que evita tener que ocuparse de los aspectos de bajo nivel del funcionamiento de los agentes.

Desde su aparición, ZEUS se ha convertido en referente de cómo debe ser una herramienta para el desarrollo de sistemas multiagente, sobre todo por la forma en que combina los distintos resultados de investigación en agentes (planificación, ontologías, asignación de responsabilidades, relaciones sociales entre agentes) en un sistema completamente funcional.

La plataforma ZEUS consta de diversos componentes escritos en lenguaje Java, agrupados en tres bibliotecas (Blanco y Pavón, 2000): la biblioteca de componentes de agente, los programas de construcción de agentes y la biblioteca de agentes de utilidad. A continuación se enumeran los elementos y características principales de cada una de estas bibliotecas:

- La *biblioteca de componentes de agente* posee los elementos necesarios para construir agentes. Contiene componentes de planificación y temporización, de comunicación, de estructuras de datos, de interfaz de usuario y de interacción social.
- ZEUS genera código automáticamente a partir de las especificaciones aportadas a través de un entorno visual. Para ello dispone de diversos *programas para la construcción de agentes* que permiten editar las ontologías, los hechos/variables (instancias y valores de los conceptos definidos en la ontología), los agentes, las descripciones de sus tareas, e incluso su organización y coordinación.
- Por último, la *biblioteca de agentes de utilidad* incorpora varios agentes predefinidos que proporcionan algunos servicios de utilidad. Los más importantes son el servidor de nombres, los facilitadores y los visualizadores. Los primeros funcionan como un servicio de búsqueda de direcciones de agentes y los segundos como un servicio de búsqueda de habilidades de agentes (como unas páginas amarillas). Los agentes visualizadores permiten ver y depurar sociedades de agentes ZEUS.

Un aspecto interesante de la plataforma ZEUS es que propone su propia metodología basada en cuatro etapas⁴⁹: análisis del dominio, diseño de agentes, instanciación de agentes y soporte en tiempo de ejecución. La herramienta de desarrollo soporta estas dos últimas etapas. Además, también es conveniente remarcar el uso de FIPA-ACL como lenguaje de comunicación entre agentes.

49 En (Collis y Ndumu, 1999) se puede encontrar una descripción más detallada de cada una de estas etapas.

5.3.4 JADE

JADE (*Java Agent DEvelopment framework*) (JADE web, 2007) es un entorno para el desarrollo de aplicaciones basadas en agentes conforme a las especificaciones FIPA y totalmente desarrollado en Java. Puede ser considerado un *middleware* que implementa un marco para el desarrollo de sistemas multiagente que garantiza su interoperabilidad.

Se compone de (Bellifemine *et al.*, 2001) una plataforma de agentes acorde con las especificaciones FIPA, una biblioteca de clases Java que implementan aspectos básicos de los agentes, y un conjunto de herramientas gráficas que permiten administrar y monitorizar la actividad de los agentes.

- La *plataforma de agentes JADE* posee ciertos agentes de utilidad que facilitan la gestión de la sociedad de agentes: el AMS (*Agent Management Service*) que gestiona los agentes y el DF (*Ditectory Facilitator*) que ofrece un servicio de páginas amarillas. Existen, además, agentes que permiten monitorizar otros aspectos del sistema (estado de los agentes, intercambio de mensajes, etc.).
- Las *librerías Java* permiten la construcción de agentes. Poseen clases, clases abstractas e interfaces que facilitan la construcción de los agentes, sus componentes, comportamientos, mensajes ACL, ontologías, protocolos de comunicación, etc.
- Con las distintas *herramientas gráficas* se puede, por ejemplo, controlar el ciclo de vida de la plataforma de agentes y de todos los agentes registrados, que los usuarios interactúen con los agentes JADE de forma personalizada, visualizar la interfaz gráfica del agente DF que se generó al iniciar la plataforma, comprobar cómo se produce la interacción e intercambio de mensajes entre los agentes, supervisar la cola de comportamientos de los agentes, etc.

Actualmente JADE se ha convertido en un referente para la programación de sistemas multiagente (Pavón y Gómez, 2005) debido, sobre todo, a su estricta conformidad con el estándar FIPA. Además, posee otras cualidades como son: flexibilidad, buena gestión de la plataforma y del intercambio de mensajes, extensibilidad del código, facilidad para desarrollar aplicaciones distribuidas en diferentes máquinas, abundante documentación, etc.

En el lado negativo, JADE puede decepcionar a aquellos investigadores del campo de la Inteligencia Artificial, dado que carece de mecanismos para

dotar a los agentes de “inteligencia”, capacidad de planificación o razonamiento. Sin embargo, al tener una base Java JADE puede interactuar de un modo relativamente fácil con implementaciones Java de Prolog o de sistemas expertos.

5.3.5 Discusión

Valorar las plataformas existentes es una labor bastante tediosa y complicada. Afortunadamente, existen estados del arte, como el presentado por Garneau y Delisle (Garneau y Delisle, 2002), que han realizado esta tarea. Estos autores consideran 15 criterios de clasificación a los que asignan una puntuación de 0 a 4, de forma que la valoración global de las plataformas analizadas se obtiene sumando la puntuación obtenida en cada criterio. Dados los objetivos del presente trabajo, sólo nos interesa evaluar las plataformas desde un punto de vista técnico, por lo que algunos de esos 15 criterios resultan “ineficaces”. Por ello, en nuestro análisis únicamente consideraremos:

- *Flexibilidad*: entendida como la polivalencia de los componentes y metodologías de la plataforma para poder ser usada en problemas de diferente índole.
- *Comunicación entre agentes*: utilidades para establecer la comunicación entre agentes. El programador no debe ocuparse de aspectos como la conexión entre máquinas, protocolos de comunicación, sincronización o seguridad.
- *Utilidades de depuración*: la plataforma debe poseer herramientas para detectar errores de coordinación o sincronización entre agentes.
- *Gestión del MAS*: deben existir utilidades que permitan al usuario del sistema influir sobre el ciclo de vida de los agentes en tiempo de ejecución. Se podrán añadir agentes, suspenderlos o eliminarlos.
- *Extensibilidad del código*: facilidad de modificación de los componentes predeterminados o del código generado.
- *Despliegue*: valora la posibilidad de ejecutar el sistema resultante en un entorno distribuido y heterogéneo donde los agentes operan en diferentes máquinas.
- *Documentación disponible*: calidad de la documentación existente.

La tabla 5.3 muestra la evaluación de las diferentes plataformas de desarrollo MAS descritas en este apartado, considerando el esquema y los datos presentados por Garneau y Delisle.

<i>Criterios</i>	<i>Plataformas multiagente</i>			
	<i>AgentTool</i>	<i>AgentBuilder</i>	<i>ZEUS</i>	<i>JADE</i>
Flexibilidad	0	1	1	3
Comunicación entre agentes	2	4	4	4
Utilidades de depuración	2	4	4	3
Gestión del MAS	1	3	3	4
Extensibilidad del código	0	1	2	4
Despliegue	1	2	2	4
Documentación disponible	1	4	4	3
TOTAL	7	19	20	25

Tabla 5.3. Evaluación de las plataformas de desarrollo en base a criterios técnicos. Adaptado de (Garneau y Delisle, 2002).

Se observa, por tanto, que Jade es la herramienta que presenta mejores cualidades técnicas, que son las que buscábamos, obteniendo una puntuación bastante superior a la de sus rivales. En consecuencia, desde este punto de vista preferimos Jade. Por otro lado, esta plataforma está teniendo una gran aceptación entre los programadores de sistemas multiagente debido, fundamentalmente, a una completa adecuación a las especificaciones FIPA y a la calidad de las soluciones que ofrece ante aspectos fundamentales en la programación multiagente.

Resumiendo, los agentes Jade tienen todas las características que necesitamos (y alguna más): la comunicación entre “agentes” que corren en diferentes *hosts* de una red es trivial, la programación es eficiente y tolera los fallos, sigue los estándares FIPA, etc. Además, es software libre, posee una gran experiencia en la implementación de grandes sistemas de agentes, el grupo de usuarios es muy activo, los implementadores normalmente responden a los problemas en un plazo breve, etc.

Existe, además, una última razón para adoptar Jade. En el departamento de Organización de Empresas y Comercialización e Investigación de Mercados de la Universidad de Valladolid, al cual pertenezco, se ha desarrollado un trabajo doctoral que aborda la problemática asociada al control de planta desde la perspectiva multiagente, y en la que se ha empleado Jade como herramienta de desarrollo. Es propósito de los autores unir ambos trabajos en un futuro próximo, por lo que resulta aconsejable compartir la misma plataforma de desarrollo.

5.4 METODOLOGÍAS DE DESARROLLO MULTIAGENTE

La construcción de sistemas multiagente integra tecnologías de distintas áreas de conocimiento (Gómez, 2003): técnicas de *ingeniería del software* para estructurar el proceso de desarrollo; técnicas de *inteligencia artificial* para dotar a los programas con la capacidad de afrontar situaciones imprevistas y tomar decisiones, y *programación concurrente y distribuida* para tratar la coordinación de tareas ejecutadas en diferentes máquinas bajo diferentes políticas de planificación. Esta combinación de tecnologías complica notablemente el desarrollo de los sistemas multiagente.

Como ya hemos visto, existen plataformas de desarrollo que dan soluciones parciales al modelado del comportamiento y a la coordinación de los agentes. Sin embargo, y a pesar de que facilitan el proceso de creación de sistemas multiagente, las plataformas de desarrollo quedan incompletas sin un proceso de desarrollo del software específico para agentes que asimile la creación de sistemas multiagente a la producción de software convencional (Gómez, 2003).

Por tanto, además de disponer de estándares que homogeneicen las soluciones que los desarrolladores de software dan a los problemas y de plataformas que permitan desarrollar software adaptado a este paradigma, la construcción de sistemas reales requiere tener en cuenta otros aspectos como es el uso de metodologías que asistan en todas las fases del ciclo de vida de los sistemas multiagente. Estas técnicas proporcionan medios para construir soluciones MAS de forma disciplinada y son más necesarias cuanto mayor es la complejidad del sistema que se aborde.

El objetivo básico de toda metodología es la definición de un nivel conceptual que permita un análisis de los sistemas multiagente. Este objetivo es independiente de la teoría, arquitectura o lenguaje utilizado en la implementación (Iglesias, 2000). Por lo que se refiere al nivel conceptual, éste debe describir los *modelos de agentes* (características de cada agente, sensores, etc.) y los *modelos de la sociedad* de agentes (relaciones estructurales y relaciones entre agentes).

Existe una gran variedad de metodologías que han sido analizadas con profusión en multitud de trabajos (Ricordel y Demanzeau, 2001; Iglesias, 2000; Wooldridge y Ciancarini, 2001; Jennings, 2000; Shehory y Sturm, 2001; Cernuzzi y Rossi, 2001; Gómez, 2002 y 2003; Julián y Botti, 2003;

Tveit, 2001; Gómez-Sanz, 2000; Pavón y Gómez, 2005; etc.). Algunas se han elaborado extendiendo las técnicas de orientación a objetos, como AUML, o de ingeniería del conocimiento, como MASCommonKADS. Otras, como GAIA o las técnicas de modelado BDI, se han propuesto directamente desde el paradigma multiagente, aunque también heredan las ideas y la notación de metodologías anteriores.

Algunas de estas metodologías, como ZEUS o MASCommonKADS, están soportadas por herramientas visuales que facilitan notablemente el proceso completo de elaboración del sistema multiagente (desde el análisis a la implementación). Otras, como GAIA, han optado por quedarse en un nivel de abstracción bastante alto con el fin de desacoplar el análisis y el diseño de la implementación, dotando al desarrollador de una mayor libertad.

Dado el objetivo del presente trabajo no parece lógico describir con detenimiento todas y cada una de las metodologías existentes. No obstante, sí resulta útil comentar de forma breve aquellas metodologías que se consideran más interesantes y relevantes.

5.4.1 AUML

Desde la aparición de la tecnología multiagente se ha aceptado su estrecha relación con la programación orientada a objetos concurrente⁵⁰. De hecho, un agente puede ser considerado como un objeto activo que goza de autonomía y posee un estado mental del que depende la respuesta a los estímulos (Ferber, 1999).

UML (Jacobson *et al.*, 2000) es un lenguaje concebido para realizar una representación gráfica que estandarice el diseño de objetos y sus relaciones. Este lenguaje es, de hecho, un sistema de diseño de software orientado a objetos. Dado que los agentes poseen ciertas particularidades que no poseen los objetos, UML es insuficiente para el modelado de agentes. Esta es la razón por la que se ha propuesto AUML (Odell *et al.*, 2000; Odell *et al.*, 2001; Bauer *et al.*, 2001) como una extensión de UML para la tecnología de agentes.

50 Numerosos autores ven el paradigma de los sistemas multiagente como una evolución de los sistemas orientados a objetos (Molina *et al.*, 2004; Flores-Mendez, 2000; Shoham, 1997).

Aunque AUML es un lenguaje de modelado que aborda todos los aspectos del diseño de sistemas multiagente, su contribución más aplaudida ha sido la representación de los protocolos de comunicación entre agentes. Numerosos investigadores han optado por esta notación para formular sus protocolos, al considerar que la sencillez de AUML conduce a una mejor comprensión de su trabajo. De hecho, el estándar FIPA hace uso de este lenguaje para representar sus protocolos.

Otras aportaciones importantes de AUML para extender UML son:

- *Roles de agentes*: pueden ser considerados como un conjunto de agentes que satisfacen una interfaz, descripción de servicios o comportamientos concretos. Un agente puede, por tanto, satisfacer diferentes roles.
- *Diagramas de protocolos*: representan una interacción, es decir, un conjunto de mensajes intercambiados entre los roles.
- *Línea de vida de los agentes*: define el periodo de tiempo durante el que existe el agente, indicando los eventos de su existencia como la creación, migración, suspensión o destrucción.
- *Hilos de interacción*: a diferencia de los objetos, cuando un agente recibe un mensaje puede optar por ejecutar una tarea determinada o no hacer nada. Las diferentes opciones con las que un agente puede actuar ante un mensaje se representan mediante estos hilos de interacción.
- *Mensajes*: los agentes pueden enviar e intercambiar información por medio de mensajes, aunque dicho envío puede realizarse de diferentes formas. AUML propone una notación basada en flechas para simbolizar el intercambio de información.

5.4.2 MAS-CommonKADS

MAS-CommonKADS (Iglesias, 1998; Iglesias *et al.*, 1999) es una metodología que extiende los modelos definidos en CommonKADS (Tansley y Hayball, 1993; Schreiber *et al.*, 2000) añadiendo técnicas de las metodologías orientadas a objetos y metodologías de diseño de protocolos.

CommonKADS fue diseñada para el desarrollo de sistemas expertos que interactuasen con el usuario. Propone la construcción de siete modelos interrelacionados que permiten la definición del sistema y de su entorno.

Estos modelos, descritos profusamente por Iglesias (Iglesias, 1998), son los siguientes: modelo de organización, tarea, agente, comunicación, modelo de experiencia y modelo de diseño. La metodología gira alrededor del modelo de experiencia⁵¹, descuidando el desarrollo del resto de modelos, y en especial los modelos de agente, comunicación y organización.

CommonKADS no fue diseñada para desarrollar MAS, por lo que al aplicar directamente esta metodología al desarrollo de tales sistemas afloraron algunos problemas: la concepción de agente es muy restrictiva y considera que existen únicamente dos agentes (el usuario y el sistema experto); el modelo de comunicación trata principalmente las interacciones hombre-máquina por lo que no es muy apropiado para abordar las interacciones entre software; la asignación de tareas se realiza de forma fija, lo que no se ajusta a la filosofía multiagente; etc.

En un intento por subsanar dichos problemas surgió MASCommonKADS (Iglesias, 1998). Esta metodología toma como marco de referencia los modelos identificados en CommonKADS e incorpora ciertas técnicas de modelado que facilitan su aplicación y que cubren los aspectos relevantes de un sistema multiagente. En concreto, realiza las siguientes extensiones:

- Definición de un ciclo de desarrollo de una aplicación empleando una perspectiva de agente⁵².
- Definición de una fase de conceptualización para identificar los agentes del problema y sus relaciones básicas. Para este fin, se define una técnica de casos de uso.
- Definición de un modelo de agente inteligente que extiende el modelo homónimo de CommonKADS.
- Definición de un modelo de organización que extiende el modelo homónimo de CommonKADS, permitiendo la especificación de las relaciones en la sociedad multiagente.
- Definición de un modelo de coordinación que permite describir las interacciones entre agentes.
- Extensión del modelo de experiencia en aquellos aspectos específicos de sistemas multiagente.

51 Modela el conocimiento de resolución de problemas y distingue entre el conocimiento de la aplicación y el conocimiento necesario para la resolución del problema.

52 Esta metodología fue la primera en hacer un planteamiento MAS integrado con un ciclo de vida de software, concretamente el espiral dirigido por riesgos (Pressman, 1982).

- Definición de un modelo de diseño que contempla la selección de la plataforma multiagente, descomposición en módulos de los requisitos de los modelos previamente desarrollados, selección de la arquitectura más adecuada para cada agente y determinación de la arquitectura y los servicios ofrecidos por la sociedad multiagente.

Esta metodología, aunque está bien valorada por los investigadores, presenta un inconveniente crucial: el nivel de detalle alcanzado en la descripción no es realizable sin el apoyo de herramientas de soporte (Gómez, 2003). A fin de cuentas, propone identificar una lista detallada de elementos y relaciones en el sistema, lo que puede desembocar en un proceso demasiado costoso, y dado a errores, si la documentación se genera manualmente.

5.4.3 Técnica de modelado de agentes BDI

Los modelos de agentes BDI (*Belief, Desire, Intention*) están basados en la teoría cognitiva del ser humano de Bratman (Bratman, 1987). Según esta teoría, los agentes poseen sensores a través de los que reciben estímulos del entorno que les permiten elaborar creencias de cómo es el mundo, es decir, les permiten definir un modelo del entorno que irá evolucionando a medida que posean nuevas percepciones. Para guiar sus acciones los agentes tienen deseos, o estados que quieren alcanzar, y que lograrán satisfacer a través de unas acciones especiales denominadas intenciones, que podrán abortarse en función de la evolución del entorno.

Aunque la formulación inicial es de Bratman, fueron Georgeff, Rao y Kinny (Georgeff y Rao, 1991; Rao y Georgeff, 1995; Kinny y Georgeff, 1997; etc.) quienes formalizaron este modelo y le dieron visos de metodología (de hecho, es su trabajo lo que aquí se comenta). Para especificar el sistema de agentes en esta metodología se emplean varios modelos que operan a dos niveles de abstracción (Georgeff *et al.*, 1999), el externo y el interno.

Desde un punto de vista externo, un sistema se modela como una jerarquía de herencia de clases de agentes, de la que los agentes individuales son instancias. Las clases de agente se caracterizan por su propósito, sus responsabilidades, los servicios que ejecutan, la información acerca del mundo que necesitan y las interacciones externas (Mas, 2005). Para describir los tipos de agentes que aparecen en el sistema y sus interacciones se emplean el *modelo de agente* y el *modelo de interacciones*.

Desde un punto de vista interno, la tecnología de modelado BDI emplea un conjunto de modelos que permiten imponer una estructura sobre el estado de información y motivación de los agentes, así como las estructuras de control que determinan su comportamiento (Gómez, 2003). Este conjunto de modelos, conocidos como modelos internos, está constituido por los *modelos de creencias, objetivos y planificación*.

Un inconveniente importante de esta metodología consiste en que la generación de los modelos que propone es bastante complicada; se trata de modelos que están estrechamente relacionados entre sí, lo que obliga a seguir un proceso de análisis iterativo. Además, carece de herramientas de soporte, lo que explica su escasa utilización desde el punto de vista práctico. Aún así, es una referencia obligada en todos los trabajos sobre metodologías orientadas a agentes.

5.4.4 GAIA

GAIA (Wooldridge *et al.*, 2000) es una metodología para el diseño de sistemas basados en agentes que busca especificar cómo debe colaborar una sociedad de agentes para alcanzar los objetivos del sistema, y qué se requiere de cada uno de ellos para lograrlos. Pretende ayudar al analista a ir sistemáticamente desde unos requisitos iniciales a un diseño que esté lo suficientemente detallado como para ser implementado directamente.

Esta metodología diferencia dos fases (representadas en la figura 5.1): la fase de análisis y la fase de diseño.

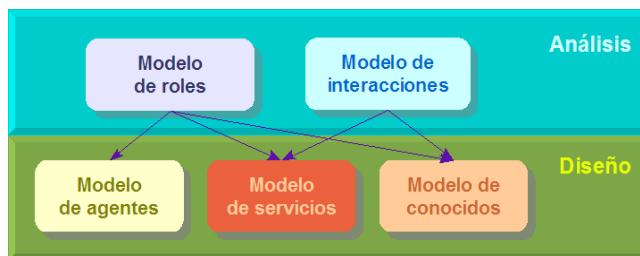


Figura 5.1. Relación entre los modelos GAIA de las fases de análisis y diseño (Wooldridge *et al.*, 2000).

Por lo que se refiere al análisis, en GAIA se entiende que su objetivo es conseguir comprender el sistema y su estructura sin referenciar ningún aspecto de la implementación. Por ello, en esta primera fase de análisis se definen el modelo de roles, en el que se identifican los roles y sus propiedades distintivas (responsabilidades, los recursos que se le permite utilizar y las tareas asociadas), y el modelo de interacciones, que define las interacciones mediante un modelo de intercambio de mensajes.

La segunda etapa, que toma como datos de partida los modelos de roles e interacciones, es el diseño. Su objetivo es generar tres modelos (Gómez, 2002): el *modelo de agentes* que define los tipos de agente que existen y sus instancias, el *modelo de servicios* que identifica los servicios (funciones del agente) asociados a cada rol, y un *modelo de conocidos*, que define los enlaces de comunicaciones que existen entre los agentes. A partir de aquí, los autores proponen aplicar técnicas clásicas de diseño orientado a objetos.

El principal inconveniente que presentaba la primera versión de GAIA consistía en que no se adaptaba totalmente al análisis y diseño de los sistemas abiertos (Araújo, 2003), entre los que se encuentra Internet: su estructura organizacional estática en la que las funciones de los agentes están perfectamente definidas choca de frente con la estructura dinámica propia de los sistemas abiertos, en los que existe la posibilidad de que aparezcan agentes auto-interesados en cualquier instante.

Dada la importancia que tiene Internet en el mundo de los agentes, estos autores decidieron adaptar GAIA al diseño de aplicaciones de Internet. De este modo, en (Zambonelly *et al.*, 2000) propusieron una extensión de GAIA que, entre otros aspectos, añadía la definición de “leyes sociales” en la fase de análisis y sustituía el modelo de conocidos de la fase de diseño por la definición del “comportamiento del medio de coordinación”.

5.4.5 MESSAGE e INGENIAS

Una de las metodologías de desarrollo de sistemas multiagente más recientes es MESSAGE (Evans, 2001; Caire *et al.*, 2002), *Methodology for Engineering Systems of Software Agents*. Esta metodología nació, en palabras de sus autores, con la intención de “integrar los resultados de las metodologías anteriores”.

Propone el análisis y diseño desde cinco puntos de vista: el de *organización*, que captura la estructura global del sistema; el de *tareas/objetivos*, que determina qué hace el MAS y sus agentes constituyentes en términos de los objetivos que persiguen y las tareas implicadas en el proceso; el de *agente*, que contiene una descripción detallada y extensa de cada agente y rol dentro del sistema; el de *dominio*, que actúa como repositorio de la información (para entidades y relaciones) concerniente al dominio del problema; y el de *interacción*, que trata las interacciones a distintos niveles de abstracción.

A pesar de las atractivas intenciones con las que surgió MESSAGE, sus objetivos no se llegaron a ver totalmente completados. Algunos de sus participantes decidieron entonces profundizar en el trabajo realizado dando lugar, como resultado, a la aparición de INGENIAS (INGENIAS web, 2006). Se puede decir que INGENIAS es, fundamentalmente, una evolución de la ideas de la metodología definida en MESSAGE a la que se han incorporado nuevas herramientas de soporte.

La metodología define un conjunto de metamodelos (una descripción de alto nivel de los elementos del modelo) con los que se pretende describir el sistema (Gómez-Sanz, 2001). Los metamodelos señalan lo que se necesita para modelar los agentes, organizaciones, entorno, interacciones, tareas y objetivos. Estos metamodelos se construyen mediante un lenguaje de metamodelado, el GOPRR (*Graph, Object, Property, Relationship, and Role*) (Kelly *et al.*, 1996). La instanciación de estos metamodelos produce diagramas que definen el sistema multiagente.

Sin embargo, el proceso de instanciación no es sencillo ya que existen muchas entidades, relaciones y dependencias entre los distintos modelos. Por eso, INGENIAS define un conjunto de actividades para producir estas instancias que se pueden realizar mediante una herramienta visual de soporte denominada INGENIAS IDE. Esta herramienta almacena la especificación en documentos XML a partir de los cuales se genera tanto código como documentación.

La principal crítica a esta metodología se haya en que el proceso de desarrollo que propone es bastante complejo, por lo que no parece recomendable para sistemas sencillos. En contrapartida, tiene la ventaja de ser muy flexible, además de disponer de la herramienta INGENIAS IDE que soporta el desarrollo completo del sistema multiagente y la generación automática de código.

5.4.6 Metodología para Jade

Hasta hace poco, entre las desventajas importantes de la plataforma Jade destacaba la ausencia de una metodología que guiase el desarrollo de los sistemas multiagente (Garneau y Delisle, 2002). Por fortuna, este problema ha sido subsanado recientemente gracias al trabajo propuesto por M. Nikraz, G. Caire y P.A. Bahride en (Nikraz *et al.*, 2006).

La metodología propuesta se centra principalmente en el proceso (es decir, en la secuencia de fases y pasos que guíen al desarrollador en la construcción del sistema) y en los artefactos que se deben producir (es decir, en los diagramas, esquemas o documentos que representan de forma gráfica o textual uno o más modelos del sistema). También se introduce un borrador de notación para que sea utilizado en la construcción de estos artefactos y se presentan algunas reglas heurísticas, que ayudan al desarrollador a elegir entre las opciones relevantes, y patrones de diseño, que se pueden aplicar para solucionar situaciones comunes.

El proceso descrito cubre la fase de análisis y diseño, tal y como se muestra en la figura 5.2. La fase de análisis es general en naturaleza e independiente de la plataforma adoptada. Por contra, la de diseño asume específicamente que Jade es la plataforma de implementación y se centra directamente en las clases y los conceptos proporcionados por Jade. Observando la figura 5.2 puede apreciarse que no existe un límite estricto entre ambas fases. Por otra parte, la metodología es de naturaleza iterativa, permitiendo así que el diseñador se mueva hacia adelante y hacia atrás entre ambas fases.

Al final de la fase de diseño, el desarrollador debe ser capaz de progresar hacia la implementación, que es donde se produce la codificación real de la solución. Sin embargo, este punto, al igual que el de prueba y despliegue, no son tratados formalmente en la versión actual de la metodología (tal y como se puede apreciar en la figura 5.2), aunque los autores pretenden abordar estos aspectos en un futuro próximo.

Por lo que se refiere a la etapa de planificación, estos autores dejan en manos del diseñador la decisión de utilizar o no sistemas multiagentes en la solución (también suministran algunas referencias a documentos en los que se describen cuándo los MAS son adecuados). Sólo si el diseñador justifica que los MAS son adecuados puede pasar al análisis; en caso contrario, el diseñador debe buscar una solución alternativa.

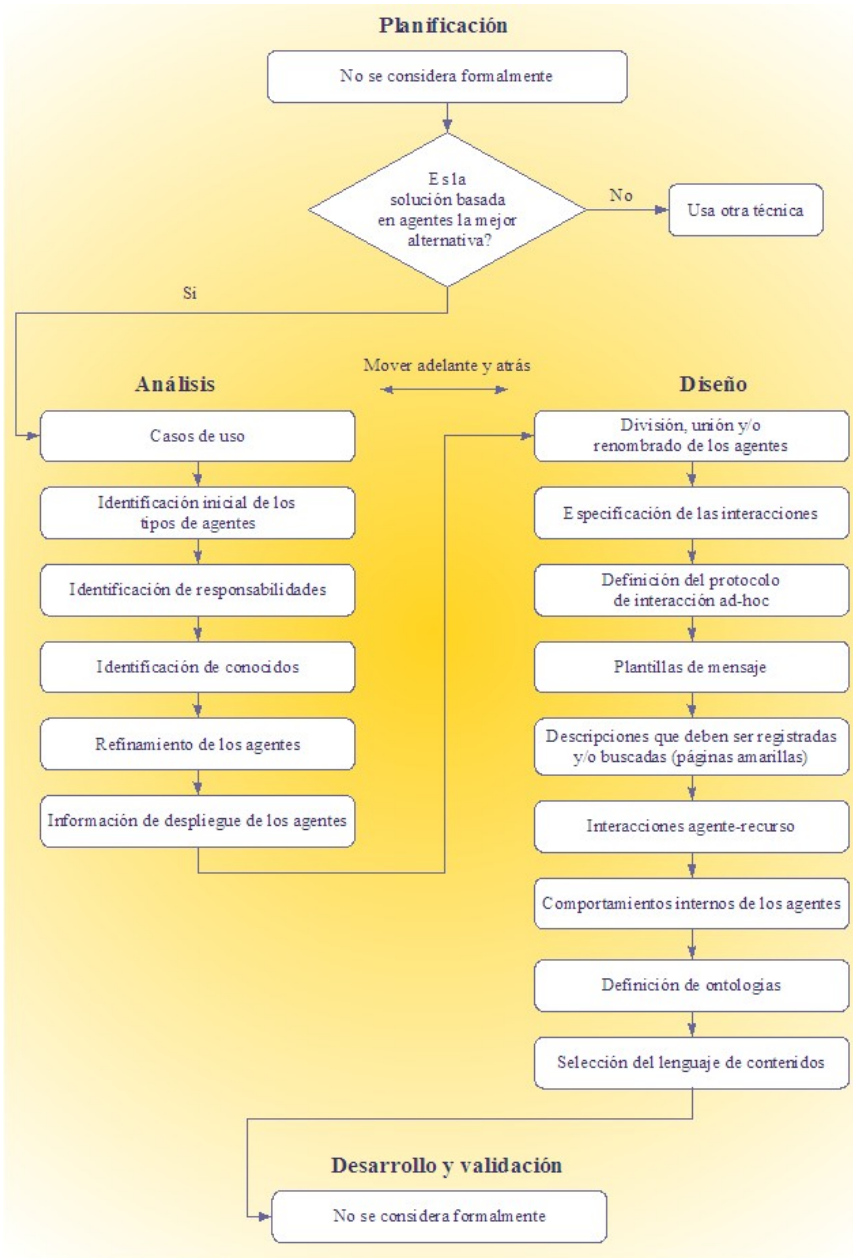


Figura 5.2. Visión general de la metodología Jade (Nikraz *et al.*, 2005).

5.4.7 Discusión

El objetivo de este estudio sobre las metodologías más relevantes no es otro que el de seleccionar la más adecuada para el trabajo que nos ocupa, y para ello es necesario establecer una comparativa entre metodologías. En la literatura sobre el tema existen dos aspectos que se consideran de especial interés para caracterizar las metodologías: la existencia de herramientas de soporte y la procedencia u orientación de los métodos empleados (metodologías extendidas de la ingeniería del conocimiento, metodologías extendidas del paradigma de la orientación a objetos y las desarrolladas específicamente para los MAS).

Si se está acostumbrado a trabajar con sistemas basados en el conocimiento, y el problema se adapta a esta aproximación, lo lógico es utilizar aquellas metodologías que extiendan estas técnicas, como por ejemplo MAS-CommonKADS. Si, por contra, se prefiere hacer uso de un enfoque más orientado a objetos, también se dispone de metodologías basadas en esta perspectiva, como AUML. Finalmente, existe la posibilidad de trabajar con técnicas más cercanas a los nuevos conceptos en los que se basa el paradigma de sistemas multiagente; es el caso de GAIA, ZEUS, INGENIAS o la metodología para Jade.

Otro aspecto importante en la selección de la metodología es la disponibilidad de herramientas de soporte concebidas para facilitar el proceso de análisis y diseño (y, en ocasiones, para generar código automáticamente). Aunque esta característica parece siempre deseable, las metodologías que disponen de ellas suelen ser más complejas y en algunos casos, como el de ZEUS, su uso exige necesariamente la utilización de la herramienta de soporte construida *ad hoc*. Otras metodologías que poseen herramientas de apoyo, además de ZEUS, son MAS-CommonKADS, AUML e INGENIAS.

En este trabajo, se ha preferido optar por una metodología directamente creada para abordar la problemática de los sistemas multiagente, y dado que hemos decidido seleccionar Jade como plataforma de desarrollo, creemos que la mejor metodología posible es la creada específicamente para este entorno. Sin embargo, debemos reseñar que hasta que no apareció dicha metodología nuestra primera elección había sido INGENIAS ya que, además de aunar muchas de las ventajas de otras metodologías, posee una herramienta de apoyo bastante potente (aunque en evolución) que permite la generación de código Jade.

5.5 ONTOLOGÍAS

Cuando dos agentes se comunican lo hacen transfiriéndose información a través de un mensaje. Dentro de cada mensaje, la información está representada como una expresión de contenido coherente con alguno de los múltiples lenguajes de contenido existentes y codificada en un determinado formato. La forma en que los agentes representan internamente una pieza de información debe permitir su fácil manejo; una forma de lograr esto es mediante la definición de ontologías que sean compartidas por todos los agentes involucrados.

5.5.1 Definición

El término ontología tiene su origen en el campo de la filosofía donde, como ciencia, representa la rama de la metafísica que estudia la naturaleza de la existencia, de los seres y de sus propiedades transcendentales. En este sentido, la Real Academia Española de la Lengua la define como “la parte de la metafísica que trata del ser en general y de sus propiedades transcendentales”. Por tanto, y desde este punto de vista, una ontología se ocupa de establecer las categorías fundamentales o modos generales de ser de las cosas.

Derivado de su significado original, aunque con un entendimiento mucho más pragmático y aplicado, en la actualidad el término ontología se usa para referirse a un conjunto de conceptos organizados jerárquicamente, representados en algún sistema informático⁵³, cuya utilidad es la de servir de soporte a diversas aplicaciones que requieren de conocimiento específico sobre la materia que la ontología representa. A éste segundo significado de ontología, entendida como un cuerpo estructurado de conocimiento, es al que queremos circunscribirnos.

En esta línea, “especificación explícita y formal sobre una conceptualización consensuada” (Studer *et al.*, 1998), “conjunto de términos estructurados jerárquicamente que describen un dominio, y que servirán de esqueleto sobre el cual se construirá la base de conocimientos” (Swartout *et al.*, 1997), etc., son sólo algunos de los múltiples ejemplos de definiciones que se pueden hallar en la literatura y que son reflejo del significado de ontología.

53 Por tanto, en el ámbito informático una ontología es una entidad computacional.

Una definición más operativa, aunque acorde con la acepción anterior, es la facilitada por Noy y Mcguinness (Noy y Mcguinness, 2001): “una ontología es una descripción explícita de los conceptos del dominio de discurso, propiedades de cada concepto (que describen varias características y atributos del concepto), y restricciones sobre dichas propiedades”. Dado su carácter práctico, esta es la definición que más se ajusta a las necesidades que plantea la presente tesis.

En cualquier caso, el propósito principal en la elaboración de ontologías no es otro que el de capturar conocimiento para que pueda ser empleado en diferentes sistemas, independientemente de la tarea que pretendan resolver. Para ello se ha de concretar, de forma consensuada y con algún lenguaje formal, un vocabulario (conjunto de términos) común que será utilizado por todos los sistemas que deban manejar conocimiento sobre un dominio determinado.

Sin embargo, las ontologías no son solamente la base para los sistemas basados en conocimiento. A lo largo de los últimos años el ámbito de aplicación de las ontologías ha escapado a este contexto y abarca también otras áreas como son el procesamiento de lenguaje natural, web semántica, comercio electrónico, gestión de conocimientos, etc., y el que nos concierne para nuestro propósito, la comunicación en sistemas multiagente.

En concreto, para un agente una ontología define, básicamente, su entorno. Aunque no es estrictamente necesaria la definición de ontologías para desarrollar sistemas multiagente, su uso es recomendable dado que facilita el intercambio y la reutilización del conocimiento y, además, permite separar el dominio de conocimiento de los aspectos operacionales.

5.5.2 Taxonomía y ejemplos

Dependiendo del punto de vista que se desee considerar se pueden establecer diferentes clasificaciones de ontologías. De entre todas, conviene citar las dos que parecen más interesantes: la que depende del formalismo utilizado y la que es función del grado de reusabilidad. Atendiendo al primer criterio se puede diferenciar entre ontologías ligeras (*Lightweight ontologies*) y pesadas (*Heavyweight Ontologies*). Esta clasificación depende del contenido de axiomas, de manera que a más axiomas más pesadas son (Gómez-Pérez, 1999).

En función del grado de reutilización, Van Heist (Van Heijst *et al.*, 1997) propone cuatro tipos básicos de ontologías. Ordenadas de mayor a menor reusabilidad encontramos las *ontologías de representación*, que proporcionan el vocabulario necesario para modelar otras ontologías; las *ontologías genéricas o meta-ontologías*, que proporcionan términos genéricos reutilizables en diferentes dominios (estado, evento, acción, componente, etc.); las *ontologías de dominio*, que expresan conceptualizaciones específicas para dominios particulares (generalmente se definen como especializaciones de conceptos existentes en las ontologías genéricas); y las *ontologías de aplicación*⁵⁴, que contienen las definiciones necesarias para modelar los conocimientos requeridos por una aplicación particular.

Dentro del ámbito particular que nos atañe, el de las organizaciones, encontramos múltiples proyectos orientados al desarrollo de ontologías que permitan modelar las empresas, tanto públicas como privadas. Entre los ejemplos más representativos de estas ontologías podemos mencionar TOVE, *TOronto Virtual Enterprise* (TOVE web, 2007), ENTERPRISE (ENTERPRISE web, 2007), REA (REA web, 2007) y BMO, *Business Management Ontology* (BMO web, 2007).

5.5.3 Desarrollo de ontologías

Aunque para garantizar la máxima utilidad de las ontologías lo deseable es que sean desarrolladas con suficiente rigor, en la realidad, y dependiendo del propósito para el que sean creadas, son elaboradas con distintos grados de formalismo. Así, se pueden encontrar desde las más informales expresadas en lenguaje natural, hasta las rigurosamente formales expresadas en lenguajes con lógica de primer orden, lógica descriptiva, etc. En los casos más formales las ontologías se concretan identificando términos que se suelen organizar según la siguiente clasificación (Gómez-Pérez *et al.*, 2003):

- *Clase*: conceptos del dominio o categorías, que representan a colecciones de objetos que se identifican como del mismo tipo.
- *Subclase*: clase que representa un subtipo dentro de un tipo (clase padre o superclase). La subclase hereda características de la superclase, aunque posee otras propias.

54 Incluyen conceptos tomados tanto de las ontologías de dominio como de las genéricas.

- *Instancias (casos)*: representan elementos específicos de una clase o concepto.
- *Roles o propiedades (slots)*: características o atributos del concepto (clase). Se especifican indicando los siguientes aspectos: tipo de característica, valor de la característica, restricciones (facetas) o limitaciones de los valores que puede tomar un slot, valor por defecto que toma el slot y, finalmente, la cardinalidad o número de elementos que puede contener un slot individual.
- *Variables*: expresiones que representan un elemento genérico no conocido a priori.
- *Relaciones* (en muchos casos definidas como *predicados*): representan interacciones entre conceptos del dominio. Suelen aparecer las relaciones del tipo: *subclase-de*, *parte-de*, *conectado-a*, etcétera. Pueden ser binarias o *n*-arias dependiendo de los argumentos que tomen.
- *Funciones*: son un tipo especial de relación, en la que para un conjunto de argumentos de entrada se obtiene un único argumento de salida.
- *Axiomas*: son fórmulas, en la mayoría de los casos expresadas en lógicas de primer orden, que se usan para modelar sentencias que son siempre verdad.

La definición de ontologías es un tema ampliamente tratado en la literatura de investigación. De hecho, se pueden encontrar diferentes trabajos (Fernández, 1999; Beck y Pinto, 2002) en los que discute el estado del arte de las metodologías destinadas a este propósito. Dado su carácter práctico, en este trabajo se seguirán las recomendaciones de Noy y McGuinness (Noy y McGuinness, 2001) que proponen un método muy sencillo e intuitivo, que además está soportado por la herramienta Protégé (Protégé web, 2006).

En ese documento, además de tratar los aspectos generales a tener en cuenta, proponen un procedimiento de desarrollo que sigue un enfoque iterativo: en una primera pasada se define la ontología de forma imprecisa, y en las sucesivas iteraciones⁵⁵ se revisa, refina y completa. Esta evaluación y depuración se da a lo largo de todo el ciclo de vida de la ontología y puede realizarse usándola en aplicaciones o métodos que resuelvan problemas o discutiéndola con expertos en el área, o bien de ambas formas a la vez.

55 Por tanto, consideramos las ontologías como entes en continua evolución.

Teniendo presente que una ontología es un modelo de la realidad del mundo y que los conceptos embebidos en ella deben reflejar dicha realidad, estos autores proponen seguir siete pasos: determinar el dominio de la ontología y su alcance, considerar la posibilidad de reutilización de otras ontologías, enumerar los términos importantes, definir las clases y su jerarquía, establecer las propiedades o *slots* de las clases, definir las restricciones de los *slots* y, finalmente, crear las instancias de las clases.

5.6 SISTEMAS EXPERTOS

La elección del sistema experto es una tarea complicada que está profundamente influenciada por nuestras necesidades (reflejadas en el capítulo previo) y por las elecciones efectuadas en los apartados anteriores.

5.6.1 Desarrollo de un sistema experto

Durante muchos años el desarrollo de sistemas expertos fue una tarea *ad hoc* del problema a resolver. No fue hasta 1983 cuando Weiss y Kulikowski sugirieron (Hayes-Roth *et al.*, 1983) una serie de etapas que concretarían un año más tarde (Weiss y Kulikowski, 1984). En concreto, estos autores sugieren que el diseño y construcción de un sistema experto deben evolucionar a través de las diferentes fases que aparecen representadas en la figura 5.3.

La primera de las etapas sugeridas es la del *planteamiento del problema*. Como en todo proyecto lo primero es definir clara y específicamente el problema a resolver⁵⁶ para evitar todo un desarrollo en vano. Si no se hace una buena definición del problema lo más probable es que el sistema funcione incorrectamente, por lo que las decisiones que se tomen en base a las conclusiones que arroje el sistema serán equivocadas.

Una vez definido el problema comienza la *búsqueda de los expertos* que puedan resolverlo. Siempre es conveniente buscar y seleccionar aquellos que posean la mayor experiencia en la resolución del problema definido. También es posible llevar a cabo, de forma previa, una búsqueda de la bibliografía

56 Puesto que el objetivo principal de un sistema experto es responder a preguntas y resolver problemas, esta etapa es quizás la más importante.

especializada sobre el problema que sirva de base para comenzar la carga de conocimientos en el sistema. Se debe tener en cuenta que los conocimientos deben formalizarse antes de ser incorporados a la base de conocimiento, y que la información recogida en un libro suele ser más fácil de estructurar y formalizar que los conocimientos de un experto humano.

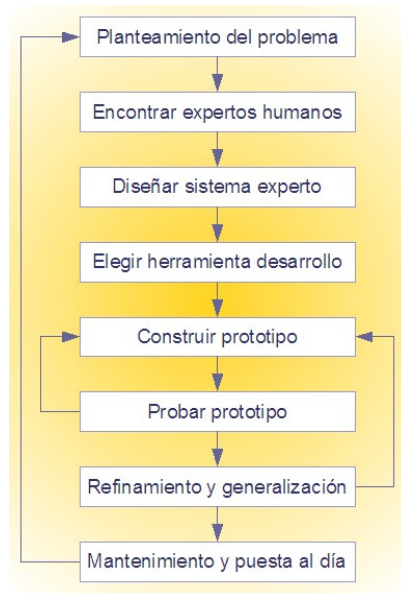


Figura 5.3. Etapas en el desarrollo de un sistema experto (Weiss y Kulikowski, 1984).

La tercera etapa es el *diseño del sistema experto*. En este punto se comienza con el diseño de los componentes del sistema: motor de inferencias, estructura de la base de conocimientos, el subsistema de explicación, la interfaz de usuario, etc.

Con el diseño del sistema terminado se debe realizar la *elección de la herramienta de desarrollo, shell⁵⁷ o lenguaje de programación* para la construcción del sistema. Profundizaremos sobre este tema a lo largo de los siguientes apartados.

57 Un shell es, en pocas palabras, un sistema experto vacío (posee la base de conocimiento vacía). Esto permite que el desarrollador sólo tenga que centrarse en la incorporación de información en la base de conocimiento y no en propio desarrollo del sistema.

La quinta etapa consiste en el *desarrollo y prueba de un prototipo*. Si el prototipo no se ajusta a los requisitos previos, se deben repetir las etapas anteriores (con las modificaciones oportunas) hasta obtener un prototipo apropiado para la resolución del problema. Tras aceptar el prototipo llega la etapa de *refinamiento y generalización* en la que se corrigen los fallos y se incluyen nuevas posibilidades no contempladas en el diseño inicial.

Falta, por último, realizar el *mantenimiento y puesta al día* del sistema experto. Con la ayuda del usuario se plantean nuevos problemas o se detectan defectos del prototipo, se corrigen errores, se actualiza el producto con los nuevos avances, etc. Esta etapa exige la repetición de las anteriores ya que permite descubrir los cambios que precisan cada una de ellas para obtener las mejores soluciones.

Todas estas etapas influyen en la calidad del sistema experto resultante, que siempre deberá ser evaluado y mejorado con las aportaciones de los usuarios.

5.6.2 Herramientas de desarrollo de sistemas expertos

Históricamente, los primeros sistemas basados en conocimiento fueron desarrollados utilizando lenguajes de programación como LISP y PROLOG. A medida que el desarrollo de los sistemas expertos iba aumentando en cantidad y complejidad, la comunidad científica comenzó a buscar nuevas formas de desarrollar los sistemas en menos tiempo y con un menor esfuerzo.

Esto dio lugar a la aparición de sistemas vacíos, también conocidos como *shells*, que ofrecen toda la arquitectura de un sistema experto y a los que sólo hay que incorporar la base de conocimientos. Posteriormente irrumpieron con fuerza en el mercado otras herramientas que incorporaban, además de opciones de representación del conocimiento, esquemas de inferencia y control. A estas herramientas se las conoce como entornos de desarrollo de sistemas expertos o entornos híbridos de desarrollo.

TIPOS DE HERRAMIENTA

Para el desarrollo de un sistema experto se han utilizado diversos tipos de “herramienta”, que se pueden clasificar en 6 categorías, que de menor a mayor grado de desarrollo son:

- Lenguajes tradicionales orientados a procesos numéricos como son Pascal, C, FORTRAN, COBOL, etc.
- Lenguajes orientados a objetos (Smalltalk, Eiffel, C++, Java, etc.).
- Lenguajes orientados a manipulación simbólica (lenguaje funcional) como el LISP.
- Lenguajes de “programación lógica” como son PROLOG, CHIP, u OPS5.
- Herramientas dedicadas para sistemas expertos, los conocidos como “Expert System Shell” (EMYCIN, Crystal, Leonardo, XiPlus, EXSYS, VP-Expert, Intelligence Compiler, etc.).
- Entornos híbridos de desarrollo (*CLIPS*, *Jess*, KEE, ART, EGERIA, Kappa, Nexpert Object, Goldworks, LOOPS, Flavors, etc.).

En general, se recomienda utilizar herramientas dedicadas para el desarrollo de sistemas expertos, a no ser que tengamos un objetivo especial o académico (como, por ejemplo, investigar otras estructuras de datos, mecanismos de inferencia o tratamientos de la incertidumbre) que las herramientas comerciales no ofrezcan o no permitan explorar con tanta libertad como si las implementáramos nosotros mismos.

CRITERIOS DE SELECCIÓN DE LA HERRAMIENTA

Los principales factores que se deben considerar para seleccionar una herramienta de desarrollo que permita implementar un sistema experto son las siguientes (Stylianou *et al.*, 1992):

- Que tenga suficiente potencia en cuanto a la capacidad de gestión de la base de conocimientos y con una buena velocidad de respuesta.
- Que sea flexible para que el desarrollo se pueda realizar ágilmente.
- Que las estructuras que usa para representar el conocimiento sean adecuadas para el tipo de problema que deseamos resolver.
- Que los métodos de razonamiento que utiliza también sean adecuados al tipo de problema al que nos enfrentamos.
- Que las interfaces que ofrece al desarrollador y al usuario sean de fácil manejo y contribuyan a facilitar tanto el desarrollo como la explotación, en vez de dificultarlos.
- Que ofrezca los interfaces necesarios con otros sistemas (como, por ejemplo, bases de datos).

- Que el proveedor respalde el producto con entrenamiento, asesoría y soporte.
- Que el software funcione en la plataforma de hardware que tenemos disponible para el desarrollo y la posterior explotación del sistema.
- Que los costos estén dentro del presupuesto destinado al proyecto.

5.6.3 Lenguajes

Tradicionalmente LISP y PROLOG han sido los lenguajes que se han utilizado para la programación de sistemas expertos. Estos lenguajes ofrecen características especialmente diseñadas para manejar los problemas que tradicionalmente se tratan en Inteligencia Artificial. Por este motivo se los conoce como lenguajes de inteligencia artificial.

Una de las principales características que comparten los lenguajes LISP y PROLOG, como consecuencia de su respectiva estructura, es que pueden ser utilizados para escribir programas capaces de examinar a otros programas, incluyéndose a ellos mismos. Esta capacidad se requiere, por ejemplo, para hacer que el programa explique sus conclusiones.

LISP

LISP, cuyo nombre se deriva de *LIS*t *Processor*, fue el primer lenguaje para procesamiento simbólico. John McCarthy lo desarrolló en 1958, en el Instituto de Tecnología de Massachusetts (MIT, *Massachusetts Institute of Technology*), inicialmente como un lenguaje de programación con el que los investigadores pudieran implementar eficientemente programas de computadora capaces de razonar. Rápidamente LISP se hizo popular por su capacidad de manipular símbolos y fue escogido para el desarrollo de muchos sistemas de Inteligencia Artificial.

En la actualidad existen varios dialectos de LISP en el mercado informático, sin embargo Common Lisp y Scheme pueden considerarse como los más importantes. En cuanto a sus aplicaciones, LISP se está utilizando en varios dominios que incluyen los sistemas para diseño mecánico asistido por ordenador (AutoCAD) o las animaciones gráficas.

PROLOG

El lenguaje de programación PROLOG, del francés *PROgrammation avec LOGique*, fue creado por Alain Colmerauer y sus colaboradores a principios de los 70 (Roussel, 1975) en la Universidad de Marseille-Aix, si bien uno de los principales protagonistas de su desarrollo y promoción fue Robert Kowalski (Kowalski, 1974) de la Universidad de Edimburgh⁵⁸.

Prolog es un lenguaje de programación que se utiliza para resolver problemas en los que existen objetos y relaciones entre objetos. La programación en Prolog consiste en: declarar hechos sobre los objetos y sus relaciones (construir la base de hechos), definir reglas sobre dichos objetos y relaciones (base de conocimiento), y finalmente hacer preguntas. Prolog permite deducir lógicamente las respuestas a las consultas realizadas en base al conocimiento que previamente se ha incorporado

El lenguaje PROLOG juega un importante papel dentro de la Inteligencia Artificial, y se propuso como el lenguaje nativo de las máquinas de la quinta generación (*Fifth Generation Kernel Language*, FGKL), de las que se pretendía que fueran Sistemas de Procesamiento de Conocimiento. La expansión y el uso de este lenguaje propició la normalización del lenguaje Prolog con la norma ISO (propuesta de junio de 1993).

OPSS

Official Production System 5 (OPSS), es un lenguaje para ingeniería cognoscitiva que soporta el método de representación del conocimiento en forma de reglas. Su importancia radica en que fue el primer lenguaje basado en reglas que se utilizó de forma exitosa en un sistema experto.

OPSS es un miembro de la familia de lenguajes de programación desarrollados en la Universidad Carnegie-Mellon⁵⁹. Incorpora un módulo unificador, un intérprete que incluye un mecanismo de encadenamiento progresivo, así como herramientas para la edición y depuración de los programas.

58 Los orígenes de Prolog se detallan ampliamente en un artículo de Colmerauer y Roussel (Colmerauer y Roussel, 1996).

59 Fue desarrollado por Charles Forgy a finales de los 70 (Forgy y McDermott, 1977).

5.6.4 CLIPS

CLIPS, acrónimo de *C Language Integrated Production System* (Sistema de Producción Integrado en Lenguaje C), es una herramienta que suministra un entorno de desarrollo completo para la construcción y ejecución de sistemas expertos. CLIPS es, probablemente, el sistema experto más ampliamente usado debido, entre otras razones, a que es rápido, eficiente y gratuito.

Las primeras versiones de CLIPS se desarrollaron a partir de 1985, en el centro espacial Johnson de la NASA, hasta mediados de la década de los 90. Desde su nacimiento, CLIPS ha sido utilizado por más de 5.000 empresas tanto del sector público como privado, incluyendo la propia NASA, divisiones militares de los Estados Unidos, entidades federales, Universidades y empresas privadas de todo el mundo.

Su autor original, Gary Riley, actualiza y mantiene constantemente esta herramienta. En la actualidad se puede acceder a su última versión estable (la versión 6.2) como un software de dominio público a través de la web de descarga (CLIPS web, 2007), en el que también se encuentra la documentación asociada. Los ejecutables, el código fuente y los documentos que allí se encuentran pueden ser usados libremente y redistribuidos sin restricción alguna.

Las características principales de CLIPS son (CLIPS web, 2007):

- *Representación del conocimiento*: permite manejar una amplia variedad de conocimiento, soportando tres paradigmas de programación: declarativa, imperativa y orientada a objetos. La programación declarativa permite que el conocimiento se represente mediante reglas heurísticas que especifican las acciones a ejecutar en cada situación. La imperativa permite ejecutar algoritmos de la misma manera que en C, Java, etc. Gracias a la OOP se pueden modelar sistemas complejos como componentes modulares.
- *Portabilidad*: CLIPS fue escrito en C con el fin de hacerlo más portable y rápido, y ha sido instalado en diversos sistemas operativos (Windows 95/98/NT, MacOS X, Unix) sin ser necesario modificar su código fuente. CLIPS puede ser ejecutado en cualquier sistema con un compilador ANSI de C, o un compilador de C++. El código fuente de CLIPS puede ser modificado en caso que el usuario lo considere necesario, con el fin de agregar o quitar funcionalidades.

- *Integrabilidad*: CLIPS puede ser embebido en código imperativo, invocado como una subrutina, e integrado con lenguajes como C, Java, FORTRAN y otros. CLIPS incorpora un completo lenguaje orientado a objetos (OOPL) para la elaboración de sistemas expertos. Aunque está escrito en C, su interfaz más próxima se parece a LISP. Pueden escribirse extensiones a CLIPS sobre C, y al contrario, CLIPS puede ser llamado desde C. CLIPS puede ser extendido por el usuario mediante el uso de protocolos definidos.
- *Desarrollo interactivo*: la versión estándar de CLIPS proporciona un entorno de desarrollo interactivo y basado en texto, que incluye herramientas para la depuración, ayuda en línea y un editor integrado. Las interfaces de este entorno tienen menús, editores y ventanas que han sido desarrollados para MacOS, Windows 95/98/NT/XP,... , entre otros.
- *Verificación/validación*: CLIPS contiene funcionalidades que permiten verificar las reglas incluidas en el sistema experto que está siendo desarrollado, incluyendo diseño modular y particionamiento de la base de conocimientos del sistema, chequeo de restricciones estático y dinámico para funciones y algunos tipos de datos, y análisis semántico de reglas para prevenir posibles inconsistencias.
- *Documentación*: en la página web oficial de CLIPS se encuentra una extensa documentación que incluye un manual de referencia y una guía del usuario.
- *Bajo costo*: CLIPS es un software de dominio público.

Otra de las razones que explica el amplio uso de CLIPS está en el gran número de sus derivados e interfaces con otros lenguajes; ejemplos significativos son FuzzCLIPS, que incorpora a CLIPS la posibilidad de usar razonamiento difuso, y JESS, que es una implementación de CLIPS en Java. Precisamente, en el próximo apartado conoceremos algunas de las características más distintivas de esta última herramienta.

5.6.5 JESS

JESS (*Java Expert System Shell*) fue creado a finales de los 90 por Ernest Friedman-Hill de Sandia National Laboratories en Livermore, California. Se trata de un un motor de inferencia y un scripting language⁶⁰ escrito

60 Un lenguaje que indica al ordenador cómo ejecutar un procedimiento específico.

enteramente en lenguaje Java de Sun Microsystem (JESS web, 2007), por lo que soporta el desarrollo de sistemas expertos basados en reglas que después podrán acoplarse perfectamente a cualquier código escrito en lenguaje Java.

Usando Jess, se puede construir software (aplicaciones de línea de comandos, aplicaciones con interfaz de usuario, *servlets* y *applets*) en Java con capacidad para obtener resultados inteligentes, usando conocimiento que se provee en forma de reglas declarativas. Jess es pequeño, ligero, y uno de los motores de reglas más rápidos disponibles; de hecho, para algunos problemas posee un rendimiento que supera a Clips (especialmente si se utiliza un buen compilador *Just In Time*). Además, su lenguaje de *scripting* permite el acceso a todas las API de Java.

Para realizar su cometido, utiliza una versión avanzada del algoritmo RETE para procesar reglas. Este algoritmo RETE mejora la eficacia del emparejamiento de reglas en el motor de inferencias. Jess permite encadenamiento de reglas hacia adelante, emparejando hechos en Memoria Activa con antecedentes, y hacia atrás, emparejando hipótesis en Memoria Activa con consecuentes. Además puede manipular y razonar directamente sobre objetos Java: se pueden crear objetos, llamar a métodos, o ejecutar interfaces sin tener que compilar ningún código Java.

El paso más importante en el desarrollo de aplicaciones Jess es elegir una arquitectura de entre la gama casi ilimitada de posibilidades. Una manera de organizar las posibilidades es enumerarlas en orden creciente de la cantidad de programación Java implicada (Friedman-Hill, 2007).

1. *Scripts* puros en lenguaje Jess. Nada de código Java.
2. *Scripts* puros en lenguaje Jess que acceden a las API Java.
3. Principalmente *scripts* en lenguaje Jess, pero un cierto código Java personalizado en la forma de nuevos comandos Jess escritos en Java.
4. La mitad son *scripts* escritos en lenguaje Jess, con una cantidad significativa de código Java que proporciona comandos personalizados y API; el método *main()* lo proporciona Jess.
5. La mitad de los *scripts* están escritos en lenguaje Jess, con una cantidad significativa de código Java que proporciona comandos personalizados y API; el método *main()* lo escribe el programador.
6. Sobre todo código Java, que carga los *scripts* en lenguaje Jess en tiempo de ejecución.
7. Todo código Java, que manipula Jess a través de su API Java.

En cuanto a sus desventajas, conviene recordar que Jess es un sistema experto basado en reglas, por lo que pertenece a la AI simbólica, es decir, se obtiene un resultado “inteligente”, pero para llegar a él no se realiza ningún tipo de razonamiento humano, simplemente se hacen emparejamientos de reglas preestablecidas. Por otro lado, Jess no es gratuito, aunque está disponible sin ningún coste para uso académico.

5.6.6 Discusión

SOBRE LAS HERRAMIENTAS

Al estudiar las distintas herramientas de desarrollo de sistemas expertos que existen en el mercado se llega fácilmente a una serie de conclusiones con respecto a los *shells* y a los entornos híbridos de desarrollo:

- Poseen la gran ventaja que una vez escogido el más idóneo bastará conocer los conceptos básicos del lenguaje con el que se trabaje para poder completar su base de conocimientos.
- La mayor parte de estas herramientas cuentan con los mecanismos de inferencia de “encadenamiento hacia adelante” y “hacia atrás”.
- Realizar modificaciones o crear nuevas funciones exige un mayor conocimiento del lenguaje y requiere una interfaz de usuario flexible.
- Existen herramientas sencillas que sólo ofrecen ayuda en la representación del conocimiento y otras más complejas que ponen a disposición distintas formas de representación del conocimiento y diferentes estrategias de inferencia.
- Su potencia depende en gran medida del hardware utilizado.
- La mayoría de las herramientas utilizan reglas de producción como herramienta para representar el conocimiento.
- La elección de una herramienta muy avanzada y de gran potencia no garantiza el éxito en la creación de un sistema experto. Es necesario, cumplir de la mejor manera posible las etapas establecidas para el desarrollo de un sistema experto.
- Están sujetas a numerosos controles de calidad (en su mayoría son herramientas comerciales o pseudocomerciales) que los hacen más fiables.
- Se trata, generalmente, de la solución más rápida.

Y SOBRE LA ELECCIÓN MÁS ADECUADA

La elección de la herramienta de desarrollo es, sin duda, una de las etapas más importantes en la creación de un sistema experto. De su elección dependen multitud de factores como son el tiempo de desarrollo, el esfuerzo requerido en su construcción y mantenimiento, la velocidad de respuesta, el coste del proyecto, la relación con el usuario, el experto u otras aplicaciones software, etc. Por tanto, debe ser una decisión bastante bien meditada, sobre todo considerando que tal elección se va a ver condicionada por la amplia variedad de productos existentes en el mercado.

Sin embargo, en el caso que nos ocupa, la dificultad y complejidad asociadas a esta decisión se ven notablemente reducidas si tenemos en cuenta tanto la idiosincrasia de nuestro problema como el conjunto de decisiones que se han tomado previamente en relación a la herramienta de desarrollo del sistema multiagente. Ambos aspectos van a guiarnos claramente hacia una herramienta concreta.

En efecto, dado que sólo necesitamos crear un sistema experto basado en reglas el abanico de posibilidades se ve reducido significativamente. De hecho, de entre todas las herramientas disponibles, CLIPS destaca significativamente sobre todo por ese conjunto de propiedades que son las responsables de que su uso se haya extendido de forma tan notable. Sin embargo, el que esté basado en el lenguaje C constituye un impedimento importante si tenemos en cuenta que la plataforma de desarrollo multiagente que hemos seleccionado, Jade, está basado en el lenguaje Java.

Sin embargo, de nuevo el camino ante nosotros vuelve a despejarse gracias a la existencia en el mercado de una herramienta basada en CLIPS aunque escrita en lenguaje Java; nos estamos refiriendo a Jess. Jess ofrece toda la funcionalidad de Clips, e incluso superior, y soporta el desarrollo de sistemas expertos basados en reglas que después podrán acoplarse perfectamente a cualquier código escrito en lenguaje Java. En definitiva, Jess va a permitirnos integrar fácilmente las dos tecnologías de AI aquí consideradas.

El único inconveniente real de utilizar Jess es que no es una herramienta gratuita, aunque sí está disponible para su uso académico. Por ese motivo, desde estas líneas quiero volver a agradecer a Craig A. Smith (Software Licensing Manager de Sandia National Laboratories) todas las facilidades que nos ha dado para poder utilizarla.

Jess y Jade parecen ser, por tanto, una combinación acertada. De hecho, su idoneidad se ve reforzada si consideramos que a lo largo de los últimos años han empezado a aparecer algunos trabajos en los que se hace una utilización conjunta de ambas herramientas para distintos y variados propósitos: energía eléctrica, juegos, simulación, control del tráfico aéreo, etc.

5.7 RESUMEN

Durante los últimos años la tecnología basada en agentes ha ido concretándose de un modo significativo, apareciendo diferentes propuestas de estandarización, desarrollo e implementación destinadas a posibilitar la construcción de sistemas multiagente más prácticos. En la actualidad, el estándar que parece ser aceptado por un mayor número de desarrolladores es FIPA, ya que recoge todos los aspectos del paradigma multiagente descritos hasta el momento tanto por la comunidad investigadora como por la industria.

Por este motivo, en este trabajo se ha optado por seguir las especificaciones FIPA y, además, utilizar Jade como herramienta de implementación. Jade es un entorno de desarrollo bastante utilizado, no sólo por su estricta conformidad con el estándar, sino también por otras cualidades como son flexibilidad, buena gestión de la plataforma y del intercambio de mensajes, extensibilidad del código, facilidad de depuración y para desarrollar aplicaciones distribuidas en diferentes máquinas, abundante documentación, etc. Además, es software libre y el grupo de usuarios es muy activo.

La adopción de Jade como herramienta de desarrollo también se ha visto condicionada por una última cuestión. En el departamento de Organización de Empresas y Comercialización e Investigación de Mercados de la Universidad de Valladolid, al cual pertenezco, se ha desarrollado recientemente un trabajo que aborda la problemática asociada al control de planta desde la perspectiva multiagente. Es propósito de los autores unir ambos trabajos en un futuro próximo, por lo que resulta aconsejable compartir el mismo entorno de desarrollo.

En definitiva, creemos que la adopción de FIPA y Jade es una solución bastante adecuada dadas las características de nuestro problema y la amplia aceptación tanto del estándar como de la herramienta. Dicha elección, no obstante, condiciona significativamente la selección de la metodología de

desarrollo. Así, si durante bastante tiempo aquellos que optaban por Jade se encontraban como principal problema la falta de una metodología propia que guiase el análisis y el diseño; esta situación ha cambiado desde 2005, momento en el cual Nikraz, Caire y Bahride propusieron una metodología expresa para Jade, que es la que seguiremos aquí.

Otro de los elementos clave en la construcción de sistemas multiagente, sobre todo para lograr un intercambio de información eficiente entre agentes, son las ontologías. Una ontología es simplemente un modelo de la realidad del mundo y para el caso concreto de un agente una ontología define su entorno. En este trabajo hemos optado por seguir la metodología propuesta por Noy y McGuinness en la que se afronta su construcción desde un enfoque iterativo en el que la ontología ha de evaluarse y depurarse a lo largo de todo su ciclo de vida.

Finalmente, hemos abordado la problemática asociada a la construcción del sistema experto, ya que constituirá el módulo de decisión de nuestra plataforma multiagente. En el mercado existen multitud de lenguajes, *shells* y entornos de desarrollo que poder utilizar para construir nuestro sistema. Sin embargo, no todos son adecuados a nuestros propósitos y a las decisiones que se han resumido en los párrafos anteriores. En consecuencia, nos hemos decidido finalmente por Jess, una herramienta con las mismas, e incluso superiores, funcionalidades que CLIPS pero que al estar basada en Java nos permite una fácil integración con la plataforma Jade.

ANÁLISIS Y DISEÑO DE LA PLATAFORMA MULTIAGENTE

6.1 INTRODUCCIÓN

El éxito de la empresa virtual dinámica depende de multitud de factores, pero de entre todos ellos destaca uno por encima de los demás: la correcta selección de los socios. El modelo presentado en el capítulo cuarto pretende ser un punto de partida en la construcción de aplicaciones software que soporten dicha tarea de un modo más realista y eficiente. En consecuencia, si queremos comprobar su validez es preciso desarrollar una plataforma software que lo implemente.

Esa aplicación software ha de permitir una ágil configuración de la red incurriendo, a su vez, en una óptima utilización de los recursos disponibles, tanto humanos como materiales, económicos, etc. Para lograr este propósito nos hemos decantado por el uso de una tecnología que, aunque emergente, está dando excelentes resultados en las diferentes áreas del conocimiento donde se aplica: nos estamos refiriendo a los sistemas multiagente.

Pero lo primero que debemos tener claro cuando se pretende construir una solución de este tipo es que desarrollar un MAS equivale, sencillamente, a desarrollar un sistema software. Por eso, muchos de los aspectos que se han venido aplicando convencionalmente al desarrollo de software se pueden trasladar fácilmente al desarrollo de soluciones basadas en agentes; entre ellos, el concepto de ciclo de vida del desarrollo de software. Otros, sin embargo, deben adaptarse y particularizarse a la tecnología que nos ocupa y, por tanto, al nivel de abstracción que conlleva; este es el caso, por ejemplo, de los estándares o de la metodología de desarrollo.

En cuanto al ciclo de vida del desarrollo software, éste se compone de cinco fases fundamentales (Nikraz *et al.*, 2006): planificación, análisis, diseño, implementación y validación. Para poder progresar de forma adecuada a través de estas etapas a menudo resulta útil disponer de una lista con los pasos importantes a seguir: éste es el cometido principal de una metodología. Una metodología permite ahorrar mucho tiempo y esfuerzo al proporcionar al diseñador una dirección adecuada, una “receta” (Bussmann *et al.*, 2004), que le ayuda a construir la solución especificando algunos de los pasos del proceso, mientras que el resto los deja a la creatividad del diseñador.

Ya conocemos algunas de las metodologías que actualmente existen para el desarrollo de los MAS (Gaia, AUML, INGENIAS,...). La mayoría de ellas intentan adaptar las metodologías de análisis y diseño orientadas a objetos al diseño basado en agentes (Wooldridge, 2002) y, además, siguen un enfoque *top-down* (de arriba hacia abajo). Esta forma de actuar presenta varias desventajas que tienen su origen, principalmente, en el hecho de que los objetos y los agentes proporcionan abstracciones diferentes y, por consiguiente, exigen pensar a distintos niveles (Odell, 2000). Además, asumir enteramente un enfoque *top-down* no es adecuado en aquellos sistemas que contienen recursos que deben utilizarse dentro del MAS.

Resulta más acertado, por tanto, una metodología que no intente extender las técnicas orientadas a objetos y que se centre específicamente en los agentes y en las abstracciones proporcionadas por dicho paradigma. Por otro lado, y puesto que hemos elegido JADE como entorno de desarrollo, lo ideal sería disponer de una “receta” centrada específicamente en dicha plataforma. Sin embargo, la ausencia de una metodología propia que guíase el desarrollo se reconocía, hasta hace poco, como una de las desventajas importantes de esta plataforma (Garneau y Delisle, 2002). Por fortuna, este problema ha sido subsanado recientemente gracias a Nikraz *et al.* (Nikraz *et al.*, 2006).

Esta nueva metodología, además de centrarse específicamente en la plataforma JADE, combina los enfoques *top-down* y *bottom-up* (de abajo a arriba) de modo que pueden tenerse en cuenta tanto las capacidades existentes del sistema como las necesidades generales de las aplicaciones (estas últimas basadas en los requisitos). Sin embargo, se ocupa casi exclusivamente de las fases de análisis y diseño: la etapa de planificación, donde se decide qué herramienta usar y se valora si una opción basada en agentes es la opción más apropiada, no se cubre formalmente, al igual que la de implementación y validación, que se dejan para un futuro próximo.

Con respecto a la fase de análisis (orientada a clarificar el problema identificando los requisitos del dominio), ésta es de naturaleza genérica e independiente de la plataforma adoptada. Por contra, la fase del diseño (cuyo propósito consiste en especificar la solución del problema) asume específicamente que JADE es la plataforma de implementación y se centra directamente en las clases y los conceptos proporcionados por dicho entorno. Además, la metodología es de naturaleza iterativa, permitiendo que el diseñador se mueva hacia adelante y hacia atrás entre ambas fases, lo que explica que no exista un límite estricto entre ambas fases.

Análisis y diseño de la plataforma constituyen los pasos siguientes en la evolución natural de nuestro trabajo. En efecto, en los capítulos previos hemos conocido la importancia trascendental que posee la correcta selección de los socios en el éxito de la DVE, lo que nos ha conducido a definir un modelo realista y multienfoque. Pero, además, también hemos realizado la planificación de la plataforma que nos permitirá probar las prestaciones del modelo: en primer lugar, valorando y justificando el paradigma de agentes y los sistemas expertos como las tecnologías más adecuadas para la construcción de la plataforma, y en segundo lugar, seleccionando los estándares, metodologías y herramientas más próximas a nuestros propósitos.

Atendiendo a estas consideraciones, el presente capítulo se ha estructurado en dos grandes bloques orientados, respectivamente, hacia el análisis (sección 6.3) y el diseño (sección 6.4) de la plataforma. Partiendo de una breve descripción de la funcionalidad que ha de poseer la plataforma (sección 6.2), a lo largo de las secciones siguientes se exponen los aspectos más interesantes del trabajo desarrollado atendiendo a las etapas contempladas en la metodología. La sección 6.5 permite, finalmente, completar el diseño al profundizar en la definición del dominio de aplicación que emplearemos para validar el modelo.

6.2 BREVE DESCRIPCIÓN DE LA FUNCIONALIDAD DE LA PLATAFORMA

En base a las consideraciones anteriores, y teniendo presente lo descrito en los capítulos previos, es posible establecer algunas de las funcionalidades que debe poseer nuestra plataforma multiagente.

Podemos empezar, por ejemplo, con aquellas relacionadas con su gestión. Así, lo más evidente es que los usuarios, clientes y empresas del VBE, han de poder solicitar participar en la plataforma. En otras palabras, sus agentes han de poder darse de alta, con todo lo que esto supone: aportar información sobre sus competencias (procesos y actividades) y capacidades, datos identificativos, etc. Y dado que las empresas son entes vivos que evolucionan a lo largo del tiempo, también han de poder modificar en cualquier momento la información registrada (porque ofrecen nuevos servicios o dejan de prestar algunos que ya suministraban, etc.) e incluso darse de baja.

El administrador de la plataforma debe, por su parte, tener la posibilidad de verificar toda esa información, registrarla o eliminarla si fuera necesario, etc. Para ello ha de tener acceso a la información existente en el entorno de gestión sobre sus componentes y sobre sus actividades (desempeño, eficiencia, calidad, grado de confianza, etc.). También ha de poder modificar la información sobre los usuarios fruto de su estrecha relación con el administrador del VBE (lo ideal es que coincidan ambas figuras, es decir, que el administrador de la plataforma sea, al mismo tiempo, el administrador del entorno de gestión) siempre que sea necesario.

Además de las funcionalidades propias de la administración de la plataforma, están aquellas relacionadas más directamente con el objeto de la tesis: la selección de socios. Y la primera que aflora no es otra que la de permitir a los clientes (ya sean externos o internos) publicar sus necesidades de provisión de un proceso de negocio o actividad particular, o bien solicitar una lista con los posibles suministradores. De esta forma nuestra plataforma permitirá la creación de VE dinámicas de una forma más realista. Naturalmente, ha de proporcionar todas las herramientas necesarias para que se establezcan las negociaciones oportunas.

Cualquier usuario ha de poder seleccionar el grado de control que desea sobre los diferentes subprocesos y actividades que componen un proceso de negocio; por ejemplo, tener un control total sobre todos ellos, ninguno o bien

sobre unos pocos (estableciendo un menor control sobre el resto o incluso permitiendo la formación de DVE espontáneas). En consecuencia, la plataforma ha de permitir que los usuarios puedan definir fácilmente su comportamiento, estableciendo tanto el grado de control sobre los diferentes procesos de negocio remotos como los criterios y reglas de actuación necesarios.

Por otro lado, cuando en la plataforma no existan candidatos para un proceso determinado, o bien los existentes no sean apropiados, los usuarios han de poder comunicárselo al administrador de la plataforma. Éste, a su vez, informará al administrador del VBE que se encargará de invitar a nuevas empresas (internas o externas al VBE dependiendo de la necesidad) para que llenen el vacío existente. También han de poder informar de la necesidad de introducir nuevas características de los procesos en la negociación. De esta forma, el administrador podrá modificar las plantillas que definen las actividades de negocio y sus relaciones, lo que le permitirá mantener el sistema continuamente actualizado.

Una vez presentados los elementos que mejor describen la funcionalidad de la plataforma llega el momento de centrarnos en aquellos aspectos relacionados con su análisis y diseño.

6.3 ANÁLISIS

El análisis tiene por objeto clarificar el problema a un nivel de detalle suficiente. En la metodología adoptada esta fase se lleva a cabo a través de varios pasos, que pueden resumirse como sigue (Nikraz *et al.*, 2006):

- *Paso 1: construcción de los casos de uso.* Se crea un diagrama de casos de uso en base al análisis de los requisitos del sistema.
- *Paso 2: identificación de los tipos de agentes iniciales.* Aplicando ciertas reglas se genera un diagrama inicial del sistema multiagente llamado diagrama de agentes.
- *Paso 3: identificación de las responsabilidades.* Observando los tipos de agentes generados en el diagrama de agentes y aplicando reglas adicionales se construye la tabla inicial de responsabilidades para aquellos agentes cuyas responsabilidades estén claras inicialmente.

- *Paso 4: identificación de las amistades.* Se identifican las amistades evidentes entre los agentes y, posteriormente, se actualizan tanto el diagrama de agentes como la tabla de responsabilidades.
- *Paso 5: refinamiento de los agentes.* El diagrama de agentes y la tabla de responsabilidades se actualizan en base a consideraciones relacionadas con el soporte, descubrimiento, gestión y supervisión.
- *Paso 6: información de despliegue de los agentes.* Se genera el diagrama de despliegue de los agentes, donde se indican los agentes y los dispositivos físicos donde los agentes van a ser desplegados.
- *Iterar entre los pasos 1 a 6.*

Los elementos importantes que se logran obtener al llevar a cabo los pasos anteriores son los artefactos, es decir, los diagramas, esquemas o documentos que representan de forma gráfica o textual uno o más modelos del sistema. Estos artefactos forman la base para la fase de diseño. Los artefactos producidos en cada paso y sus relaciones se resumen en la figura 6.1.

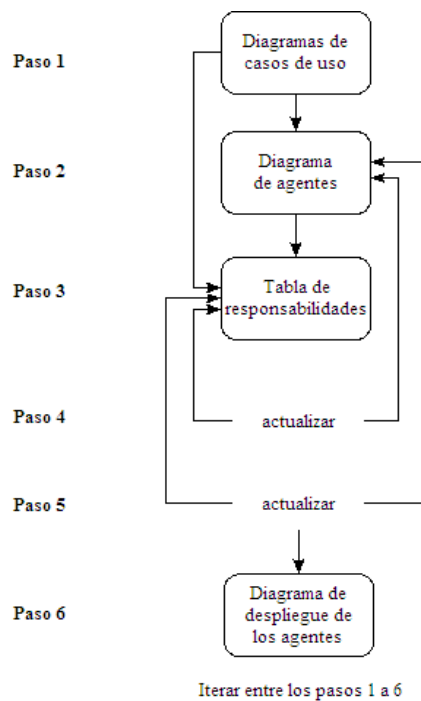


Figura 6.1. Resumen de la fase de análisis desde la perspectiva de los artefactos producidos (Nikraz *et al.*, 2006).

Los apartados 6.3.1 a 6.3.6 describen el análisis de nuestro problema atendiendo a esta secuencia de pasos. Naturalmente, aquí sólo se muestran los resultados y artefactos finales, fruto de las sucesivas iteraciones (incluidas aquellas que se producen durante la fase de diseño), ya que mostrar todos los generados durante el proceso sería redundante y totalmente innecesario para los propósitos de este capítulo.

6.3.1 Paso 1: construcción de los casos de uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario (Schmuller, 2000). Cada caso de uso presenta unos o más escenarios que muestran cómo debe interactuar el sistema con el usuario final, u otro sistema, para alcanzar una meta específica. Para los desarrolladores de sistemas esta es una herramienta valiosa ya que constituye una manera eficaz de capturar los posibles requisitos funcionales de un nuevo sistema desde el punto de vista del usuario.

Aunque los casos de uso surgen en el campo de los sistemas orientados a objetos, su aplicabilidad no se restringe a dichos sistemas (Hampton *et al.*, 1997). Por tanto, también es posible aplicar los casos de uso para capturar los requisitos funcionales de los sistemas multiagente. Existen varios estándares para representar los casos de uso, aunque el más popular es la especificación UML (UML web, 2007) que define una notación gráfica.

Partiendo de la descripción de la funcionalidad de la plataforma presentada en el apartado anterior, es posible construir una lista preliminar de posibles escenarios atendiendo a los diferentes servicios que debe prestar la plataforma (por ejemplo, darse de alta en la plataforma, buscar proveedores de un proceso de negocio, etc.). Esto nos permite definir los casos de uso de la plataforma y producir diferentes diagramas como los que se muestran en las figuras 6.2 a 6.5.

En particular, la figura 6.2 representa el diagrama de casos de uso de alto nivel para el estudio de la plataforma, donde aparecen reflejados los diferentes usuarios de la plataforma y las funcionalidades de la misma. La figura 6.3 representa, por su parte, la relación jerárquica existente entre los actores de la plataforma. Como puede observarse, todos ellos derivan de la figura genérica “Empresa”, que representa tanto a empresas reales, personas físicas u otras clases de organizaciones.

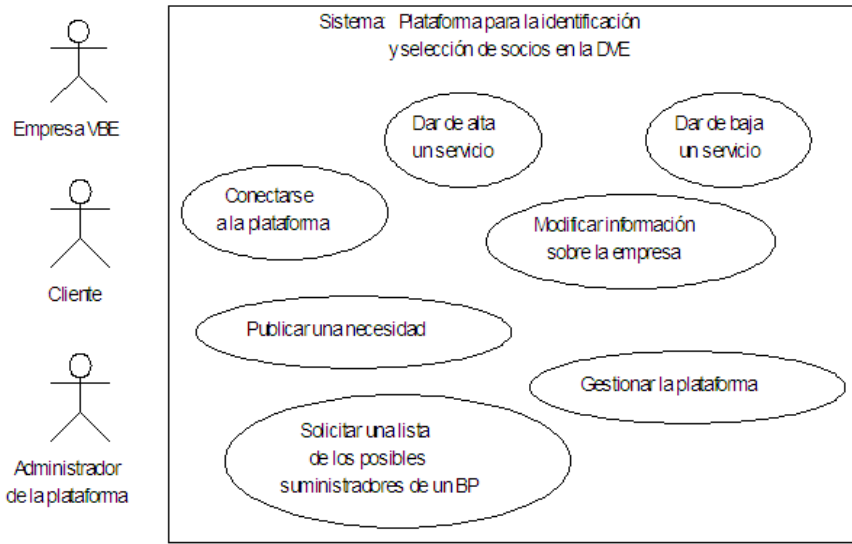


Figura 6.2. Diagrama de casos de uso de alto nivel para el estudio de la plataforma.

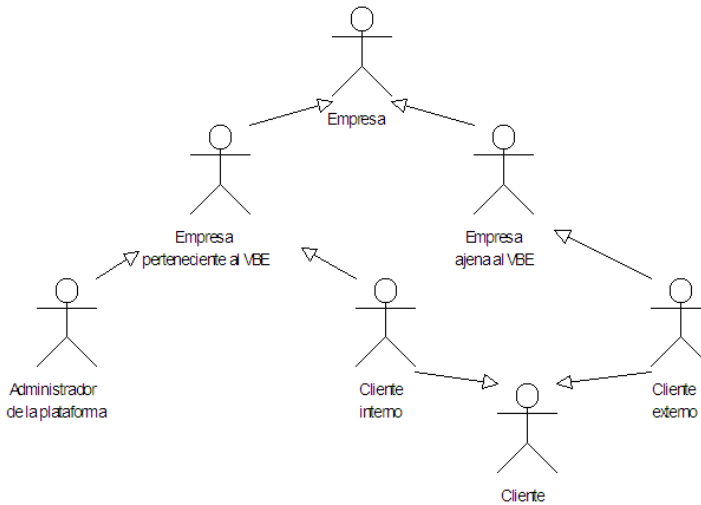


Figura 6.3. Jerarquía de usuarios de la plataforma.

Conviene recordar que el VBE ya está constituido antes de implantar la plataforma multiagente. Es decir, la plataforma tan sólo es una parte de la infraestructura ICT común de los diferentes miembros del VBE, incluidos los clientes externos⁶¹. Además, consideramos que el espacio de búsqueda está formado inicialmente por los miembros del VBE y por sus socios en las actividades habituales. La búsqueda en el universo abierto de empresas no se considera formalmente, aunque obviamente cada uno de los agentes puede decidir rechazar las ofertas de las empresas existentes en la plataforma y encontrarse en la necesidad de buscar fuera de ella⁶².

De los diferentes casos de usos de la plataforma, los más interesantes en este momento son “dar de alta un servicio⁶³ en la plataforma” y “selección del proveedor”, en las dos vertientes consideradas en el modelo de partida, que se muestran a continuación en las figuras 6.4 a 6.6.

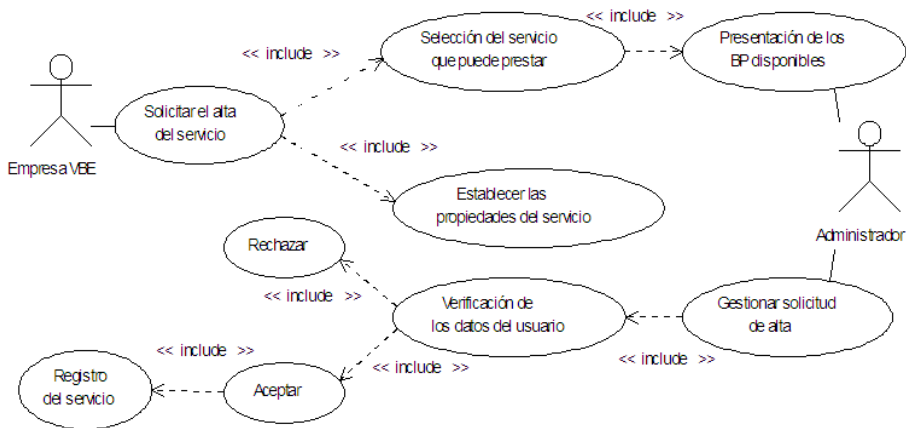


Figura 6.4. Diagrama de casos de uso “Dar de alta un servicio en la plataforma”.

61 El cliente externo es, temporalmente, parte del VBE: se le deja compartir la infraestructura tras una exhaustiva investigación previa.

62 Recordar nuevamente que para estas situaciones la plataforma ha de permitir que el usuario informe sobre la necesidad de nuevos proveedores cuando estos no existan, al menos en el número y condiciones adecuados.

63 Obviamente, cuando hablamos de un servicio nos estamos refiriendo a un proceso, subproceso o actividad que será local si la empresa es capaz de desarrollar por sí misma, o que será remota o distribuida si es necesario que le suministre un tercero.

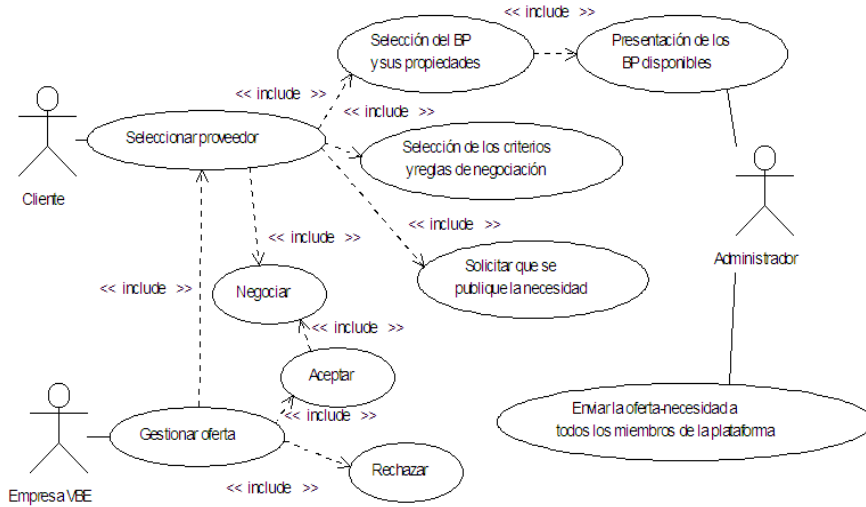


Figura 6.5. Diagrama de casos de uso “Selección del proveedor a partir de la publicación de la necesidad”.

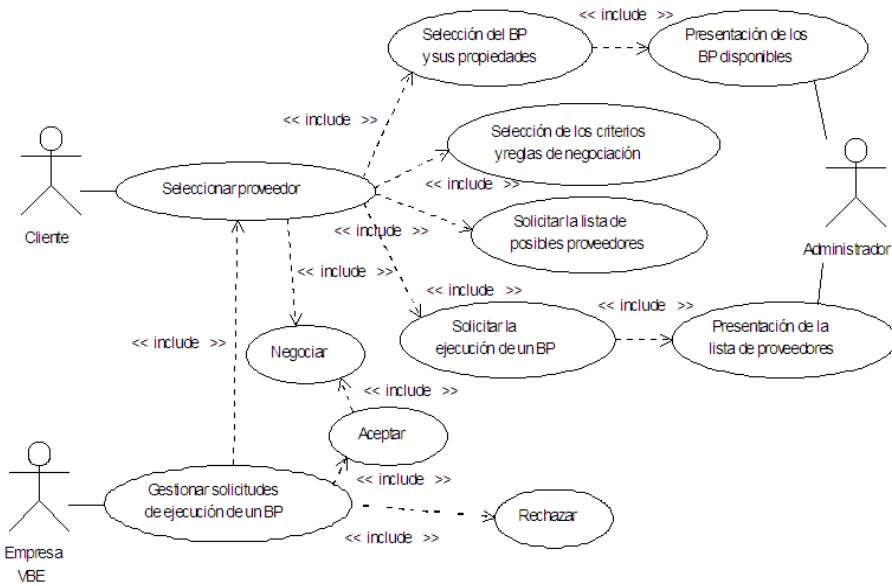


Figura 6.6. Diagrama de casos de uso “Selección del proveedor a partir de la solicitud de una lista de proveedores”.

6.3.2 Paso 2: identificación de los tipos de agente iniciales

Este paso conlleva la identificación de los principales tipos de agentes y de la consiguiente elaboración de un primer bosquejo del diagrama de agentes. En este diagrama se distinguen cuatro tipos de elementos (Nikraz *et al.*, 2006):

- *Tipos de agentes*: representados mediante círculos.
- *Seres humanos*: las personas físicas que deben interactuar con el sistema bajo desarrollo. Se representan mediante el símbolo de actor de UML.
- *Recursos*⁶⁴: sistemas externos que deben interactuar con el sistema bajo desarrollo, representados mediante rectángulos.
- *Relaciones*: están representadas mediante una flecha que une las instancias de los elementos anteriores, especificando que los elementos enlazados tendrán que interactuar de alguna manera mientras el sistema esté operativo.

Atendiendo a la metodología adoptada, debe agregarse un tipo de agente por cada usuario/dispositivo y por cada recurso. Si aplicamos esta sencilla regla al caso de la plataforma se obtiene el diagrama de la figura 6.7.

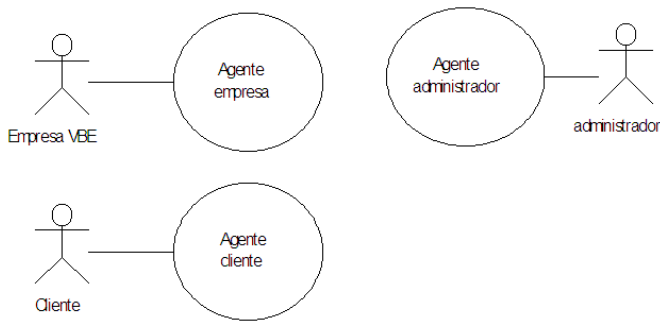


Figura 6.7. Diagrama de agentes después del paso 2.

64 Como puede apreciarse, y al contrario de lo que sucede en los diagramas de casos de uso de UML, en el diagrama de agentes se establece una distinción explícita entre los seres humanos y los sistemas externos (recursos). Este hecho se debe a que la interacción con una persona a través de una interfaz de usuario presenta algunos problemas adicionales con respecto a la interacción con un sistema externo, tal y como se pondrá de manifiesto posteriormente en la fase de diseño.

En el diagrama de agentes de la figura 6.7 los agentes están actuando como traductores, es decir, como una interfaz entre los sistemas externos/personas y los otros agentes del sistema. De forma resumida, el agente traductor acepta mensajes de los agentes del sistema (en el lenguaje de comunicación de agentes), los traduce a la lengua materna de los sistemas externos/personas, y finalmente les transmite los mensajes equivalentes. De igual manera ocurre en la dirección contraria.

6.3.3 Paso 3: identificación de las responsabilidades

En este paso, para cada tipo de agente identificado se hace una lista inicial de sus responsabilidades principales; el artefacto que resulta de este proceso es la tabla de responsabilidades. En este paso debemos derivar el conjunto de responsabilidades de los casos de uso identificados en el paso primero e identificar las responsabilidades de los agentes. Aplicando estas reglas al estudio de la plataforma se genera la tabla 6.1 que aparece a continuación.

Tipo de agente	Responsabilidades
Agente empresa	Atender las peticiones de la empresa perteneciente al entorno de gestación DVE para dar de alta un proceso de negocio en la plataforma. Permitir a la empresa insertar y seleccionar los datos necesarios para el alta del servicio-proceso. Recuperar los servicios disponibles en la plataforma. Atender las peticiones de baja de servicios. Atender las peticiones de la empresa para modificar la información registrada sobre los diferentes procesos de negocio que suministra. Permitir modificar la información disponible actualmente en la plataforma. Recuperar la información registrada en la plataforma sobre la empresa. Atender las peticiones de la empresa para seleccionar un proveedor de un determinado servicio. Permitir seleccionar el servicio que se desea obtener. Dar la posibilidad de elegir entre solicitar la lista con los potenciales proveedores de un proceso de negocio o solicitar que se publique la necesidad en la plataforma.

Tabla 6.1. Tabla de responsabilidades después del paso 3.

Tipo de agente	Responsabilidades
Agente cliente	<p>Atender las peticiones del cliente para seleccionar un proveedor de un determinado servicio.</p> <p>Recuperar los servicios disponibles en la plataforma.</p> <p>Permitir seleccionar el servicio que se desea obtener.</p> <p>Dar la posibilidad de elegir entre solicitar la lista con los potenciales proveedores de un proceso de negocio o solicitar que se publique la necesidad en la plataforma.</p>
Agente administrador	<p>Permitir al administrador modificar los datos sobre los servicios (eliminar, modificar, añadir).</p> <p>Permitir al administrador modificar los datos sobre los miembros de la plataforma.</p>

Tabla 6.1. (Continuación) Tabla de responsabilidades después del paso 3.

6.3.4 Paso 4: identificación de las amistades⁶⁵

En este paso, la atención se centra en las interacciones que se precisan en el sistema. El resultado es una actualización del diagrama de agentes (figura 6.7) en el que se agregan relaciones de amistad que conectan aquellos agentes que necesitan tener una o más interacciones.

Se requiere una relación de amistad obvia entre el agente cliente y el agente empresa (el solicitante y el solicitado) entre los que se dará el proceso de negociación. Naturalmente, cuando el cliente sea interno las relaciones serán entre los propios agentes empresa. Además, dado que el agente administrador debe informar sobre los servicios disponibles en la plataforma y sobre los agentes proveedores de un determinado servicio, tanto al agente cliente como a los agentes empresa, existirá ciertamente una relación de amistad entre el agente administrador y los agentes cliente y empresa.

Volviendo un paso atrás (paso 3), se pueden agregar algunas nuevas responsabilidades a los agente cliente, empresa y administrador. Para el agente empresa, éstas son:

⁶⁵ El término “amistades” o “conocidos” viene de Gaia (Wooldridge *et al.*, 2000), y se utiliza en el mismo sentido en la metodología para JADE.

- Recuperar las solicitudes/ofertas que lleguen de otros agentes.
- Aceptar/rechazar una solicitud/oferta entrante.
- Negociar con el agente solicitante.

Para el agente cliente, las nuevas responsabilidades son idénticas a las del agente empresa sólo que, en lugar de tratar tanto solicitudes como ofertas, sólo considerará estas últimas. Finalmente, el agente administrador precisa de las siguientes responsabilidades:

- Responder a las peticiones de recuperación de los servicios existentes que realizan tanto el agente cliente como el agente empresa.
- Responder a las solicitudes de alta de los servicios (aceptar o rechazar).
- Verificar solicitudes (solicitante y servicios).
- Registrar los servicios de los solicitantes.
- Responder a las solicitudes de baja de los servicios.
- Eliminar el registro de los servicios que han de darse de baja.
- Enviar la oferta/necesidad a los miembros de la plataforma.
- Responder a las solicitudes de modificación de información.

En consecuencia, tanto el diagrama de agentes como la tabla de responsabilidades se actualizan (aunque aquí tan sólo se muestra el nuevo diagrama de agentes en la figura 6.8). Obsérvese que no se hace ninguna distinción entre las amistades y las responsabilidades; por tanto, todas las amistades se sitúan en la tabla de responsabilidades y no por separado.

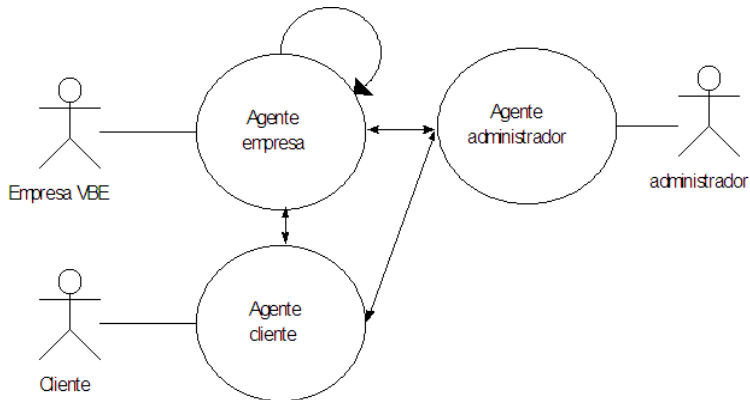


Figura 6.8. Diagrama de agentes refinado tras el paso 4.

6.3.5 Paso 5: refinamiento de los agentes

En este paso, el conjunto de tipos de agentes identificados inicialmente en el paso 2 es refinado aplicando varias consideraciones. Éstas están relacionadas con el soporte (qué agentes de información de apoyo se necesitan, y cómo, cuándo y dónde se genera/almacena dicha información), el descubrimiento (cómo se descubren los agentes unidos por una relación de amistad) y, finalmente, la gestión y supervisión (se requiere que el sistema siga la pista de los agentes existentes, o bien sólo su inicio y finalización bajo demanda).

En cuanto al soporte la pregunta principal es, ¿cómo se verifican los datos de los usuarios?. La verificación se realiza a través de información precisa que posee el administrador del VBE y a la que se accede a través de un nuevo *agente proveedor de información*. Volviendo al paso 4, debe agregarse una relación de amistad entre el agente administrador y el agente proveedor de información (véase figura 6.9), y una nueva responsabilidad para cada agente: “permitir recuperar información sobre el solicitante” y “responder a las peticiones de recuperación de información del agente administrador”, respectivamente (véase tabla 6.3).

Por otro lado, ¿qué sucede si han de colaborar empresas con distinta divisa?, ¿cómo puede un cliente comparar ofertas si en cada una de ellas se emplea una moneda diferente?. Para solucionar esta situación necesitamos un agente capaz de realizar, de forma actualizada, la migración de una moneda a otra⁶⁶. Este agente, al que denominaremos *agente de servicio divisa*, solamente interactuará con el agente empresa, ya que consideramos más realista que sea éste último el que responda a las ofertas en la moneda del solicitante y no que el solicitante tenga que realizar el cambio para cada oferta.

Por tanto, en el diagrama de agentes debe agregarse una relación de amistad entre el agente empresa y el agente de servicio divisa (figura 6.9), y la tabla de responsabilidades debe actualizarse con dos nuevas responsabilidades: el agente empresa podrá “solicitar el cambio de una moneda a otra” y el agente de servicio divisa deberá “responder a las peticiones de cambio de moneda” (véase tabla 6.3).

⁶⁶ En nuestro caso, la información monetaria estará almacenada en documentos XML que gestionará el propio administrador de la plataforma. No obstante, lo ideal sería que dicha información se actualizase de forma continua y automática a través de Internet, y que fuese independiente del administrador.

Puede plantearse, además, una nueva pregunta, ¿cómo calculo el coste de transporte de las distintas materias primas y productos elaborados o semielaborados?. Para responder a esta pregunta necesitaremos otro agente de servicio capaz de establecer dicho coste en función de los puntos de origen y destino, el tipo de travesía (la más rápida, la más económica, la más corta, etc.), el medio de locomoción, el coste del combustible actualizado, las aduanas, los peajes, el tamaño y número de los vehículos, etc. En definitiva, un agente⁶⁷ capaz de calcular costes al estilo de algunos portales cartográficos como Via Michelin (Via Michelin web, 2007) o Mappy (Mappy web, 2007).

Este nuevo *agente de servicio transporte* solamente interactuará con el agente empresa, ya que son sólo las empresas pertenecientes a la plataforma (y no los clientes) las que deben proporcionar un valor monetario por su servicio/producto y, por tanto, deben conocer todos los costes en los que incurren durante la prestación del servicio. En consecuencia, necesitamos agregar una nueva relación de amistad entre el agente empresa y el agente de servicio transporte (figura 6.9), y cuatro nuevas responsabilidades: el agente empresa podrá “solicitar el cálculo” tanto “del coste de transporte” como “de la distancia entre dos puntos”, mientras que el agente de servicio transporte deberá “responder a las peticiones de cálculo” “del coste de transporte” o “de la distancia entre dos puntos” (véase tabla 6.3).

En cuanto al descubrimiento, podríamos optar por la utilización de convenciones de nombrado adecuadas dada su simplicidad y eficiencia. Sin embargo, tiene algunas limitaciones entre las que destacan las siguientes: los nombres de los agentes deben ser globalmente únicos, debe conocerse con antelación los agentes que van a estar implicados en una interacción, normalmente no es muy extensible, pueden generar trabajo adicional cuando se aplican a un agente que puede aparecer y desaparecer dinámicamente, y no pueden ser adoptadas cuando diferentes usuarios puedan comenzar sus propios agentes y elegir sus nombres.

Una manera más sofisticada de solucionar este problema consiste en la adopción de un mecanismo de páginas amarillas, *Yellow Pages*. Esto permite descubrir a los agentes a partir de sus características, por ejemplo los servicios que proporcionan. Un mecanismo de páginas amarillas puede estar

67 Al igual que sucede con el agente de servicio divisa, la información relativa al cálculo del coste de transporte entre dos puntos se obtiene de una base de información en XML gestionada por el administrador de la plataforma.

completamente distribuido a través de todos los agentes del sistema o centralizado con un único agente responsable de él. Aun cuando se trata de una elección de diseño de alto nivel, dado que hacemos uso de la plataforma JADE adoptaremos un enfoque centralizado que se mapea totalmente con el agente *df* (*directory facilitator*) suministrado por JADE, lo que permite ahorrar mucho trabajo en las fases sucesivas del proceso de desarrollo.

En cuanto a la gestión introduciremos un nuevo agente, denominado *agente de acceso*, cuya única función será verificar la pertenencia del nuevo usuario a la plataforma (a través del agente proveedor de información) cuando éste quiera conectarse. Por lo que se refiere a la supervisión, en el estudio que nos trae aquí no se necesita añadir ningún nuevo tipo de agente para estos propósitos. No obstante, podría ser interesante incluir en trabajos futuros agentes para, por ejemplo, supervisión de fallos, restauración, etc.

Una vez refinado el conjunto de tipos de agentes, el proceso exige volver a los pasos 2, 3, y 4, e iterar hasta que se logren unas descripciones lo suficientemente detalladas de los tipos de agentes, sus responsabilidades y las relaciones de amistad, respectivamente. Haciendo esto con nuestro ejemplo, se obtienen los artefactos mostrados en la figura 6.9 y la tabla 6.2.

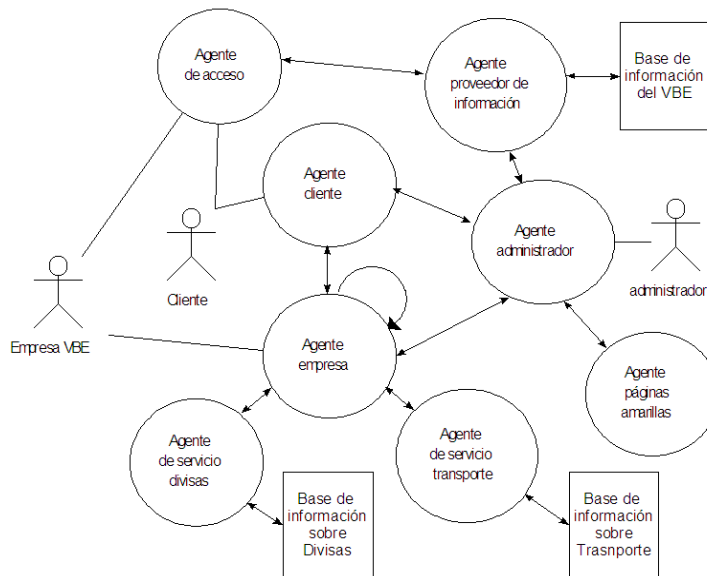


Figura 6.9. Diagrama de agentes después del paso 5.

Tal y como puede apreciarse en la figura 6.9, el agente “páginas amarillas” sólo se relaciona con el agente administrador. Este hecho es premeditado, ya que pretendemos que tanto el registro como la búsqueda de los servicios solamente las pueda realizar dicho agente; de este modo, el agente administrador podrá verificar en cualquier momento los datos que se precisa introducir y nadie podrá modificar arbitrariamente la información registrada sin el permiso correspondiente.

Tipo de agente	Responsabilidades
Agente empresa	<ol style="list-style-type: none"> 1 Atender las peticiones de la empresa perteneciente al VBE para dar de alta un servicio en la plataforma. 2 Permitir a la empresa insertar y seleccionar los datos necesarios para el alta del servicio. 3 Recuperar y presentar los servicios disponibles en la plataforma. 4 Atender las peticiones de baja de servicios. 5 Atender las peticiones de la empresa para modificar la información registrada sobre los servicios que suministra. 6 Permitir modificar la información disponible. 7 Recuperar y presentar la información actualmente registrada en la plataforma sobre la empresa. 8 Atender las peticiones de la empresa para seleccionar un proveedor de un determinado servicio. 9 Permitir seleccionar el servicio que se desea obtener. 10 Dar la posibilidad de elegir entre solicitar la lista con los potenciales proveedores de un proceso de negocio o solicitar que se publique la necesidad en la plataforma. 11 Recuperar la lista de potenciales proveedores. 12 Solicitar que se publique la oportunidad/oferta de negocio (equivale a enviar la invitación a todos los agentes que han registrado el proceso de negocio en la plataforma). 13 Recuperar las ofertas y solicitudes que lleguen de otros agentes (empresa o cliente). 14 Filtrar las solicitudes y ofertas de otros agentes. 15 Presentar al usuario las ofertas/solicitudes entrantes. 16 Negociar con el agente proveedor. 17 Negociar con el agente solicitante. 18 Solicitar el cambio de una moneda a otra. 19 Solicitar el cálculo del coste de transporte. 20 Solicitar el cálculo de la distancia entre dos puntos.

Tabla 6.2. Tabla de responsabilidades tras el paso 5.

Tipo de agente	Responsabilidades
Agente cliente	El agente cliente comparte ciertas responsabilidades con el agente empresa, como son todas aquellas relacionadas con la selección de un proveedor para un servicio concreto; nos estamos refiriendo a las responsabilidades 9 a 17, aunque teniendo presente que el agente cliente sólo podrá evaluar ofertas y no solicitudes de provisión de un servicio dado.
Agente de acceso	<ol style="list-style-type: none"> 1 Permitir al usuario introducir los datos de acceso. 2 Comprobar que el usuario pertenece a la plataforma.
Agente administrador	<ol style="list-style-type: none"> 1 Permitir al administrador modificar los datos sobre los servicios (eliminar, modificar, añadir). 2 Recuperar y mostrar los servicios y sus características. 3 Responder a las peticiones de recuperación de servicios. 4 Responder a las solicitudes de alta, baja y modificación de servicios por parte de los agentes empresa. 5 Verificar solicitudes (solicitante y servicios). 6 Solicitar información sobre los usuarios (calidad, fiabilidad, etc.) al agente proveedor de información. 7 Registrar, modificar o eliminar la información sobre los servicios en el agente páginas amarillas incluyendo información sobre los proveedores.
Agente proveedor de información	<ol style="list-style-type: none"> 1 Responder a las peticiones de recuperación de información del agente administrador y del agente de acceso. 2 Permitir al agente administrador modificar los datos sobre los miembros de la plataforma. 3 Recuperar información sobre los miembros de la plataforma. 4 Mostrar la información sobre los miembros.
Agente de servicio divisa ⁶⁸	<ol style="list-style-type: none"> 1 Responder a las peticiones de cambio de una moneda a otra.
Agente de servicio transporte ⁶⁸	<ol style="list-style-type: none"> 1 Responder a las peticiones de cálculo del coste de transporte 2 Responder a las peticiones de cálculo de la distancia entre dos puntos.

Tabla 6.2. (Continuación) Tabla de responsabilidades tras el paso 5.

68 Obsérvese que no incorporamos la posibilidad de que estos agentes permitan al administrador modificar la información presente en la base de información. Esta decisión tiene como objeto una mejora futura que permita obtener la información relativa a la divisa y al transporte por medio de herramientas *ad hoc* de Internet.

6.3.6 Paso 6: información de despliegue de los agentes

Otro artefacto cuya generación puede ser útil es el diagrama de despliegue de los agentes, donde se indica cómo van a ser distribuidos los agentes físicos. El diagrama de despliegue para nuestro escenario se muestra a continuación:

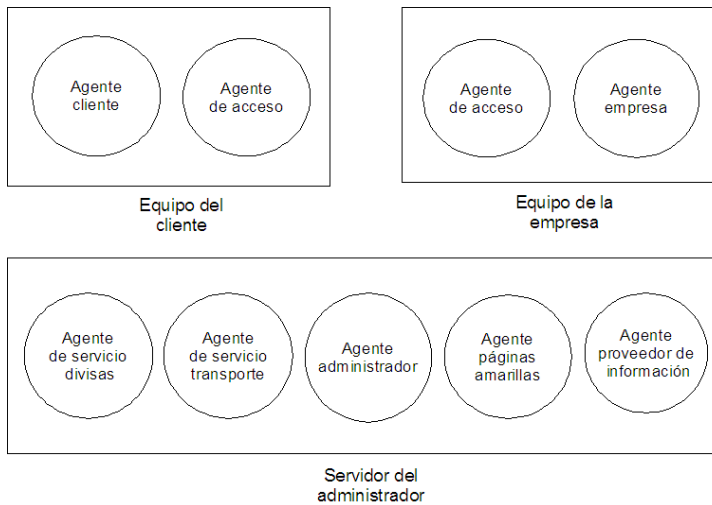


Figura 6.10. Diagrama de despliegue después del paso 6.

La mayor parte de los agentes residen en el servidor del administrador que es, por otra parte, el lugar en el que de momento reside la mayor parte de la información que precisa la plataforma y de la que se nutren el agente proveedor de información y los agentes de servicio. En los equipos de las empresas y los clientes encontramos, a parte de sus respectivos agentes, un agente de acceso encargado de validar su entrada a la plataforma.

Debe observarse que este diagrama no está pensado para dar información detallada sobre el despliegue⁶⁹. El propósito único del diagrama de despliegue de los agentes es destacar los requisitos básicos del despliegue a los que se refieren durante el diseño cuando se aplican consideraciones tales como división y combinación de agentes (sección 6.4.1) o cuando se considera la eficiencia de la comunicación.

⁶⁹ En contraste con el diagrama de despliegue de UML donde se dan detalles tales como los modos de comunicación entre los nodos.

6.4 DISEÑO

Una vez clarificado el problema a un nivel de detalle suficiente, debemos pasar a la fase de diseño, cuyo propósito no es otro que el de especificar la solución. Aunque algunos enfoques consideran las fases de análisis y diseño como elementos estancos, en la metodología adoptada no existe un límite estricto entre ambas fases, tal y como puede apreciarse al observar la figura 6.11. Además, dicha metodología es de naturaleza iterativa, permitiendo que el diseñador pueda moverse hacia adelante y hacia atrás entre ambas fases siempre que sea necesario.

Otra peculiaridad interesante es que durante la fase de diseño se asume que JADE es la plataforma de implementación, por lo que la metodología se centra específicamente en las construcciones que suministra (clases, conceptos, comportamientos, etc.). Este hecho permite una transición directa a la fase de construcción sin tener que adaptar los resultados de la fase de diseño a la plataforma de agentes seleccionada. Obviamente, esta decisión proporciona una mejora significativa de tiempo y provee, además, de una visión mucho más clara de cómo progresar en la implementación.

Al igual que el análisis, la fase de diseño se realiza mediante un número de pasos lógicos consecutivos, con un cierto grado de solapamiento. En este caso, los autores de la metodología consideran que no es preciso seguir estrictamente el conjunto de hitos que establecen y se pueden agregar o quitar pasos, y otras especificaciones relevantes, cuando se desee. No obstante, nosotros nos ceñiremos estrictamente a las etapas definidas para lograr una mayor adecuación a la metodología.

En concreto, los pasos que definen para la fase de diseño se pueden resumir como sigue (Nikraz *et al.*, 2006):

- *Paso 1: dividir/unir/renombrar agentes.* Teniendo en cuenta el funcionamiento y la complejidad del sistema en relación con el diagrama de despliegue de agentes generado en el análisis, se establece si los agentes deben ser partidos, combinados o se les deja como están.
- *Paso 2: especificación de las interacciones.* Se consideran todas las responsabilidades (tabla de responsabilidades) que precisen de una relación de amistad con otro agente, y se genera la tabla de interacciones para cada tipo de agente.

- *Paso 3: definición ad hoc del protocolo de interacción.* Para aquellas interacciones en las que no se puede emplear un protocolo de interacción existente, se define un protocolo *ad hoc* usando un formalismo adecuado.
- *Paso 4: plantillas de mensajes.* La tabla de interacciones se actualiza para especificar los objetos `MessageTemplate` adecuados en comportamientos para recibir mensajes entrantes.
- *Paso 5: descripción para ser registrados/buscados (páginas amarillas).* En este paso se formalizan las convenciones de nombrado y los servicios que han de ser registrados/buscados por los agentes en el catálogo de páginas amarillas mantenida por el *directory facilitator* de JADE. Para su representación se utiliza un diagrama de clases de UML.
- *Paso 6: interacciones agente-recurso.* De acuerdo con el diagrama de agentes generado en el análisis, se identifican los recursos pasivos y activos en el sistema, y se determina cómo interactuarán los agentes con estos recursos.
- *Paso 7: interacciones agente-usuario.* En base al diagrama de agentes de la fase de análisis, se identifican y se detallan las interacciones agente-usuario.
- *Paso 8: comportamientos internos del agente.* Partiendo de la tabla de responsabilidades de la fase de análisis, las responsabilidades del agente se mapean a comportamientos del agente. Diferentes tipos de responsabilidades (incluidas las interacciones) requerirán que se especifiquen diferentes tipos de comportamientos de agente.
- *Paso 9: definición de la ontología.* Se especifican las ontologías necesarias, entre las que se encuentra la correspondiente al dominio de aplicación.
- *Paso 10: selección del lenguaje de contenido.* Siguiendo algunas reglas básicas se selecciona un lenguaje de contenido de entre los disponibles.
- *Iterar entre los pasos 1-10.* Esta metodología también propone moverse adelante y atrás en el diseño, y entre el análisis y el diseño, siempre que sea necesario.

La figura 6.11 muestra los artefactos producidos en cada paso del diseño y sus relaciones con aquellos generados durante el proceso de análisis de la plataforma. Las relaciones propias del diseño aparecen en negrita para lograr una mejor comprensión del modelo.

6.4.1 Paso 1: división/unión/renombrado de los agentes

Una vez realizado el análisis de la plataforma, es necesario observar los artefactos generados y determinar si los tipos de agentes que aparecen en el diagrama de agentes deben ser divididos o pueden combinarse entre sí. Esta decisión condiciona notablemente la eficiencia y complejidad del sistema en su conjunto; por ejemplo, tratar con muchos agentes aumenta la complejidad del sistema global y disminuye la eficacia del sistema al producirse una comunicación excesiva, y a veces innecesaria, entre agentes.

En nuestro escenario podría considerarse la unión de los agentes empresa y cliente en uno sólo, dado que este último comparte algunas responsabilidades del primero. Sin embargo, existen razones para no hacerlo: en primer lugar, durante el análisis se ha identificado un número relativamente pequeño de tipos de agente por lo que su posible combinación pasa a ser un asunto de escasa importancia; en segundo lugar, aunque unir ambos agentes no conduciría a crear agentes demasiado grandes y complejos, es más acertado especializar nuestros agentes en función de la pertenencia o no al VBE; por último, cada tipo de agente se sitúa en máquinas distintas.

Por otro lado, y dado que un mismo agente empresa puede verse involucrado en múltiples negociaciones, podría considerarse la posibilidad de dividir dicho agente. Sin embargo, dada la naturaleza de los agentes JADE (“un agente, un hilo”) la operación de la plataforma podría verse ralentizada por el consiguiente aumento de ciclos de CPU que esta opción conlleva. La división de este agente supondría, además, un mayor consumo de memoria. A partir de estas observaciones se intuye que esta decisión puede ser especialmente importante en entornos donde existen recursos limitados, como es el caso de los teléfonos móviles y las PDA⁷⁰.

En consecuencia, y pensando en desarrollos futuros, creemos más acertado dotar a este tipo de agentes de comportamientos que se ejecuten en un mismo hilo buscando así una mayor simplicidad del modelo y una mayor rapidez, puesto que el cambio de comportamiento es más rápido que el cambio de hilo. Además, de esta forma se evitan los problemas de sincronización que pueden aparecer cuando dos comportamientos intentan acceder al mismo recurso (lo que también se traduce en un aumento de la rapidez).

70 Aunque de momento no hemos considerado estos dispositivos en nuestra plataforma, si creemos acertado incluirlos en futuras extensiones.

6.4.2 Paso 2: especificación de las interacciones

En este paso se consideran las responsabilidades que presentan las relaciones de amistad entre los agentes (a partir de la tabla de responsabilidades generada en el análisis), dando lugar a una tabla de interacciones para cada tipo de agente. Cada fila de la tabla representa una interacción e incluye:

- Un *nombre* descriptivo para la interacción.
- La *responsabilidad* (identificada en la tabla de responsabilidades) que origina esta interacción. De esta forma se ligan los artefactos del diseño con los del análisis, lo que podrá ser utilizado más adelante para comprobar la consistencia.
- Un *protocolo de interacción* (IP, *Interaction Protocol*) adecuado elegido para implementar la interacción. Los protocolos de interacción estándares de FIPA deben ser los primeros candidatos; si se considera que ninguno de estos protocolos es apropiado se debe definir un protocolo de interacción *ad hoc*.
- El *rol* que desempeña el agente considerado en el protocolo de interacción. Éste puede ser I para el iniciador (*initiator*) o R para el que responde (*responder*).
- El *tipo de agente* del rol complementario.
- La *condición de activación*, es decir, cuándo se produce la interacción. Debe expresarse de manera informal pero descriptiva.

A continuación se muestra la tabla de interacciones para el agente empresa (tabla 6.3).

Interacción	Resp	IP	Rol	Con	Cuándo
<i>Recuperar los servicios disponibles</i>	3	FIPA Request	I	Agente Administrador	El usuario quiere dar de alta un servicio o seleccionar un proveedor
<i>Modificar la información registrada</i>	6	FIPA Request	I	Agente Administrador	El usuario quiere modificar la información registrada en la plataforma
<i>Recuperar la información registrada</i>	7	FIPA Request	I	Agente Administrador	El usuario quiere recuperar/modificar la información registrada en la plataforma

Tabla 6.3. Tabla de interacciones del agente empresa.

Interacción	Res	IP	Rol	Con	Cuándo
<i>Recuperar la lista de proveedores</i>	11	FIPA Request	I	Agente Administrador	El solicitante quiere negociar directamente con los posibles proveedores
<i>Publicar oportunidad</i>	12	FIPA Request	I	Agente Administrador	El solicitante desea que se formen DVE espontáneas
<i>Recuperar las ofertas y solicitudes</i>	13	FIPA Request	R	Agentes Cliente o Empresa	El usuario recibe las ofertas o solicitudes publicadas en la plataforma por otros agentes
<i>Negociar con el agente proveedor</i>	15	Contract net	I	Agente Empresa	Cuando la empresa busca seleccionar un proveedor para un servicio dado
<i>Negociar con el agente solicitante</i>	16	Contract net	R	Agente Cliente o Empresa	Cuando optamos a suministrar un proceso de negocio
<i>Solicitar el cambio de moneda</i>	18	FIPA Request	I	Agente Divisa	Si la empresa necesita conocer el valor de un producto u operación en otra moneda
<i>Solicitar el coste del transporte</i>	19	FIPA Request	I	Agente Transporte	Cuando queremos conocer el coste de transporte asociado a nuestro proceso
<i>Solicitar la distancia</i>	19	FIPA Request	I	Agente Transporte	Cuando queremos conocer la distancia entre dos lugares

Tabla 6.3. (Continuación) Tabla de interacciones del agente empresa⁷¹.

6.4.3 Paso 3: definición *ad hoc* del protocolo de interacción

Siempre que sea posible, debe adoptarse uno de los protocolos de interacción definidos por FIPA. No obstante, si se considera que ninguno de estos protocolos es adecuado para el caso de estudio, será preciso definir un protocolo de interacción *ad hoc*. Sin embargo, este no es nuestro caso ya que los protocolos que incorpora JADE son suficientes para nuestros propósitos.

71 Aunque no aparecen en este documento, se han construido tablas similares para el resto de tipos de agentes identificados en la fase de análisis.

6.4.4 Paso 4: plantillas de mensajes

Todos los roles del protocolo de interacción identificados en el paso anterior se implementan como comportamientos JADE (véase sección 6.4.8). En este paso se deben especificar los objetos `MessageTemplate` que emplearán aquellos comportamientos que permitan recibir mensajes entrantes. Estas plantillas se agregan a las filas de la tabla de interacciones, por lo que la tabla 6.3 debe ser actualizada con la información que aparece representada en la tabla 6.4.

Interacción	Resp	IP	Rol	Con	Plantilla
<i>Negociar con el agente proveedor</i>	15	Contract net	I	Agente Empresa	Conv-ID, Perf=CFP Prot=CFP, InReplyTo
<i>Negociar con el agente solicitante</i>	16	Contract net	R	Agente Cliente o Empresa	Conv-ID, Perf=CFP Prot=CFP, InReplyTo

Tabla 6.4. Extracto de la tabla de interacciones para el agente empresa después del paso 4 del diseño.

El resto de las interacciones no se han mostrado puesto que su plantilla de mensaje está compuesta por los mismos tres componentes básicos: *conversation-id*, a la que pasamos simplemente un *String* que describe el tipo de interacción que se está produciendo (“registrar-proceso-de-negocio”, “calcular-coste-de-transporte”, etc.), el protocolo de interacción, *Request* en todos los casos, y la performativa⁷² (request, inform, query, etc).

```
MessageTemplate registroDeProcesos =
  MessageTemplate.and (
    AchieveREResponder.createMessageTemplate
      (FIPANames.InteractionProtocol.FIPA_REQUEST),
    MessageTemplate.and (
      MessageTemplate.MatchPerformative (ACLMessage.REQUEST),
      MessageTemplate.MatchConversationId (
        "registrar-proceso-de-negocio")));
```

Figura 6.12. Ejemplo de una plantilla de mensaje para una interacción de tipo FIPA-REQUEST.

⁷² La performativa, o intención comunicativa, indica lo que el remitente intenta conseguir enviando el mensaje.

6.4.5 Paso 5: descripción a ser registrada/buscada (*Yellow Pages*)

En este paso se formalizan las convenciones de nombrado y los servicios que han de ser registrados/buscados por los agentes en el catálogo de páginas amarillas mantenido por el agente *directory facilitator* de JADE. Como es lógico suponer, las convenciones de nombrado dependen del dominio de aplicación que en nuestro caso, y como se discutirá en el apartado 5, es el de la instalación de módulos fotovoltaicos. Utilizamos, además, un lenguaje natural para especificar los servicios lo que permite una fácil interpretación de su significado por parte de cualquier persona.

La figura 6.13 representa el esquema de funcionamiento de los agentes con respecto al servicio de páginas amarillas. Como se puede apreciar, algunos agentes (agente empresa, agente de servicio transporte, etc.) publican los servicios que pueden prestar en el DF, mientras otros buscan los que necesitan para llevar a cabo sus procesos de negocio. Además, un mismo agente puede publicar ciertos servicios y requerir que se le presten otros (Agente AE1).

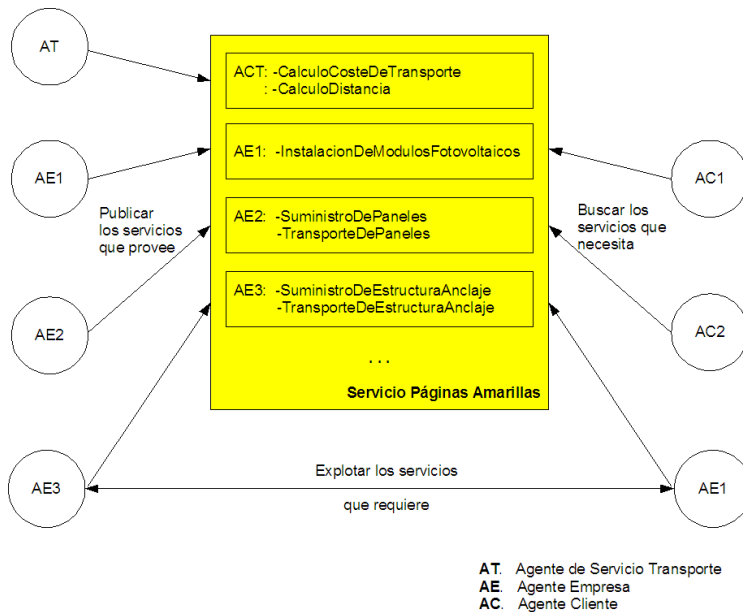


Figura 6.13. Ejemplo simplificado del funcionamiento asociado al servicio de páginas amarillas.

6.4.6 Paso 6: interacciones agente-recurso

A menudo uno o más agentes en el sistema deben interactuar con los recursos externos tales como bases de datos, archivos que almacenan información, aplicaciones software, etcétera. Los agentes que interactúan con los recursos externos se han identificado durante la fase de análisis y están representados en el diagrama de agentes mediante una relación de amistad con un elemento recurso. En nuestro caso sólo hay cuatro recursos activos⁷³, los documentos XML que contienen la información sobre los miembros del VBE, los procesos de negocio, el cambio de divisas y la necesaria para el cálculo del coste de transporte o la distancia entre dos localidades.

Para estas situaciones, la metodología de JADE propone ajustarse al enfoque transductor y evitar embeber código relativo al acceso a los datos dentro de los agentes que necesiten consultar el archivo con la información. Por esta razón se ha optado por emplear cuatro clases como elementos transductores: *XMLInformacionUsuarioScanner*, *XMLInformacionProcesosScanner*, *XMLInformacionMonedaScanner* y *XMLInformacionTransporteScanner*. Estas clases contienen los métodos necesarios para acceder y gestionar (esto último sólo para la gestión de usuarios) la información correspondiente a través de código DOM.

De esta forma logramos que tanto el agente proveedor de información como los agentes de servicio no tengan embebido código DOM, lo que se traduce en un gran número de ventajas de cara al futuro: modularidad, fácil modificación, reutilización, etc.

6.4.7 Paso 7: interacciones agente-usuario

En nuestro escenario existen cuatro agentes que necesitan interactuar con los actores-usuarios del sistema. Estos agentes, que ya han sido identificados en el paso 2 de la fase de análisis y se han representado en el diagrama de agentes mediante una relación de amistad con un elemento actor, son el agente empresa, el agente cliente, el agente de acceso y el agente administrador.

73 Los recursos activos son aquellos que pueden cambiar su estado independientemente del agente que les controle, mientras que los recursos pasivos sólo cambian su estado como consecuencia de algún estímulo generado por el agente que controla el recurso.

Existen varias maneras de que un usuario humano puede interactuar con un software, como es un agente JADE. Creemos acertado emplear la interfaz de usuario gráfica (GUI, *Graphical User Interface*) que es, de lejos, el tipo de interfaz de usuario más comúnmente usada. Dentro de las GUI podemos hablar de GUI web implementadas usando tecnología JSP (*JavaServer Pages*) o GUI locales, normalmente construidas usando Swing, AWT o algún otro conjunto de herramientas gráficas. Dados los requisitos actuales de la aplicación por ahora sólo es necesario crear GUI locales basadas en Swing.

Por lo que se refiere a la interacción, JADE permite varias formas de afrontar el problema de que los agentes y los elementos gráficos operen en diferentes hilos. Nosotros hemos optado por hacer uso de la clase `jade.core.GuiAgent`, construida para este propósito específico. Esta clase es una simple extensión de la clase `jade.core.Agent`, de forma que cuando se ejecuta el método `setup()` instancia un comportamiento *ad hoc* que gestiona una cola de objetos `jade.gui.GuiEvent` que pueden recibirse de otros hilos. La ventaja, en este caso, reside en que este comportamiento se oculta al programador por lo que solamente necesitamos implementar el código específico de la aplicación relativo a cada evento.

6.4.8 Paso 8: comportamientos internos de los agentes

El trabajo real que tiene que hacer un agente normalmente se realiza dentro de sus “comportamientos”. Su definición suele ser una tarea más propia de la implementación que del diseño, salvo cuando el comportamiento se deriva de alguna de las responsabilidades identificadas en la fase de análisis. En estos casos, para cada responsabilidad asociada a una interacción en la tabla de interacciones descrita en el paso 2 de la fase del diseño, se ha de obtener la clase JADE que implemente el protocolo de interacción y el rol seleccionado para esa interacción y proporcionar una extensión adecuada.

Para comprender mejor esta forma de actuar, que será la que permita concretar la mayoría de los comportamientos de nuestra aplicación, vamos a plantear un ejemplo. Consideremos, en concreto, la responsabilidad “solicitar el cálculo del coste de transporte” correspondiente a una interacción en la tabla de interacciones donde el agente empresa desempeña el rol de iniciador en un protocolo de tipo Fipa-Request. En este caso, el comportamiento que implemente dicha responsabilidad será una subclase apropiada de la clase `jade.proto.AchieveREInitiator`.

6.4.9 Paso 9: definición de las ontologías

Necesitamos definir una ontología principal que estará asociada al dominio de negocio considerado, la instalación de módulos fotovoltaicos. También utilizaremos otro conjunto de ontologías asociadas con los distintos tipos de información que se precisan durante su operación. En concreto, crearemos una ontología para la gestión de la plataforma (en la que se considerarán los usuarios y sus características, sus relaciones, la solicitud de información, etc.), y una para cada uno de los agentes de servicio considerados (ontología para el cambio de divisas y para los cálculos relacionados con el transporte).

Pero antes de mostrar estas ontologías, conviene tener presente que en JADE una ontología es simplemente un conjunto de conceptos, predicados y acciones de agente (Caire y Cabanillas, 2004).

- Cuando interactúan los agentes en el sistema, intercambian la información relativa a las entidades que existen en el entorno en el que residen. Estas entidades pueden ser primitivas, por ejemplo un *String* o un número, o puede haber estructuras complejas definidas mediante plantillas especificadas en términos de un nombre y un conjunto de *slots* cuyos valores deben ser de un tipo dado. A estas complejas plantillas de entidades se las conoce como conceptos (*Concepts*). En el dominio de aplicación considerado aparecen algunos conceptos como, p.e., estructura, convertidor, batería, etc.
- Por otra parte, las entidades normalmente están relacionadas por medio de relaciones, que pueden ser bien *true* o *false*, y que, al igual que ocurre con las entidades complejas, también tienen estructuras definidas mediante plantillas. Estas plantillas de relaciones se conocen como predicados (*Predicates*).
- Finalmente, una clase particular de entidad compleja se representa mediante descriptores de las acciones que los agentes pueden realizar. Las plantillas de estos descriptores de acciones se conocen como *AgentActions*⁷⁴.

La figura 6.14 muestra, mediante un diagrama de clases UML, la ontología correspondiente a la gestión de los usuarios.

74 Las acciones, cuando son ejecutadas, pueden producir un efecto y/o generar un resultado que será enviado de vuelta al solicitante.

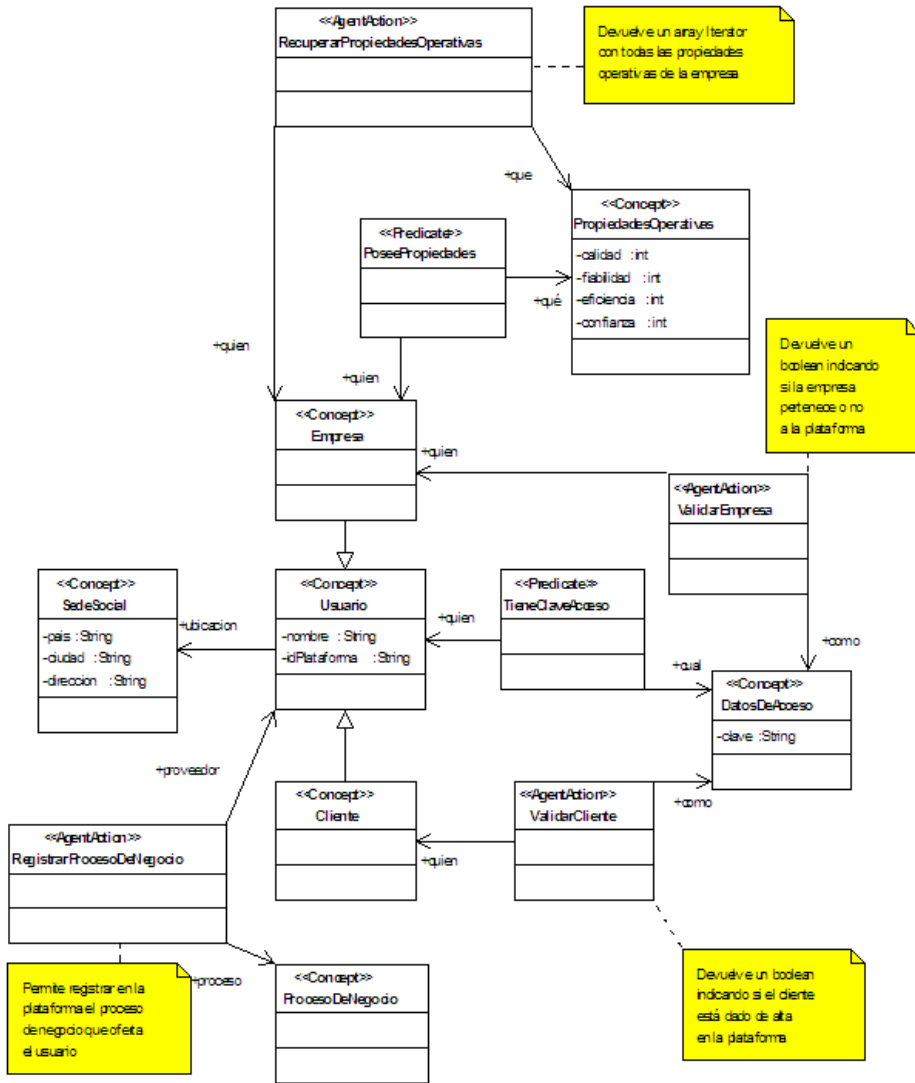


Figura 6.14. Diagrama de clases que representa la ontología de gestión de la plataforma.

En esta representación se pueden apreciar claramente los diferentes elementos que componen una ontología desde la visión JADE (conceptos, predicados y acciones de agentes) y sus relaciones.

Por lo que se refiere a la ontología de nuestro dominio de aplicación principal, la “Instalación de módulos fotovoltaicos”, el número de conceptos, predicados y acciones de agentes que la componen es considerablemente grande. Por ello, no resulta aconsejable representarla gráficamente en este documento. No obstante, la sección 6.5 contiene una representación jerárquica sólo de los conceptos, que se emplea como un artefacto de apoyo visual a la descripción que de ellos se realiza en dicho apartado.

6.4.10 Paso 10: selección del lenguaje de contenido

JADE proporciona *codecs* para dos lenguajes de contenido, SL y LEAP, a través del paquete *jade.content*. El lenguaje SL está codificado mediante *Strings* y es legible para los humanos, mientras que LEAP está codificada en *bytes* y, por tanto, no es legible. SL es adecuado para aplicaciones basadas en agentes que sean, o puedan ser, abiertas (es decir, donde deban comunicarse los agentes de diferentes desarrolladores, operando en plataformas diferentes), mientras que LEAP es preferible cuando hay fuertes limitaciones de memoria (teléfonos móviles, PDA, etc.).

Atendiendo a estas y otras consideraciones, es probable que hubiésemos optado por el lenguaje SL. Sin embargo, éstas no son las dos únicas posibilidades. Cualquier programador puede definir un *codec* si desea que los agentes “hablen” un lenguaje de contenido diferente, siempre y cuando sea consistente con los lenguajes manejados por los recursos; por ejemplo, SQL, XML, RDF, etc. En nuestro caso, los recursos que vamos a emplear son documentos XML por lo que necesitamos un *codec* apropiado. De esta tarea ya se ha encargado el Team JADE (JADE web, 2008), y está disponible desde 2006 con el nombre *XMLCodec*.

6.5 EL DOMINIO DE APLICACIÓN

No hace falta que mencionemos aquí el importante papel que ha ido adquiriendo la energía en nuestra sociedad y, en particular, su obtención a partir de fuentes “limpias”. La más importante de esas fuentes es, sin duda alguna, el Sol, y una de las formas más ampliamente utilizada para aprovechar parte de su energía consiste en el empleo de paneles fotovoltaicos, también conocidos como módulos o colectores solares fotovoltaicos.

Sólo en España, el sector de la energía solar fotovoltaica atrajo inversiones por valor de 5.000 millones de euros durante 2007, emplea ya a más de 15.000 personas y cuenta con 15.000 personas titulares de instalaciones, según anuncia ASIF, Asociación de la Industria Fotovoltaica (ASIF web, 2008). Según los datos de la Comisión Nacional de Energía (CNE), en 2007 se conectaron 428 megavatios (MW) de fotovoltaica a la red eléctrica, un 440% más que el año anterior. En total, el sector cuenta con unos 593 MW instalados, capaces de abastecer la demanda de electricidad de unos 210.000 hogares españoles (EIEconomista web, 2008).

Parece, por tanto, un dominio interesante para construir la plataforma con la que validar nuestro modelo de selección. Y para poder comprender mejor algunos de los elementos de la plataforma (como son la ontología de dominio o las reglas que definirán el comportamiento de los agentes), los próximos apartados se van a dedicar a presentar los diferentes componentes y servicios necesarios para la instalación de paneles fotovoltaicos tanto si son para uso particular como para la construcción/ampliación de huertos o granjas solares, para la instalación de elementos de señalización y control en carreteras, etc.

Pero antes de empezar con esta descripción conviene tener presente dos aspectos. Por un lado, y dado que los posibles clientes finales presentan distintas necesidades, no siempre todos los procesos de negocio que se van a describir en este apartado son necesarios. Por otro, es cierto que se podría haber realizado una descomposición más detallada de los diferentes procesos involucrados; sin embargo, y teniendo presente nuestros objetivos iniciales, este grado de detalle resulta, a todas luces, innecesario.

6.5.1 Paneles o módulos fotovoltaicos

Al conjunto formado por células conectadas en serie o en paralelo, convenientemente ensamblado y protegido contra los agentes externos, se le denomina panel o módulo fotovoltaico⁷⁵. Las células solares son elementos frágiles y se deben proteger por ambos lados. Esto se consigue colocándolas entre una capa de protección superior y otra inferior (con coeficientes de expansión térmica similares y compatibles con el de las células). En la actualidad los plásticos y el vidrio son los materiales más empleados:

75 La forma más usual de construir un generador solar consiste en dividirlo en varios paneles de igual voltaje y potencia.

- El vidrio tiene la ventaja de mantener intactas sus propiedades ópticas y eléctricas durante largos periodos.
- Los polímeros no impiden la penetración de la humedad en las uniones y la metalización, por lo que son apropiados si el silicio subyacente y los otros materiales son resistentes a la corrosión.
- Los plásticos son más ligeros que el vidrio, pero se deben escoger cuidadosamente puesto que algunos tipos pueden perder su transparencia a la luz y su solidez después de una larga exposición a la luz solar y a la atmósfera.

La curva característica del panel o curva de i-v presenta la misma forma que en el caso de las células y varía poco de unos paneles a otros. El estándar internacionalmente aceptado para medir respuestas de los paneles es una intensidad radiante de 1000 W/m², que se conoce como una intensidad de un Sol y una temperatura de la célula de 25 grados Celsius⁷⁶.

EMPLAZAMIENTO DE LAS PLACAS SOLARES

En cuanto a su posible situación existen las siguientes posibilidades (labeled web, 2007):

- *Suelo*: es la forma más usual de instalación de grupos de paneles y presenta grandes ventajas en cuanto al área enfrentada al viento, accesibilidad y facilidad de montaje. Sin embargo, es más susceptible de poder quedar enterrada por la nieve, se inunde o ser objeto de rotura por animales o personas.
- *Poste*: es usual en instalaciones de pequeña dimensión, donde se disponga previamente de un mástil. Es el tipo de montaje típico en alimentación de equipos de comunicación aisladas o farolas.
- *Pared*: presenta ventajas cuando se dispone de buenos puntos de anclaje sobre una edificación construida. Su principal inconveniente procede de los problemas de accesibilidad.
- *Tejado*: como forma de instalación es una de las más habituales si se dispone de espacio suficiente. Sin embargo, presenta problemas por cubrimiento de nieve, menor facilidad de orientación al sur, e impermeabilizado de las sujeciones del techo.

76 Así, la potencia nominal pico de un panel es la proporcionada por el mismo al recibir una irradiación de 1000 W/m² cuando la temperatura es de 25 grados Celsius.

El bastidor que sujeta el panel, la estructura soporte del mismo y el sistema de sujeción son tan importantes como el propio panel, pues un fallo de estos elementos conlleva la inmediata paralización de la instalación. El principal factor a la hora de fijar la estructura no es el peso de los paneles, al ser estos ligeros, sino la fuerza del viento que, dependiendo de la zona, puede llegar a ser muy considerable. La estructura deberá resistir vientos de, como mínimo, 150 Km/h.

CONCENTRACIÓN

Consiste en el empleo de un sistema óptico que permita concentrar la energía solar correspondiente a una superficie S sobre una superficie más pequeña S_c de la célula. El interés de la concentración reside en la obtención de una misma potencia eléctrica disminuyendo la superficie del conjunto de las células solares. Esta disminución es útil cuando los costes derivados del sistema de concentración⁷⁷ son menores que el ahorro obtenido al disminuir la superficie de células necesaria.

Como inconvenientes principales encontramos que requiere un sistema de seguimiento del sol, que la mayor intensidad de la luz provoca un aumento de la temperatura del módulo (lo que reduce su eficiencia) y que las altas temperaturas y los altos niveles de intensidad pueden provocar un fallo prematuro del módulo. Todos estos factores hacen que la concentración se emplee para aplicaciones de media y alta potencia, es decir, en sistemas en los que el grado de potencia demandada obligaría al uso de un gran número de paneles para reducir los costes.

SISTEMAS DE SEGUIMIENTO

Para aprovechar la mayor cantidad posible de energía solar que llega hasta la superficie terrestre en un determinado lugar, la superficie colectora ha de ser perpendicular a los rayos solares. En consecuencia, sólo puede conseguirse una captación óptima si la superficie colectora está dotada de un movimiento de seguimiento del sol.

⁷⁷ En los sistemas fotovoltaicos de concentración una gran parte del coste del sistema corresponde a la estructura mecánica del panel, al subsistema óptico (lentes o espejos generalmente) y al subsistema de seguimiento del sol.

Utilizando el seguimiento, la energía total recibida en un día puede ser del orden de un 30-40% mayor que para el mismo colector estático. Aunque este rendimiento se ve reducido en los casos de frecuentes nublados y en todas aquellas condiciones climatológicas en las que la relación entre la energía recibida por radiación directa y la recibida por radiación difusa tienda a disminuir, el aumento de energía que se consigue normalmente compensa el consumo de energía y el coste de los propios mecanismos de seguimiento.

El seguimiento puede realizarse a través de sensores, mediante el cálculo de coordenadas, por reloj solar, etc., siendo los dos primeros los sistemas más utilizados:

- a) *Seguimiento por sensores.* El sensor es el elemento que permite la detección y medida de la falta de puntería del panel de células solares, puntería que queda determinada por el ángulo que forma el vector Sol con la normal a la superficie colectora. El sensor suele estar constituido por pares de elementos fotosensibles montados sobre el panel y moviéndose solidariamente con él. Los fotosensores se valen de la radiación solar directa para detectar la posición del sol, por lo que ante un ocultamiento deja de producirse el seguimiento; cuando el sol reaparece es preciso emplear un tiempo en la recuperación de la puntería.
- b) *Seguimiento por coordenadas calculadas.* En estos sistemas se sigue la posición del sol mediante el cálculo de sus coordenadas astronómicas. No precisan, por tanto, de la presencia física de los rayos solares, lo que les hace inmunes a los nublados, destellos, etc., que pueden producir errores de puntería en un fotosensor. El empleo de PLC o computadores presentan las ventajas adicionales de que determinados cambios pueden hacerse a nivel de software únicamente y de poder incluir un conjunto de funciones adicionales a la de puntería propiamente dicha, tales como la de llevar los paneles a una posición de máxima seguridad (por la noche o ante las inclemencias del tiempo) o la de retorno a la posición inicial.

MOTORES Y ENGRANES

El seguimiento se realiza por medio de uno o varios motores y de un sistema de engranajes. En el mercado puede encontrarse una gran variedad de motores con las características técnicas que se precisen.

6.5.2 Baterías

La necesidad de acumular la energía suministrada por los paneles fotovoltaicos proviene del hecho de que la distribución temporal del consumo no es totalmente coincidente con las horas de insolación. Pueden describirse diferentes sistemas de almacenamiento, aunque de entre ellas destaca el uso de las baterías de acumuladores (sobre todo en pequeñas instalaciones) debido a su bajo coste de instalación y a su rendimiento.

La batería cumple dos importantes misiones. Por un lado, suministra potencia instantánea, o durante breves momentos, superior a la que el campo de paneles podría generar aún en los momentos más favorables posibles. Por otro, mantiene un nivel de tensión estable y constante (la tensión de salida del panel varía en función de la intensidad radiante, lo que es inadecuado para el funcionamiento de los aparatos) independientemente de las condiciones de incidencia luminosa.

A pesar de su diversa constitución, todas las baterías pueden ser definidas mediante unas características comunes: resistencia interna, rendimiento, velocidad de carga y descarga, vida útil, capacidad de una batería, autodescarga, etc.

MODALIDADES DE UTILIZACIÓN

Las principales modalidades de utilización son (labelec web, 2007):

- *Flotación.* Se dice que una batería trabaja en flotación cuando está conectada en paralelo con una fuente de corriente y al circuito de corriente continua y a una tensión constante. En estas condiciones la fuente suministra, normalmente, el consumo de corriente que demanda el circuito de continua al tiempo que proporciona una pequeña corriente de carga a la batería que compensa la pérdida de capacidad que se produce continuamente por la autodescarga interna, manteniéndola plenamente cargada en todo momento. Las puntas de corriente que ocasionalmente se producen durante breves intervalos de tiempo son suministradas por la batería. Cuando se produce una interrupción en la fuente ordinaria de corriente, la batería se hace cargo de la demanda de corriente continua durante todo el tiempo que dura dicha interrupción o hasta el agotamiento de su capacidad.

Al restablecer de nuevo el servicio, la fuente de corriente vuelve a abastecer las necesidades del circuito de corriente continua y simultáneamente proporciona la corriente de carga a la batería. Una vez completa su carga, se ajusta el voltaje rectificador, bien manual o automáticamente, al valor correspondiente al voltaje de flotación.

- *Ciclos de carga – descarga.* En esta modalidad, la batería se descarga sobre el circuito receptor estando desconectada del sistema de carga. Una vez que la batería se ha descargado o se ha restablecido el suministro normal de corriente haciendo innecesaria su utilización, la batería se conecta al sistema de carga hasta que queda completamente cargada. Este ciclo se repite indefinidamente durante toda la vida de la batería.
- *Tampón.* La batería se encuentra permanentemente conectada en paralelo con el generador de corriente continua. En las horas punta de consumo, la batería se descarga haciendo frente al exceso de demanda de corriente. En las horas de consumo reducido, es el generador el que atiende la misma y simultáneamente proporciona corriente a la batería para que se cargue.

6.5.3 Reguladores

El sistema de regulación tiene básicamente tres funciones (Bioclimatic web, 2007): evitar sobre cargas y descargas profundas de las baterías⁷⁸, impedir la descarga de la batería a través de los paneles en los periodos sin luz, y asegurar que el sistema trabaje siempre en el punto de máxima eficiencia. En definitiva, el regulador constituye uno de los elementos más importantes de un sistema solar fotovoltaico ya que de su correcto funcionamiento depende totalmente la vida de la batería.

La forma de evitar la sobrecarga es desconectar las baterías de los paneles. Esta desconexión se puede realizar de dos formas, que dan lugar a los reguladores tipo serie y tipo paralelo:

- *Regulador tipo serie.* Los reguladores serie realizan la función de desconectar el panel de las baterías cuando se logra el estado de plena carga.

78 Éstas pueden ocasionar daños irreversibles en las baterías.

- *Regulador tipo paralelo.* Estos reguladores detectan la tensión en los bornes de la batería y cuando el potencial alcanza un valor preestablecido crean una vía de baja resistencia para el grupo solar, derivando con ello la corriente y apartándola de las baterías.

El regulador en paralelo tiene un rendimiento inferior a su equivalente en serie. Este hecho es el responsable de que el regulador en serie sea el más empleado, relegando el uso de reguladores en paralelo para aplicaciones de niveles energéticos muy bajos en los que no es importante un pequeño dispendio de energía y en aquellas en las que es muy importante la interacción entre equipos, o sus secciones, alimentados por la misma fuente primaria. Sin embargo, desde el punto de vista de la protección del equipo, en caso de sobrecarga los reguladores en serie reciben el impacto directo, lo que les hace ser mucho más frágiles.

6.5.4 Inversores

Son elementos capaces de alterar la tensión y características de la corriente eléctrica que reciben, transformándola de manera que resulte más apta para los usos específicos a que vaya destinada en cada caso. Atendiendo a su relación con la red eléctrica, los inversores se clasifican en (Ibañez *et al.*, 2004):

- a) *Inversores autónomos.* También denominados *inversores estáticos*, están diseñados para funcionar en sistemas sin conexión con la red pública de distribución, obteniendo la energía de un sistema de acumuladores. El nivel de prestaciones y requisitos de estos inversores es muy inferior al segundo tipo de inversores.
- b) *Inversores conectados a red o inversores síncronos*, se utilizan en instalaciones conectadas a la red de distribución de una compañía eléctrica. Los inversores síncronos deben producir CA sincronizada con la CA de la red a la que está interconectada, es decir, ambas deben tener idénticas secuencias de fases, tensión y frecuencia que la red eléctrica y, además, las formas de las ondas (de la red y del inversor) deben estar en fase. En principio, estos sistemas no necesitan de un subsistema de almacenamiento (pues toda la energía generada se entrega a la red de distribución), a menos que se requiera una alimentación de emergencia.

6.5.5 Otros elementos

Además de estos elementos característicos de la mayoría de las instalaciones fotovoltaicas, existen varios accesorios que, aunque no estrictamente imprescindibles, son recomendables para una mayor seguridad o control de la instalación. A continuación se citan los más característicos.

- a) *Alarmas y desconectores de bajo voltaje.* Si por circunstancias imprevistas, o debido a un inadecuado dimensionado, la batería se descarga hasta un nivel peligroso resulta conveniente instalar un dispositivo que avise al usuario mediante una alarma o bien desconecte la batería del consumo, aún a costa de interrumpirlo, hasta que se haya recuperado un nivel mínimo.
- b) *Programadores horarios.* Son equipos ampliamente utilizados en instalaciones convencionales, existiendo también para corriente continua. Son útiles, por ejemplo, en instalaciones que alimentan señales luminosas que deben encenderse a partir de una cierta hora.
- c) *Sistemas híbridos.* La combinación de un sistema solar con un generador (un sistema híbrido) podría ser una alternativa económica en vez de invertir en un sistema solar de gran autonomía. Así el sistema de energía solar no tiene que ser tan grande para que cubra la necesidad de potencia durante las horas picos y/o una gran autonomía para días con mal tiempo. Los paneles solares pueden complementarse con el empleo de aerogeneradores (los paneles suministran energía los días despejados mientras que en los días fríos y ventosos, frecuentemente nublados, serían los aerogeneradores la principal fuente de energía) o de grupos electrógenos de apoyo cuando la potencia doméstica instalada sea elevada.

6.5.6 Conexión a la red

Haciendo uso de los distintos componentes analizados podemos disponer fácilmente de una instalación fotovoltaica sin necesidad de conectarla a la red. Esta opción suele ser apropiada para lugares donde (Energy web, 2007) no hay red eléctrica (porque el área está muy alejada de las centrales eléctricas), la red eléctrica no es fiable (por los posibles cortes de energía ante una emergencia), o la conexión a la red es demasiado cara (debido al precio de la extensión de las líneas). Los sistemas fotovoltaicos independientes también son excelentes si no se necesita mucha energía.

Sin embargo, la conexión a la red tiene muchas ventajas económicas así como ambientales. En efecto, al disponer de una red eléctrica se puede utilizar un sistema de energía solar para producir parte de la energía que se necesita y utilizar energía de la red durante la noche y en días muy nublados. De esta forma se reduce la cantidad de electricidad que se consume de la red cada mes y no se consume ningún combustible ni se genera contaminación.

También se puede vender la electricidad solar que no se utilice a la compañía eléctrica: cuando se necesita más electricidad de la que genera el sistema fotovoltaico (por ejemplo, por la noche), se utiliza electricidad de la red, mientras que cuando se utiliza menos electricidad de la que produce el sistema fotovoltaico el exceso se inyecta (o vende) a la red.

El transporte de la energía eléctrica se realiza a través de los distintos componentes de la instalación fotovoltaica mediante canalizaciones (líneas) eléctricas, generalmente cables realizados a partir de conductores de sección circular o agrupación de estos. Los criterios a tener en cuenta en su diseño son (Ibañez *et al.*, 2004):

- El *tipo de cable*. Debe especificarse el material conductor con el que se fabricará (cobre, aluminio), si es rígido o flexible, el tipo de aislamiento, el tipo de cubierta protectora, etc.
- El *modo de instalación del cable*. Es preciso establecer si se trata de una línea aérea, subterránea o interior, si el conductor discurrirá en el interior de canalizaciones entubadas, bajo tubos de protección, en el interior de huecos de las construcciones, etc.
- La *sección de los conductores*. Se define a partir de la intensidad máxima admisible por los conductores (tanto en régimen permanente como en caso de cortocircuito) y la caída de tensión máxima admisible.

6.5.7 La distribución física y los operadores logísticos

Otro aspecto a considerar en este dominio es el movimiento físico de materiales y productos. Como sabemos, la mayoría de los nuevos modelos empresariales, y muy en particular el de la VE dinámica, dependen de una logística eficaz (Sanz, 2004). Debemos conseguir que este movimiento físico se produzca de manera eficiente y rentable, alcanzando economías de escala y dentro del nuevo orden global.

La carencia en las empresas de personal cualificado y especializado en las labores de distribución física, los riesgos y los altos costes que le suponen a una empresa productora manipular y transportar pequeñas cantidades de su producto y cubrir extensas áreas para dar servicio a un número cada vez mayor de pequeños establecimientos, etc., son, entre otras, algunas de las razones de peso para argumentar a favor de la idea de que el abastecimiento ha de ser realizado por una entidad especializada⁷⁹.

Los operadores logísticos son precisamente esos servicios especializados que asumen la distribución física actuando con mucha eficacia en este mundo globalizado. Las empresas son cada vez más conscientes de que encargar la distribución física a terceros agrega valor a los bienes. Por otro lado, la creciente subcontratación de los servicios logísticos ha puesto de manifiesto la necesidad de agilizar el intercambio de información entre los operadores logísticos y transportistas y sus clientes, con el fin de acortar ciclos de entrega, optimizar los movimientos logísticos, mejorar el nivel de notificación sobre *stocks* y el servicio al cliente, etcétera.

6.5.8 La ontología de dominio

Dado el gran número de conceptos, predicados y acciones de agentes que componen la ontología del dominio de aplicación principal, su representación gráfica en este documento mediante un diagrama de clases UML resulta desaconsejable. No obstante, y para facilitar una visión general de los conceptos aquí presentados, la figura 6.15 muestra la descomposición jerárquica del proceso de negocio global considerado.

Somos conscientes que tanto esta descomposición como las propiedades empleadas para definir cada uno de los procesos de nivel inferior constituyen una simplificación de la realidad (orientada a reducir el trabajo de construcción de la plataforma y validación del modelo). En ningún momento se ha pretendido dotar a la ontología de dominio de un grado de precisión absoluto, tarea que corresponde a los miembros del VBE y, muy especialmente, a su administrador. Éstos, y no nosotros, son los responsables últimos de suministrar las plantillas que describan, de manera apropiada y completa, la actividad dentro del BP global.

79 Nuestra plataforma permite ambas soluciones: que sea la propia empresa la encargada del transporte de sus productos o que delegue en un operador físico.

Nuestra labor, en este punto, consiste en crear una herramienta software que permita verificar la idoneidad del modelo de partida, y que tenga en cuenta todas las consideraciones efectuadas en el presente capítulo. Este será, precisamente, el cometido de los próximos capítulos.

6.6 RESUMEN

La plataforma JADE se ha convertido en estos últimos años en una de las herramientas más populares para el desarrollo de sistemas multiagente. Sin embargo, hasta hace bien poco no se había propuesto una metodología formal que guiase el análisis y diseño de MAS usando dicha plataforma. Esta realidad constituía una de las principales carencias para unificar la construcción de sistemas multiagente basados en esta herramienta. Por fortuna, el trabajo desarrollado por Nikraz *et al.* (Nikraz *et al.*, 2006) ha permitido subsanar esta laguna.

Siguiendo sus guías y recomendaciones, a lo largo de este capítulo hemos realizado el análisis y diseño de nuestra plataforma con el fin de poseer la base e ideas necesarias para progresar hacia su construcción (y su posterior validación). Pero antes de llevar a cabo este análisis y diseño, hemos realizado algunas consideraciones previas sobre la funcionalidad que debía poseer nuestra aplicación.

Estas consideraciones nos han permitido sumergirnos en el análisis de la plataforma y clarificar el problema a un nivel de detalle suficiente, identificando los requerimientos de la aplicación, a través de 6 pasos. El resultado de esta fase han sido los diagramas de casos de uso, el diagrama de agentes, la tabla de responsabilidades y el diagrama de despliegue de agentes, que convenientemente refinados constituyen la base para la fase del diseño.

Y es precisamente esta fase, la de diseño, la que ha de permitirnos que se alcance el nivel de detalle suficiente como para tener una transición relativamente directa a la construcción de la plataforma, es decir, a su codificación real. Esto lo podemos hacer porque a lo largo de esta fase se han especificado aspectos como las interacciones y los protocolos de interacción, las plantillas de los mensajes, la descripción de los servicios a ser registrados/buscados en el agente DF, las interacciones agente-recurso y agente-usuario, los comportamientos internos de los agentes, la definición de las ontologías y la selección del lenguaje de contenido.

Antes de finalizar el capítulo, conviene hacer constar que el mismo no se ha elaborado con la pretensión de ser totalmente completo y minucioso a la hora de plasmar todo el trabajo realizado y todos los artefactos generados (diagramas, tablas, etc.). Este comportamiento nos hubiese llevado a representar una cantidad ingente de información, en su mayor parte innecesaria para nuestros propósitos. Más bien, el capítulo ha pretendido ser un resumen de aquellos aspectos que permiten representar mejor las decisiones y actuaciones que han definido nuestro trabajo durante este tiempo.

Tras completar con éxito los pasos del análisis y del diseño, estamos en disposición de abordar la construcción de la aplicación multiagente usando la plataforma JADE y el sistema experto JESS.

CAPÍTULO

7

CONSTRUCCIÓN DE LA PLATAFORMA

7.1 INTRODUCCIÓN

El modelo propuesto en el capítulo cuarto ha de servir de base en la construcción de aplicaciones software capaces de dar una respuesta realista e innovadora al problema de selección de socios en las DVE. Por consiguiente, para verificar su idoneidad resulta imprescindible construir y validar una de esas aplicaciones. Durante los capítulos previos se ha justificado el uso de la tecnología multiagente y de los sistemas expertos, se han seleccionado las herramientas a emplear y se ha efectuado su análisis y diseño. Ha llegado el momento, por tanto, de centrarnos en su construcción.

Con este objeto nace el presente capítulo, donde se dan cabida los aspectos más interesantes de la plataforma desarrollada. Aunque la descripción se centra principalmente en la visión de los usuarios, también se presentan ejemplos ilustrativos del código generado (JAVA, XML o JESS) con el fin de favorecer una mejor comprensión del trabajo realizado en este tiempo. Pero eso será en la segunda parte del capítulo, dado que en la primera se describen, de forma breve, algunas características básicas de las herramientas software empleadas en su construcción.

7.2 BREVE DESCRIPCIÓN DE LAS HERRAMIENTAS EMPLEADAS

En este apartado se presentan algunos aspectos básicos de aquellas herramientas que han permitido el desarrollo de la aplicación multiagente que aquí se describe. Representa, en definitiva, un breve resumen de los manuales elaborados durante este tiempo en los que se han recogido los detalles sobre estas tecnologías y su forma de utilización.

7.2.1 Una visión panorámica

La figura 7.1 muestra una visión panorámica de las herramientas y *plugins* empleadas en la construcción de nuestra plataforma multiagente.

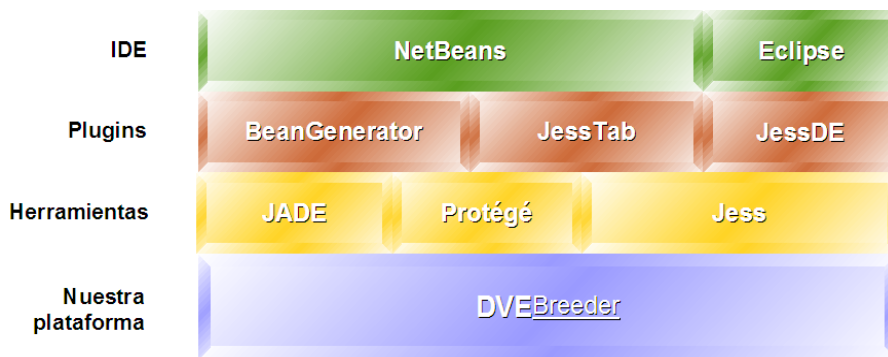


Figura 7.1. Visión general de las herramientas de desarrollo.

7.2.2 JADE

JADE es un entorno software eficiente y robusto para la creación de sistemas de agentes distribuidos. Fue desarrollado en Italia como fruto de la colaboración entre el CSELT⁸⁰ (*Centro Studi e Laboratori Telecomunicazioni*) y el Computer Engineering Group de la Universidad de Parma, y desde entonces su popularidad y utilización han ido en aumento debido, sobre todo, a su estricta conformidad con el estándar FIPA.

80 Actualmente CSELT se conoce como laboratorio de Telecom Italia (*tilab*).

JADE simplifica la creación de sistemas multiagente, proporcionando un *middleware* que cumple con las especificaciones FIPA y varias herramientas gráficas para facilitar su desarrollo y depuración. Está implementado completamente en lenguaje Java, lenguaje que también es el escogido para la programación de sus agentes, pues JADE no ofrece ningún lenguaje específico de agentes, sino más bien un conjunto de clases y herramientas de los que hacer uso para programar sus agentes.

Sus características principales son: una buena gestión de la plataforma y del intercambio de mensajes (tanto intra como interplataforma), extensibilidad del código, facilidad de depuración, flexibilidad, gran experiencia en la implementación de grandes sistemas de agentes, los agentes JADE tienen todas las características que necesitamos (y alguna más), la programación es eficiente y tolera los fallos, el grupo de usuarios es muy activo, etc., y además es software libre.

JADE consta de un *entorno de ejecución* donde los agentes JADE pueden “vivir” y que debe estar activo en una máquina antes de que los agentes puedan ejecutarse en ella, una *librería de clases* que los programadores pueden usar para desarrollar sus agentes y sus aplicaciones multiagente, y un conjunto de *herramientas gráficas* que permiten administrar y monitorizar la actividad de los agentes activos (aquellos que están en ejecución).

LA PLATAFORMA DE AGENTES

Como sabemos, JADE responde a los estándares FIPA de organización de sistemas multiagente y de comunicación entre agentes. Esto quiere decir que las especificaciones que incluye la arquitectura abstracta de FIPA también se van a encontrar en JADE.

Según FIPA, los agentes son entidades software que están localizadas en una única plataforma. Una plataforma de agentes (figura 7.2) es un entorno de ejecución en el que los agentes “viven” y a través de la cual disponen de un *servicio de páginas blancas* para buscar otros agentes, un *servicio de páginas amarillas* para buscar servicios que otros agentes ofrecen, un *módulo de gestión* para acceder a estas facilidades y, por último, un *sistema de envío/recepción de mensajes* que se utiliza para transmitir los mensajes entre los agentes ubicados en la misma plataforma.

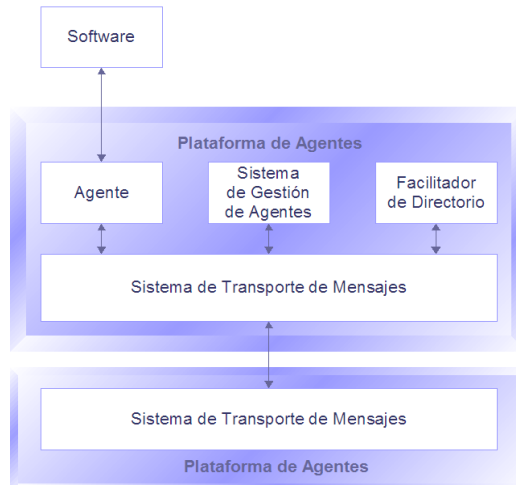


Figura 7.2. Modelo de referencia FIPA (FIPA web, 2007).

- El agente AMS (*Agent Management System*) es el encargado de supervisar y controlar tanto el acceso a la plataforma como su utilización. Proporciona los servicios de páginas blancas y de ciclo de vida de la plataforma.
- El agente DF (*Directory Facilitator*) proporciona, por defecto, el servicio de páginas amarillas: es la entidad a la que acceden los agentes para buscar los servicios que ofrecen otros agentes.
- El Sistema de Transporte de Mensajes (MTS, *Message Transport System*), también conocido como Canal de Comunicación de Agentes (ACC, *Agent Communication Chanel*), es el componente software que controla el intercambio de mensajes, tanto dentro de la plataforma como entre plataformas.

JADE se ajusta totalmente a esta arquitectura de referencia, por lo que cuando se lanza la plataforma se crean, de manera inmediata, los agentes AMS y DF, y se inicia el módulo ACC que permite el intercambio de mensajes. Pero a parte de todos estos detalles propios de FIPA, JADE incluye la noción de contenedor (*container*). La plataforma puede dividirse en varios hosts o máquinas, aunque solamente una única máquina virtual Java se ejecuta en cada host. Cada máquina virtual es, a fin de cuentas, un contenedor de agentes que proporciona un completo entorno de ejecución, permitiendo que varios agentes actúen simultáneamente (en JADE cada agente es un hilo Java).

El contenedor principal (*Main Container*) es el contenedor de agentes donde “viven” el AMS y el DF, y donde se crea el registro RMI usado internamente por JADE para la comunicación intra-plataforma. Los demás contenedores de agentes (contenedores normales) deben subscribirse al principal: la unión del contenedor principal y de los diferentes contenedores normales proporciona un entorno de ejecución completo para la ejecución de cualquier conjunto de agentes JADE.

LA LIBRERÍA JADE

JADE está compuesto por varios paquetes Java que proporcionan al programador tanto piezas funcionales listas para su utilización como interfaces abstractas adaptables a la aplicación objetivo. Para trabajar con agentes resulta imprescindible conocer la clase `jade.core.Agent` ya que para crear un agente se debe crear una nueva clase que herede de ésta y que, como mínimo, implemente el método `setup()`. Por lo tanto, desde el punto de vista del programador un agente JADE es simplemente una instancia de una clase Java (definida por el usuario) que extiende la clase base `Agent`.

Esto supone la herencia de un conjunto de características básicas que son necesarias para realizar cualquier tipo de interacción con la plataforma de agentes (entre los que se incluyen la configuración, el registro en la plataforma, la gestión remota,...) y diversos métodos a los que se puede invocar para implementar el comportamiento del agente (por ejemplo, enviar y recibir mensajes, utilizar protocolos de interacción estándar, registrarse en varios dominios, etcétera).

La clase `Agent` también proporciona varios métodos para permitir la comunicación entre agentes. De acuerdo con la especificación FIPA, los agentes se comunican a través del intercambio de mensajes asíncronos, donde los elementos intercambiados son objetos de la clase `ACLMessage`. Algunos de los protocolos de interacción definidos por FIPA también están disponibles en JADE como comportamientos *ready-to-use*; estos forman parte del paquete `jade.proto`.

En JADE cada agente es identificado mediante un identificador de agente (AID, *Agent Identifier*), representado como una instancia de la clase `jade.core.AID`. Este AID es una estructura compuesta por varios *slots*, entre los que destacan el nombre (*name*) y las direcciones (*addresses*): el

parámetro *name* es un identificador globalmente único que puede usarse como una expresión de referencia de cada agente, mientras que el *slot addresses* contiene las direcciones de la plataforma en las que vive el agente.

El modelo computacional de un agente es multi-tarea, donde las tareas (o comportamientos) se ejecutan concurrentemente. Cada servicio que presta un agente debe ser modelado e implementado como uno o más comportamientos⁸¹, es decir, como objetos `jade.core.behaviours.Behaviour`. A continuación se describen estos comportamientos:

- `OneShotBehaviour`: implementa una tarea atómica que funciona una vez y termina inmediatamente.
- `CyclicBehaviour`: para aquellas tareas que siempre están activas, y realizan las mismas operaciones cada vez que son secuenciadas.
- `TickerBehaviour`: cuando la tarea ejecuta periódicamente las mismas operaciones.
- `WakerBehaviour`: implementa una tarea atómica que funciona una vez después de una cierta cantidad de tiempo, y después finaliza.
- `ParallelBehaviour`: construye una tarea compuesta que secuencia subtareas de manera concurrente.
- `SequentialBehaviour`: implementa una tarea compuesta que secuencia subtareas secuencialmente.
- `FSMBehaviour`: para una tarea compuesta que secuencia las subtareas según una máquina de estados finita (FMS).

Los comportamientos compuestos pueden anidarse, lo que puede originar que, por ejemplo, una subtarea de `SequentialBehaviour` sea a su vez un `FSMBehaviour`, etc. En particular, todas las responsabilidades complejas que se pueden modelar como FMS pueden ser implementadas eficientemente como instancias de `FSMBehaviour`.

Finalmente, encontramos la clase `jade.domain.DFService`, que implementa un conjunto de métodos estáticos para comunicarnos con el servicio DF estándar de FIPA, es decir, con el agente páginas amarillas. En concreto, podemos solicitar acciones de registro, desregistro, modificación y búsqueda (`register()`, `deregister()`, `modify()` y `search()`) al DF.

81 La gestión del orden de ejecución de estos comportamientos la realiza un *scheduler* interno a la clase base `Agent` y oculto al programador.

GUI PARA LA GESTIÓN Y CONTROL DE LA ACTIVIDAD DE LA PLATAFORMA

JADE incorpora un conjunto de herramientas gráficas destinadas a facilitar la siempre difícil tarea de depuración de las aplicaciones multiagente. Cada una de estas herramientas es un agente que obedece las mismas reglas, posee las mismas capacidades de comunicación y el mismo ciclo de vida que un agente cualquiera de una aplicación genérica.

- El agente *RMA (Remote Monitoring Agent)* permite controlar el ciclo de vida de la plataforma y de todos los agentes registrados (figura 7.3.). La arquitectura distribuida de JADE permite también el control de un *host* remoto.

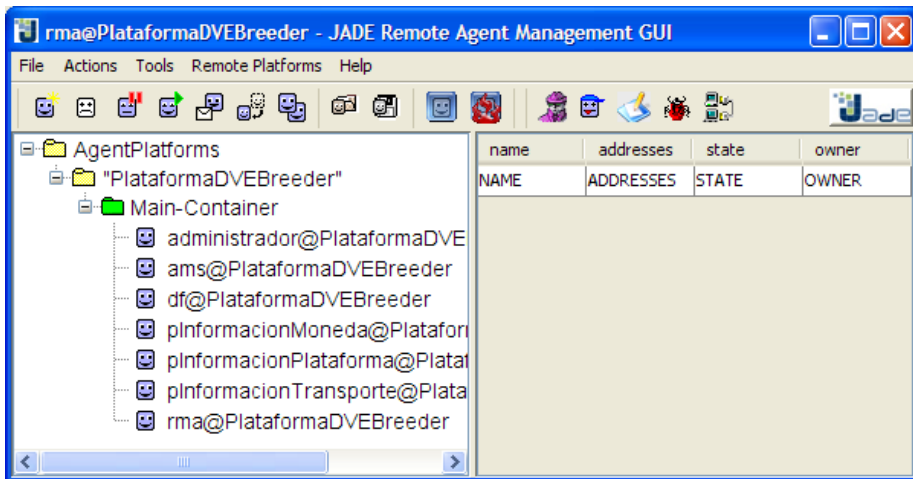


Figura 7.3. Interfaz gráfica del agente RMA.

- Gracias a la herramienta *DummyAgent* los usuarios pueden interactuar con los agentes JADE de una manera personalizada.
- JADE también permite visualizar la interfaz gráfica del agente DF (*DF GUT*) que se generó al iniciar la plataforma.
- El agente *Sniffer* se utiliza para comprobar cómo se produce la interacción e intercambio de mensajes entre los agentes que indiquemos.
- Por último, el agente *Instrospector* ha sido concebido para supervisar y controlar el ciclo de vida de un agente en ejecución, los mensajes que ha intercambiado, la cola de comportamientos, etc.

7.2.3 La IDE NetBeans

En junio de 2000, Sun Microsystems (SUN web, 2007) fundó el proyecto open source NetBeans (NETBEANS web, 2007). Desde entonces, y bajo el patrocinio de Sun, NetBeans se ha situado como líder en el mercado de desarrollo de software, con una gran base de usuarios, una comunidad en constante progresión, y con cerca de un centenar de sociedades en todo el mundo.

Sus productos hacen uso de modelos estándar (*Swing*), de la portabilidad de Java y del respaldo de una comunidad de apoyo⁸². Un ejemplo de este software es el IDE NetBeans, cuya interfaz gráfica aparece representada en la figura 7.4. Este IDE (*Integrated Development Environment*) es un entorno de desarrollo hecho para que los programadores puedan escribir, compilar, corregir errores y ejecutar programas. Está escrito en Java, pero puede servir de soporte a cualquier otro lenguaje de programación.

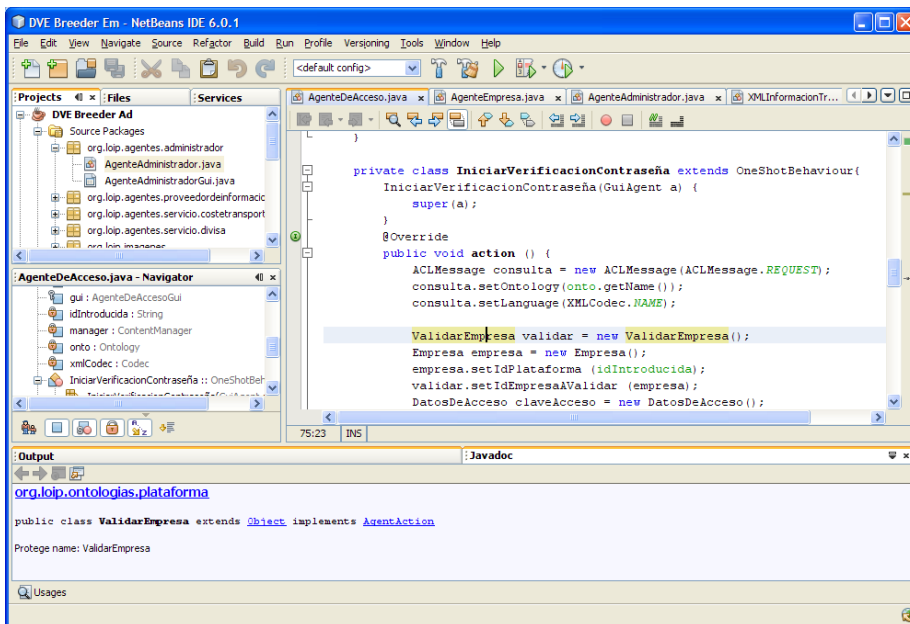


Figura 7.4. Interfaz gráfica de la IDE NetBeans.

82 El software de NetBeans es libre para el uso comercial y no comercial.

GUI BUILDER

Un aspecto básico en el desarrollo de aplicaciones es la construcción de la interfaz gráfica de usuario (GUI, *Graphical User Interface*) y en eso la actual IDE NetBeans supera significativamente a otras IDE existentes en el mercado (como es el caso de Eclipse). Esta ventaja competitiva la ha conseguido gracias a una mejora significativa del constructor de GUI (*GUI Builder*): se le ha hecho más poderosa e intuitiva, permitiendo a los usuarios construir GUI con una apariencia profesional sin necesidad de tener un conocimiento profundo de los gestores de *layout Swing*⁸³.

Este constructor soluciona el problema fundamental de la creación de GUI Java dinamizando el trabajo de creación de interfaces gráficas, y liberando así a los diseñadores de las complejidades de dichos gestores. Para lograrlo, se ha extendido el constructor de GUI para que soporte un paradigma *Free Design* con reglas de distribución fáciles de entender y utilizar. Mientras el diseñador distribuye su formulario, el constructor de GUI proporciona guías visuales que sugieren la posición y alineación óptimos de los componentes. En el fondo, el constructor de GUI traduce las decisiones de diseño a una interfaz de usuario funcional que se implementa usando el nuevo gestor de *layout GroupLayout* y otras construcciones *Swing*.

LA NATURALEZA AUTÓNOMA DE LOS AGENTES FRENTE A LA NATURALEZA REACTIVA DE LAS GUI

Cualquier aplicación, esté estructurada o no como un sistema multiagente, necesita interactuar con sus usuarios. Así pues, a menudo es necesario proveer una GUI para algunos agentes de la aplicación. Esta necesidad suscita algunos problemas que se derivan de la incompatibilidad entre la naturaleza autónoma de los agentes y la naturaleza reactiva de las interfaces de usuario gráficas habituales. Cuando se utiliza JADE, el modelo de concurrencia *thread-per-agent* (hilo por agente) de los agentes JADE debe trabajar con el modelo de concurrencia *Swing*.

JADE proporciona varias formas de abordar el problema, aunque la más eficiente consiste en la utilización de la clase `jade.gui.GuiAgent`, creada

83 Tanto el GUI Builder de la versión actual de NetBeans como el gestor de *layout GroupLayout* son resultados del proyecto Matisse (Matisse web, 2006).

expresamente para este propósito. Se trata de una simple extensión de la clase `jade.core.Agent` cuyo método `setup()` instancia un comportamiento *ad hoc* que gestiona la cola de objetos `jade.gui.GuiEvent` que pueden recibirse de otros hilos. Este comportamiento se oculta al programador que solamente necesita implementar el código específico de la aplicación relativo a cada evento.

7.2.4 Jess

Como ya se ha comentado, Jess (JESS web, 2007) es un motor de inferencia y un *scripting language* (un lenguaje que indica al ordenador cómo ejecutar un procedimiento específico) escrito enteramente en lenguaje Java de Sun Microsystems (SUN web, 2007) por Ernest Friedman-Colina en los laboratorios nacionales de Sandia en Livermore, CA. Jess es pequeño, ligero y uno de los motores existentes más rápido.

Jess soporta el desarrollo de sistemas basados en reglas que después podrán acoplarse perfectamente a cualquier código escrito en lenguaje Java. En otras palabras, usando Jess se puede construir software Java que tenga la capacidad “de razonar” usando el conocimiento que se le ha proporcionado bajo la forma de reglas declarativas. En el mundo de los sistemas multiagente, puede ser utilizado como el componente de decisión de un agente (Cardoso, 2007), implementado de una manera declarativa tal y como puede apreciarse en la figura 7.5.

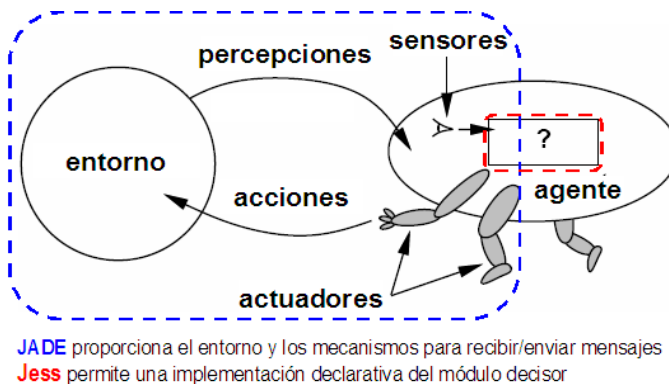


Figura 7.5. Un agente y su entorno: integración de Jess y JADE. Figura adaptada de (Russel y Norvig, 1995).

Para realizar su cometido, Jess utiliza una versión avanzada del algoritmo RETE para procesar reglas. Este algoritmo mejora la eficacia del emparejamiento de reglas en el Motor de Inferencias permitiendo, además, tanto el encadenamiento de reglas hacia adelante (emparejando hechos en memoria activa con antecedentes) como hacia atrás (emparejando hipótesis en memoria activa con consecuentes). Jess también puede manipular y razonar directamente sobre objetos Java: se pueden crear objetos, llamar a métodos, o ejecutar interfaces sin tener que compilar ningún código Java.

Para expresar el conocimiento en JESS se debe trabajar con hechos, reglas y preguntas. Existen 3 tipos de *hechos* o conocimiento: ordenados (*ordered facts*), no ordenados (*unorder facts*) y hechos conectados a objetos Java (*shadow facts*). Las *reglas* siguen la estructura *if-then* y se construyen a partir del constructor `defrule`:

```
(defrule nombreRegla (condición) => (acción))      (7.1)
```

donde *condición* (lado izquierdo de la regla) representa el conjunto de patrones que permiten encontrar los *hechos* de la *Base del Conocimiento* y *acción* (lado derecho de la regla) son llamadas a funciones. Una regla se activa cuando todas sus condiciones se cumplen y está activo el motor de inferencia. Por último, para desarrollar la *Base del Conocimiento* se necesitan preguntas⁸⁴ que permitan buscar una relación entre los hechos y las reglas que actúan en base al contenido de uno o varios hechos.

INTEGRACIÓN DE JESS Y JADE

Al ser un motor de inferencia, Jess incluye una clase especial llamada `Rete`, que implementa el motor de inferencia basado en reglas. Para embeber Jess en una aplicación Java (como por ejemplo un agente JADE), simplemente se necesita crear un objeto `jess.Rete` y manipularlo adecuadamente. Uno de los métodos incluidos en esta clase que permite activar el motor de inferencia es `Rete.run()`. Este método hará que el motor dispare consecutivamente las reglas aplicables, y regresará solamente cuando no haya más reglas que disparar, es decir, cuando el motor pare; mientras tanto el hilo que ha llamado permanecerá bloqueado.

84 Las preguntas son básicamente reglas sin parte derecha.

Para lograr la integración de Jess y JADE, nosotros hemos seguido algunas de las consideraciones realizadas por Cardoso (Cardoso, 2007), lo que nos ha llevado a embeber instancias del motor Jess dentro de los comportamientos de los agentes JADE. En concreto, y ya que buscamos que el agente pueda razonar continuamente, implementaremos varios *CyclicBehaviour* que se encargarán de activar varias instancias del motor Jess⁸⁵.

JESSDE Y ECLIPSE

JessDE es un entorno de desarrollo basado en Eclipse que posee un editor, un depurador y un visor de la red Rete.

- El editor tiene todas las características de un editor de programación moderno: coloreado de sintaxis, asistente de contenido, formateo de código automático, comprobación de errores en tiempo real, etc.
- El depurador de JessDE permite depurar un programa Jess definido en un archivo .clp. Permite, por ejemplo, suspender y reanudar un programa, ir paso a paso a través de él, etc.
- Además, se puede apreciar instantáneamente una representación gráfica de la red Rete derivada de cualquier regla.

Pero, ¿qué es Eclipse⁸⁶? Básicamente, es una plataforma de programación usada para crear IDE. Fue desarrollado originalmente por IBM (IBM web, 2007) como el sucesor de su familia de herramientas para VisualAge, aunque en la actualidad está siendo desarrollado por la Fundación Eclipse⁸⁷ (Eclipse web, 2007). En el fondo, es un armazón sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los *plugins* adecuados. La arquitectura de *plugins* permite, además, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo.

85 También establecemos el número de pasadas máximo que cualquier activación de Jess puede ser ejecutada antes de dar el control al agente. Para más información consultar el manual sobre Jess.

86 En la web oficial de Eclipse se define como “An IDE for everything and nothing in particular”.

87 Una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto. Esta fundación es independiente de IBM y, entre otras, está formado por las empresas HP, QNX, IBM, Intel, SAP, Fujitsu, Hitachi, Novell, Oracle, Palm, Ericsson y RedHat, además de algunas universidades e institutos tecnológicos.

7.2.5 Protégé

Especialmente cuando se trabaja con grandes ontologías, desarrollar a mano la clase que define la ontología (es decir los esquemas) y las clases Java que representan los predicados, las acciones de los agentes y los conceptos incluidos en la ontología, puede llegar a requerir grandes cantidades de tiempo y recursos.

Gracias al *plugin* BeanGenerator, implementado por C.J. van Aart del departamento Social Science Informatics de la Universidad de Amsterdam, es posible reducir considerablemente este proceso. Para ello debemos definir la ontología usando Protégé y después permitir que BeanGenerator cree, de forma automática, tanto la clase que define la ontología como las clases de los predicados, los conceptos y las acciones de los agentes. Este proceso es muy recomendable, ya que permite trabajar con una herramienta gráfica *ad hoc* en lugar de tener que escribir el código Java durante el proceso de definición de la ontología.

PROTÉGÉ

Protégé (Protégé web, 2007) es una herramienta gráfica desarrollada en Java por la Universidad de Stanford para el desarrollo de ontologías y bases de conocimiento. La figura 7.6 nos muestra la interfaz principal de Protégé con sus cinco pestañas estándar: Classes, Slots, Forms, Instances y Queries. Todo proyecto nuevo se abre siempre en la vista Classes, con dos clases internas de Protégé, `THING` y `SYSTEM-CLASS` (esta última para definir la estructura de algunos formularios Protégé) y ninguna instancia.

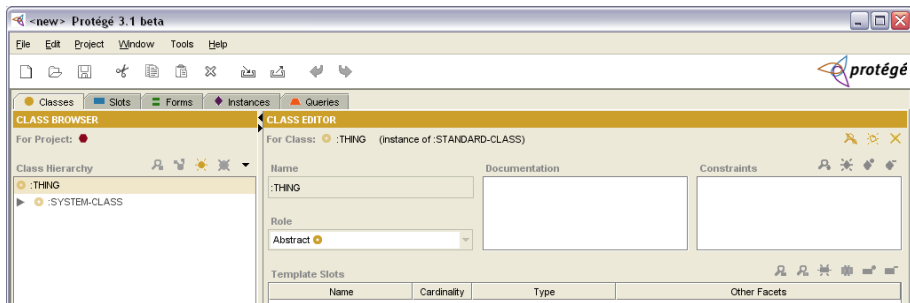


Figura 7.6. Apariencia de la ventana principal de Protégé.

Al igual que ocurre con NetBeans, en Protégé se trabaja siempre dentro de un proyecto. Una vez hayamos creado y guardado el proyecto, deberemos comenzar a construir la ontología, y para ello se han de agregar algunas clases y subclases (pestaña *Classes*), subordinadas a la clase `THING`. A continuación se agregan los *slots* o propiedades (pestaña *Slots*) para cada una de las clases y subclases creadas, y se incorporan algunas instancias⁸⁸ (pestaña *Instances*). De este modo, podemos comenzar a crear nuestra base de conocimiento.

Los formularios determinan cómo se presenta y se introduce la información de las instancias. Se puede modificar fácilmente la distribución de estos formularios gracias a las facilidades que existen en la pestaña *Forms*. Por último, la pestaña *Queries* permite que hagamos consultas a nuestro proyecto y que localicemos todas las instancias que cumplan el criterio que nosotros especifiquemos. Para crear una consulta, debemos seleccionar una o más clases, y uno o más *slots* dentro de las clases. También podemos guardar consultas en la librería de consultas para volver a llamarlas más adelante.

BEANGENERATOR

BeanGenerator (BeanGenerator web, 2007) es un *addon* de JADE implementado como un *plugin* para Protégé bajo los términos de la licencia LGPL. Esta herramienta permite generar los archivos Java que se utilizan dentro de JADE para representar ontologías. De hecho, se pueden generar ontologías conformes a FIPA/JADE a partir de proyectos RDF, XML, OWL y, por supuesto, a partir de los propios proyectos Protégé.

Para utilizar BeanGenerator han de seguirse una serie de pasos que se describen pormenorizadamente en el tutorial que hemos elaborado sobre Protégé y las ontologías. Una vez instalado dentro de Protégé, el *plugin* BeanGenerator tiene la apariencia que refleja la figura 7.7. Mediante el botón `GENERATE BEANS` se generan, a partir de la ontología que hemos creado, los archivos Java que se necesitan en JADE. Estos archivos se guardan en el directorio de destino, y dentro del paquete que hayamos especificado previamente en esa misma pestaña.

88 Las instancias son los datos reales en la base de conocimiento, por lo que es una buena idea cerciorarse de que hemos estructurado nuestro proyecto correctamente antes de empezar a introducir instancias.

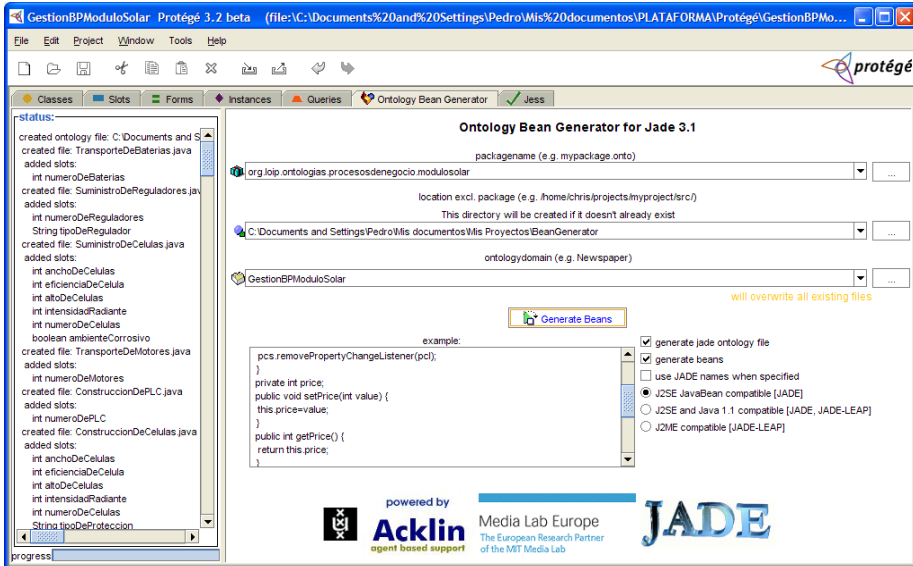


Figura 7.7. Apariencia de la pestaña Ontology Bean Generator.

JESSTAB

JessTab es un *plugin* de Protégé que proporciona una ventana de la consola Jess (figura 7.8.) a través de la que podemos interactuar con el sistema experto mientras se ejecuta Protégé (JessTab web, 2007).

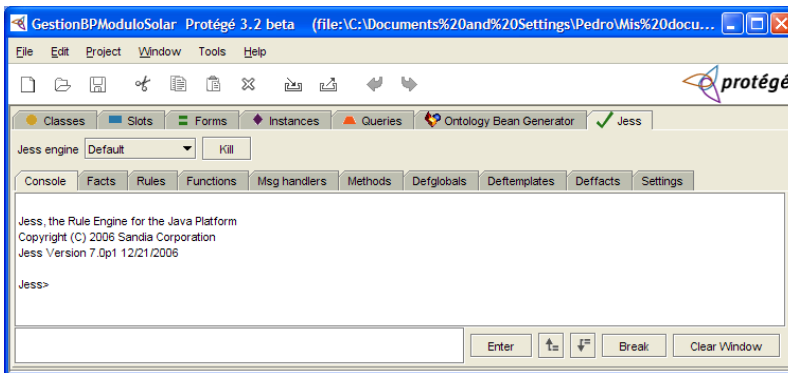


Figura 7.8. Apariencia de la pestaña JessTab.

JessTab permite convertir la información contenida en las ontologías en una estructura simple aceptada por el motor de inferencia Jess: es, básicamente, un puente entre Protege y Jess. Además, extiende Jess con funciones adicionales que permiten transformar las bases de conocimiento de Protégé a hechos Jess, e incorpora funciones para manipular las bases de conocimiento de Protégé desde Jess.

7.3 LA PLATAFORMA

La descripción de la plataforma multiagente se va a realizar desde el punto de vista de los usuarios, por lo que principalmente se mostrarán los aspectos más interesantes de las interfaces gráficas. No obstante, cuando se considere relevante también se analizará el código empleado en su construcción.

7.3.1 El agente administrador

Al ejecutar la aplicación *DVEBreeder Ad* aparece, en primer lugar, una ventana de presentación (figura 7.9). Esta ventana *splash* está diseñada para reflejar información sobre las operaciones iniciales que se producen durante el periodo de arranque de la plataforma *DVEBreeder*. Entre estas operaciones cabe destacar el lanzamiento de la plataforma JADE y la ejecución del agente administrador.



Figura 7.9. Ventana de presentación.

Como sabemos, la plataforma de agentes JADE constituye el entorno de ejecución donde nuestros agentes van a “vivir” y a través de la cual van a

disponer de diferentes servicios⁸⁹. Por tanto, debe estar activa en la máquina del administrador antes de que los agentes (entre ellos el propio agente administrador) puedan ejecutarse en ella.

En JADE existe una interfaz *in-process* que permite que las aplicaciones externas Java utilicen JADE como una especie de librería para poder lanzar el JADE *runtime* desde la propia aplicación. Se puede obtener una instancia *singleton* del JADE *runtime* gracias al método estático `jade.core.Runtime.instance()`. Este objeto *runtime* proporciona un método para crear el contenedor principal al que pasamos como parámetro un objeto `jade.core.Profile` que contiene las opciones de configuración (el *hostname*, el número de puerto del contenedor principal, etc.) requeridas para iniciar el JADE *runtime*.

El contenedor principal es el contenedor de agentes donde viven el AMS y el DF, y donde se crea el registro RMI necesario para la comunicación intra-plataforma. Es, además, el contenedor en el que residen tanto el agente administrador como el agente proveedor de información de la plataforma y los agentes de servicio. El resto de contenedores que emplearemos serán los contenedores remotos⁹⁰ donde residirán los agentes representantes de las empresas y los clientes de la plataforma, además de los agentes de acceso. La unión de estos contenedores remotos con el contenedor principal es lo que da lugar a una plataforma de agentes distribuida (véase la figura 7.10).

Volviendo a las operaciones de arranque, si se produce un error en cualquiera de ellas la aplicación se encarga de informar al administrador sobre las causas que lo han motivado. Además, y de forma inmediata, el programa finaliza su ejecución ya que todas las operaciones iniciales son fundamentales para su correcto desarrollo. En efecto, sin la adecuada ejecución de la plataforma JADE los agentes no tendrían el entorno donde vivir, pero sin los agentes de servicio, el agente proveedor de información o el propio agente administrador, la plataforma no tendría sentido, carecería de la funcionalidad con la que hemos pretendido crearla.

89 Un servicio de páginas blancas para buscar otros agentes, un servicio de páginas amarillas para buscar servicios que otros agentes ofrecen, un módulo de gestión para acceder a estas facilidades y un sistema de envío/recepción de mensajes mediante el que el sistema de gestión puede hacer llegar a su destino los mensajes generados por los agentes.

90 Éstos se crean de forma similar a la descrita para el contenedor principal.

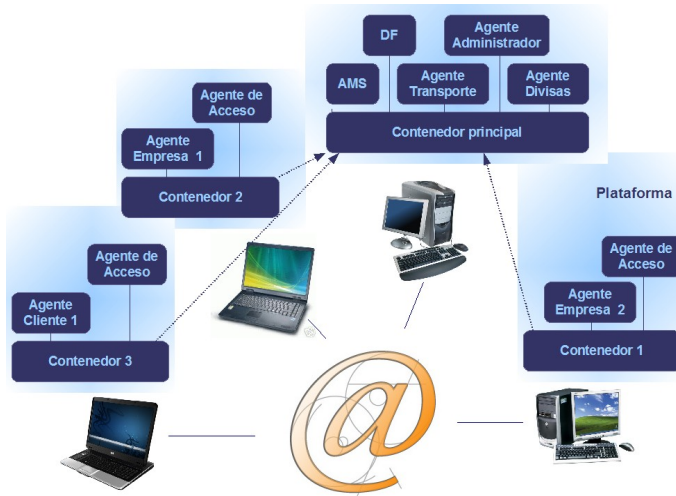


Figura 7.10. La arquitectura distribuida de nuestra plataforma.

Si no se produce ningún incidente durante el arranque de la aplicación, en pantalla aparece la interfaz de usuario del agente administrador (figura 7.11).

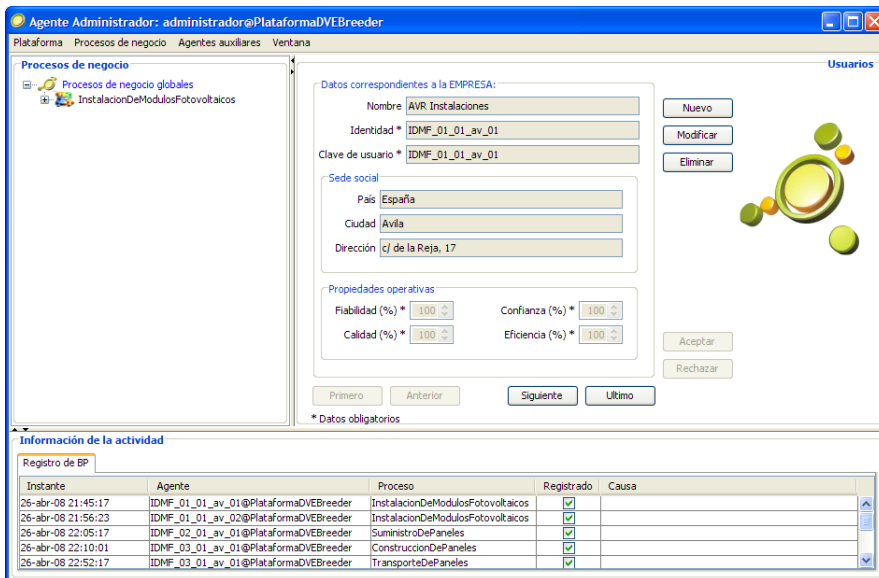


Figura 7.11. Interfaz de usuario del agente administrador.

Esta interfaz está orientada hacia la gestión de la plataforma y permite, entre otros aspectos, controlar la información de los usuarios necesaria durante la actividad de la plataforma multiagente y tener acceso a las herramientas de soporte que proporciona la plataforma JADE. En la parte superior de la interfaz se halla la barra de menús con los siguientes elementos:

- *PLATAFORMA*. Aquí encontramos las opciones relativas a la gestión y monitorización de la plataforma DVEBreeder. Entre otras acciones, podemos seleccionar la información que aparecerá en la ventana INFORMACIÓN DE LA ACTIVIDAD, o establecer el tipo de usuario (cliente o empresa) cuyos datos queremos gestionar en la ventana USUARIOS. Además, también disponemos de un submenú que permite salir de la aplicación, lo que equivale a cerrar totalmente la plataforma y a eliminar todos los contenedores (tanto el principal como los remotos) y agentes activos.
- *PROCESOS DE NEGOCIO*. Los diferentes elementos que componen este menú pretenden facilitar la navegación por el árbol de procesos de la ventana PROCESOS DE NEGOCIO. Así, gracias a ellos podemos expandir o contraer completamente el árbol (es decir, todos los procesos de negocio globales) o un BP global concreto, pero también podemos mostrar bien sólo los procesos globales existentes o bien los que dependen inmediatamente de aquellos (los procesos de primer nivel). Obtendremos idéntico resultado a través del menú emergente asociado a dicha ventana.
- *AGENTES AUXILIARES*. Este menú permite un acceso directo a varios de los agentes de soporte de la plataforma JADE. Estos agentes han sido muy útiles durante la construcción y depuración de la aplicación, y consideramos que también pueden serlo durante su uso habitual. Entre estos agentes se encuentran el RMA⁹¹, el DummyAgent, el agente Sniffer, el agente Instrospector, el agente LogManager, además de la GUI del Directory Facility de JADE.
- *VENTANA*. Contiene las opciones relacionadas con la apariencia de la interfaz de usuario. Se puede seleccionar la apariencia y comportamiento de la GUI de la aplicación seleccionado su *Look*

91 Antes de acceder a este agente, la aplicación presenta un mensaje al administrador advirtiéndole sobre el riesgo que supone acceder a él. A fin de cuentas, este agente permite cerrar la plataforma JADE, lo que dejaría a nuestros agentes sin el entorno de ejecución que precisan para “vivir”: este hecho se traduciría, en definitiva, en un mal funcionamiento de nuestra plataforma DVEBreeder.

and Feel: por defecto está activa la opción `SystemLookAndFeel`, pero se puede optar por cualquiera de los otros *L&F* disponibles (`WindowsLookAndFeel`, `MotifLookAndFeel`, etc.). También encontramos tres *Check Menu* vinculados a las tres ventanas de la interfaz, de forma que si un *Check Menu* está activado la ventana asociada aparecerá en pantalla.

El menú de agentes auxiliares no permite el acceso a la interfaz de los agentes de servicio (tanto el de divisas como el de transporte), por la sencilla razón de que éstos no la poseen. Como se recordará, estos agentes fueron creados teniendo presente que la forma más fiable y eficiente de conseguir que suministraran información continuamente “actualizada” consistía en lograr que ambos accediesen a los datos que necesitaban a través de portales web especializados. Por ello, la información asociada al transporte y a los tipos de cambio debe ser actualizada fuera de las funcionalidades de la plataforma no siendo, por tanto, labor del administrador.

Pero prosiguiendo con la descripción de la GUI del administrador, por debajo del menú encontramos varias ventanas donde se muestra información sobre los procesos de negocio, los usuarios y la actividad de la plataforma. En concreto, en la parte izquierda de la interfaz aparece representada, en forma de árbol, la estructura jerárquica que configuran los diferentes procesos de negocio que pueden registrarse en la plataforma; es la ventana `PROCESOS DE NEGOCIO`. Podemos acceder a la información más relevante de cada uno de estos procesos (incluido sus proveedores) pulsando dos veces sobre ellos; la información aparecerá inmediatamente en una pestaña de la ventana `INFORMACIÓN DE LA ACTIVIDAD` (en la parte inferior de la pantalla).

En la parte derecha se encuentra la ventana `USUARIOS` desde donde podemos gestionar la información asociada a las empresas y los clientes pertenecientes a la plataforma. Entre esta información destaca la relacionada con el acceso a la plataforma: la *identificación* del usuario y la *clave* de acceso. En nuestro caso, para construir la ID de los usuarios nos hemos basado en la estructura de procesos considerada en el dominio de aplicación (para más información consúltese el capítulo 8), de forma que todos los usuarios tienen un orden relativo al mismo.

En el caso de las empresas posee gran importancia la información vinculada a su actividad, lo que se refleja a través de las propiedades operativas. En concreto, se ha considerado que fiabilidad, confianza, calidad y eficiencia son

las variables que los usuarios pueden incorporar a su sistema de decisión (basado principalmente en los valores de coste y tiempo de suministro del servicio) para seleccionar el proveedor del proceso de negocio. Es preciso, por tanto, que toda esta información se vaya actualizando a medida que la plataforma desarrolle su actividad normal.

Todos estos datos de los usuarios se almacenan en un archivo XML, XMLInformacionUsuarios.xml (figura 7.12), al que se accede a través de la clase traductora XMLInformacionUsuariosScanner.java que proporciona los métodos necesarios tanto para gestionar dicha información como para navegar por ella (crear un nuevo usuario, modificar o eliminar uno existente, desplazarse entre los usuarios, etc.).

```
<empresa>
  <nombre>AVR Instalaciones</nombre>
  <idPlataforma>IDMF_01_01_av_01</idPlataforma>
  <clave>IDMF_01_01_av_01</clave>
  <ubicacion>
    <pais>España</pais>
    <ciudad>Avila</ciudad>
    <direccion>c/ de la Reja, 17</direccion>
  </ubicacion>
  <propiedadesOperacionales>
    <fiabilidad>100</fiabilidad>
    <confianza>100</confianza>
    <calidad>100</calidad>
    <eficiencia>100</eficiencia>
  </propiedadesOperacionales>
</empresa>
```

Figura 7.12. Ejemplo de información recogida en el archivo XMLInformacionUsuarios.xml

En la parte inferior de la GUI se encuentra la ventana INFORMACIÓN DE LA ACTIVIDAD. Inicialmente sólo se muestra una pestaña con la información sobre el registro de los procesos de negocio (REGISTRO DE BP) en la que aparece información sobre los procesos que se han registrado en la plataforma (instante, usuario que ofrece el servicio, tipo de BP, si se ha registrado correctamente y en caso negativo la causa que lo ha impedido). Esta ventana también permite desplegar otras pestañas con información sobre los procesos de negocio (al hacer doble clic sobre cualquier nodo de la ventana PROCESOS DE NEGOCIO) en la que se muestra quienes ofrecen (y, por lo tanto, quienes han registrado en la plataforma) un proceso de negocio concreto.

También podemos conocer todas las interacciones del agente administrador con los agentes de la plataforma (empresa, cliente, proveedor de información y páginas amarillas): la pestaña REGISTRO DE LA ACTIVIDAD muestra esta información, indicando tanto el momento en que se han producido como el tipo de protocolo y performativa empleada en cada caso.

7.3.2 El agente de acceso

Cuando el representante de la empresa (cliente) desea conectarse a la plataforma debe ejecutar la aplicación *DVEBreeder Em (DVEBreeder CI)*. Tras el correspondiente *splash* de presentación, en pantalla aparece la interfaz del agente de acceso (figura 7.13).



Figura 7.13. Interfaz de usuario del agente de acceso.

Este agente se encarga, como su propio nombre indica, de permitir el acceso de los usuarios a la plataforma a través de la verificación de sus datos. Con este propósito, el usuario puede introducir los datos necesarios para conectarse a la plataforma donde reside el agente administrador: la localización de la plataforma, la identidad del usuario y su clave de acceso. El administrador de la plataforma es el responsable de suministrar estos datos a todos los usuarios.

Para poder conectarnos a la plataforma es necesario introducir lo que en FIPA se conoce como *APDescription*, es decir la descripción de la plataforma. En la pantalla de acceso esta descripción se puede introducir⁹²

92 Si el agente se encuentra ubicado en la misma máquina que el AMS no es necesario introducir esta descripción.

mediante las dos formas que permite JADE; a través de la URL de la plataforma o bien mediante el AID⁹³ del AMS de la plataforma. En cuanto a los datos de acceso del usuario, estos hacen referencia a su ID en la plataforma⁹⁴ y su clave de acceso.

Tras introducir los datos correctos podemos conectarnos a la plataforma, lo que se traduce en que el agente acceso establece una comunicación con el agente proveedor de información de la plataforma para comprobar la identidad del usuario que quiere acceder a los recursos de la plataforma. Si se produce un error en la conexión o si los datos del usuario son incorrectos⁹⁵ el programa avisa con un mensaje.

Si tanto el *APDescription* de la plataforma como los datos de acceso del usuario han sido correctamente introducidos, el propio agente de acceso se encarga de iniciar la ejecución del agente empresa (o del agente cliente dependiendo del tipo de usuario que estemos considerando) antes de finalizar su propia actividad.

7.3.3 El agente empresa

Tras ejecutar el constructor del agente empresa, a éste se le da un identificador (a partir del *nickname*) que se emplea para registrarle en el AMS de la plataforma. Tras establecer la ontología y el lenguaje de contenido que contextualizarán los procesos comunicativos, en pantalla aparece su interfaz de usuario, aquí representada en la figura 7.14. Dicha interfaz está compuesta, inicialmente, por la barra de títulos, el menú de la aplicación y dos ventanas en la parte central.

En la barra de títulos se indica el tipo de agente y su AID en la plataforma. En nuestro caso, este identificador está compuesto por el *nickname* que introdujimos en el agente de acceso y el nombre de la plataforma en la que reside (HAP, *Home Agent Platform*), separados por el carácter '@'.

93 Identificador del Agente (*Agent Identifier*), también conocido como Identificador Globalmente Único del agente (GUID, *Globally Unique Identifier*). Este identificador se emplea como una expresión de referencia única del agente.

94 El ID del usuario se emplea posteriormente como el *nickname*, o nombre de pila local, del agente en la plataforma. El AID del agente se construye a partir de este valor.

95 La aplicación sólo permite tres intentos antes de desconectar completamente al usuario.

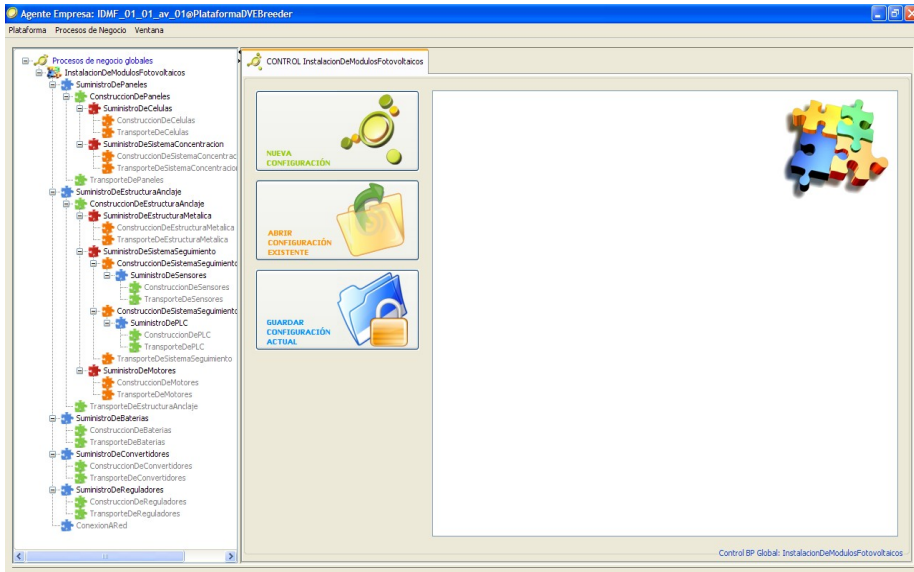


Figura 7.14. Interfaz de usuario del agente empresa.

El menú de la parte superior dispone de tres de los componentes que existían en la interfaz del agente administrador (PLATAFORMA, PROCESOS DE NEGOCIO y VENTANA) con elementos y funcionalidades similares, salvo para el caso del menú plataforma.

- En el primer menú se hallan las opciones asociadas a la actividad del agente en la *PLATAFORMA*. Predominan, sobre todo, las relacionadas con la información que se genera cuando el agente interactúa con el resto de los miembros de la plataforma. Podemos conocer, por ejemplo, los procesos de negocio que el agente ha registrado, las solicitudes de otros agentes, el estado de las negociaciones, los proveedores que han sido seleccionados para cada configuración de control, etc. Toda esta información se muestra en la ventana INFORMACIÓN DE LA PLATAFORMA y es almacenada en los respectivos archivos XML. Además, desde este menú podemos “matar” al agente empresa y eliminar de la plataforma el contenedor en el que reside. También dispone de los elementos necesarios tanto para la gestión de la configuración de control (guardar la actual, abrir una existente e iniciar una nueva) como para iniciar el proceso de búsqueda y selección de proveedores.

- El menú *PROCESOS DE NEGOCIO* es similar al menú homónimo asociado al agente administrador, y se ha diseñado para que el usuario logre una navegación más fluida entre los diferentes procesos de negocio que aparecen representados en la vista *PROCESOS DE NEGOCIO*.
- *VENTANA*. Contiene las opciones relacionadas con la apariencia de la interfaz del agente. Por un lado, podemos seleccionar la apariencia y comportamiento de la GUI seleccionado su *L&F*, pero también podemos controlar si las ventanas que componen la interfaz son visibles o no⁹⁶ a través de tres *Check Menu* asociados (por defecto sólo la ventana *INFORMACIÓN DE LA PLATAFORMA* no está activa en la GUI del agente empresa).

En la parte principal de la GUI (por debajo del menú) se sitúan las ventanas que componen la interfaz. Aunque en realidad son tres, inicialmente sólo se muestran las ventanas *PROCESOS DE NEGOCIO* y *GESTIÓN DE PROCESOS*. La primera de estas ventanas se emplea para representar, mediante una estructura en forma de árbol, tanto los procesos de negocio globales como los distintos procesos y subprocesos que los componen. Tal y como puede distinguirse en la figura 7.14, los nodos que representan procesos de negocio finales (aquellos que no requieren de la ejecución de un proceso remoto) aparecen coloreados en gris, mientras que el resto se representan en negro.

La otra ventana principal, *GESTIÓN DE PROCESOS*, tiene por objeto albergar los elementos necesarios para la visualización y gestión de la información vinculada a los distintos procesos de negocio. Inicialmente sólo aparecen en ella las pestañas de control de los diferentes procesos globales (en nuestro caso sólo existe un BP global, *InstalacionDeModulosFotovoltaicos*, cuya pestaña se mostraba en la figura 7.14). Estas pestañas de control están diseñadas para mostrar y gestionar gráficamente las configuraciones de control que el usuario establezca, tanto con el rol cliente como con el rol empresa. También nos permiten iniciar el proceso de búsqueda y selección de proveedores (sólo para el rol cliente).

Pero la ventana *GESTIÓN DE PROCESOS* también puede alojar las pestañas asociadas a cada uno de los procesos de negocio definidos en la plataforma. Así, cada vez que el usuario presiona sobre uno de los nodos de la ventana *PROCESOS DE NEGOCIO*, en la ventana *GESTIÓN DE PROCESOS* aparece una nueva

96 Se puede conseguir este mismo comportamiento gracias a los triángulos que aparecen en los *split panes* que separan las distintas ventanas.

pestaña en la que se puede gestionar la información asociada al proceso de negocio seleccionado (los hechos y las reglas que guiarán la actividad del agente en torno a ese proceso), registrar el BP en la plataforma (rol empresa) o seleccionar el tipo de control que se desea (rol cliente).

Como ejemplo, consideremos que el usuario selecciona el nodo asociado al proceso de negocio *InstalacionDeModulosFotovoltaicos*. De manera inmediata emerge la pestaña homónima en la ventana GESTIÓN DE PROCESOS, tal y como aparece representado en la figura 7.15.

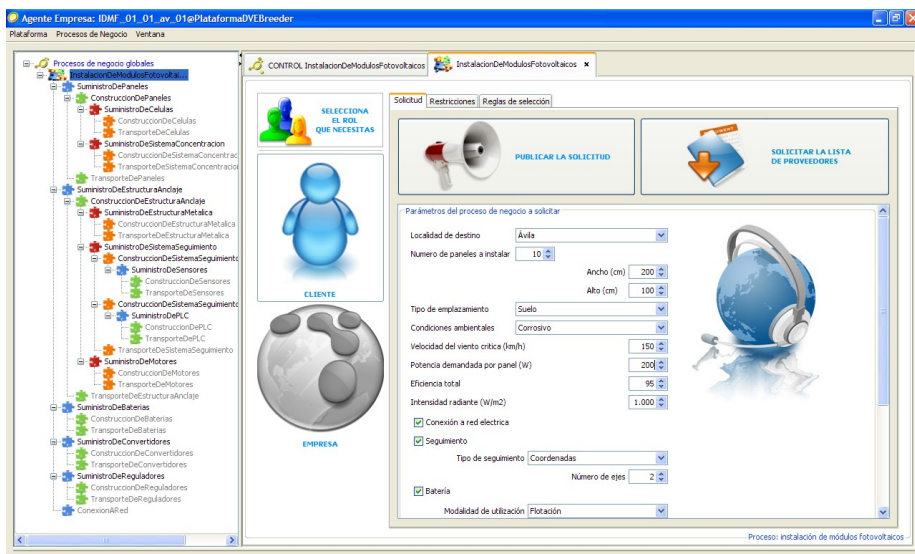


Figura 7.15. GUI del agente empresa tras seleccionar el proceso de negocio *InstalacionDeModulosFotovoltaicos*.

Los iconos de la parte izquierda de la pestaña permiten al usuario seleccionar el rol que desea para el proceso de negocio particular: cliente o empresa. Por defecto, aparece seleccionada la opción cliente ya que es, sin duda, el rol que más se emplea al establecer cualquier configuración de control. En efecto, normalmente la empresa sólo registra en la plataforma uno o muy pocos procesos de negocio (en estos casos actúa con el rol empresa) pero cada vez que lo hace debe establecer cómo desea que sea el suministro de aquellos procesos de negocio remotos que necesita (rol cliente). Además, una empresa miembro de la plataforma DVEBreeder puede iniciar en cualquier momento la formación de una VE dinámica (de nuevo, rol cliente).

Dado que el rol de cliente es el que aparece por defecto, vamos a comenzar explicando las pestañas asociadas al mismo: SOLICITUD, RESTRICCIONES Y REGLAS DE SELECCIÓN. La primera de estas pestañas, SOLICITUD (figura 7.16), contiene los elementos necesarios para seleccionar el tipo de servicio que queremos recibir: la localidad donde instalar los paneles, su número, dimensiones, el tipo de emplazamiento, etc.

The screenshot shows the 'Solicitud' (Request) tab of a business process application. At the top, there are three navigation tabs: 'Solicitud', 'Restricciones', and 'Reglas de selección'. Below these are two main action buttons: 'PUBLICAR LA SOLICITUD' (Publish the Request) with a megaphone icon, and 'SOLICITAR LA LISTA DE PROVEEDORES' (Request the List of Providers) with a document and arrow icon. The main area is titled 'Parámetros del proceso de negocio a solicitar' (Business process parameters to request) and contains a form with the following fields:

- Localidad de destino:
- Numero de paneles a instalar:
- Ancho (cm):
- Alto (cm):
- Tipo de emplazamiento:
- Condiciones ambientales:
- Velocidad del viento critica (km/h):
- Potencia demandada por panel (W):
- Eficiencia total:
- Intensidad radiante (W/m2):
- Conexión a red electrica
- Seguimiento
 - Tipo de seguimiento:
 - Número de ejes:
- Batería
- Modalidad de utilización:

On the right side of the form, there is a 3D illustration of a globe wearing a headset, symbolizing global communication or customer support.

Figura 7.16. Pestaña SOLICITUD del proceso de negocio *InstalacionDeModulosFotovoltaicos*.

Los dos botones de la parte superior, PUBLICAR LA SOLICITUD y SOLICITAR LA LISTA DE PROVEEDORES, permiten establecer el tipo de control que deseamos ejercer sobre el proveedor del servicio. Son, en definitiva, los encargados de plasmar en la aplicación las dos opciones de selección consideradas en el modelo definido en el capítulo cuarto. La elección de uno u otro botón tiene, por tanto, implicaciones diferentes en la solución final.

De acuerdo con dicho modelo, si se selecciona la opción PUBLICAR LA SOLICITUD el proveedor del proceso de negocio se seleccionará de entre las distintas DVE que hayan respondido a nuestra necesidad. En la elección de la VE dinámica se considerarán, además de las características establecidas para el servicio demandado, principalmente el coste y el tiempo de suministro⁹⁷. No se tendrán en cuenta otros parámetros operativos del proveedor (fiabilidad, calidad, etc.), lo que se traduce en un tipo de control más laxo sobre el proveedor.

En nuestra interfaz, esta elección tiene consecuencias directas en la pestaña CONTROL del proceso de negocio global *InstalacionDeModulosFotovoltaicos*, donde aparecen dos nuevos iconos: uno en representación de la empresa en su rol de solicitante (cliente) del BP global y otro representando a la DVE que se encargará de proveer dicho proceso (figura 7.17).

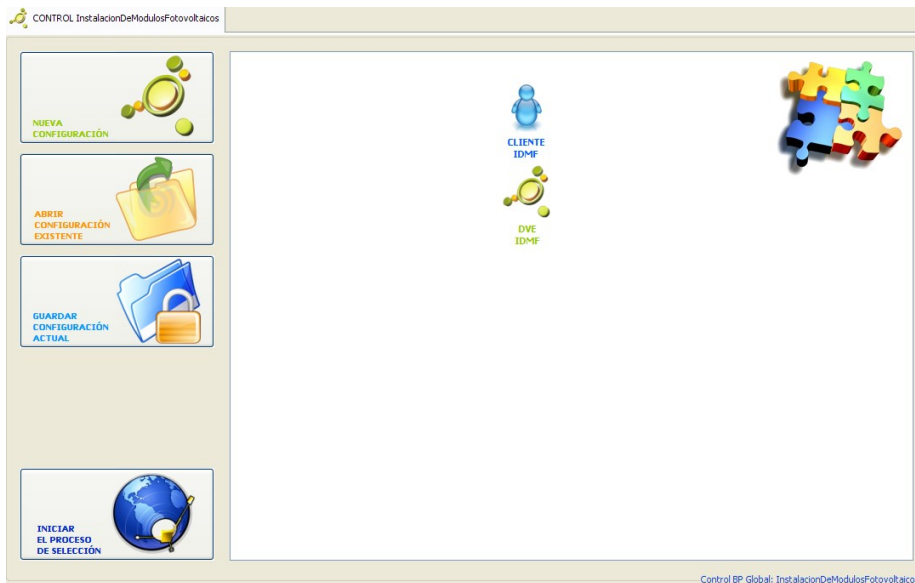


Figura 7.17. Pestaña CONTROL: INSTALACIONDEMODULOS FOTOVOLTAICOS” tras presionar el botón PUBLICAR LA SOLICITUD del BP *InstalacionDeModulosFotovoltaicos*.

97 Aunque inicialmente estos han sido los factores críticos elegidos para su elección, la utilización del sistema experto permite incorporar fácilmente nuevos criterios o modificar las reglas que guían la selección.

Al mismo tiempo, y como se habrá podido apreciar en la figura anterior, se ha hecho visible el botón INICIAR EL PROCESO DE SELECCIÓN que, como su propio nombre indica, permite comenzar el proceso de selección del proveedor en la plataforma. En el caso que nos ocupa, al presionar este botón se envía la solicitud al agente administrador, a través de un protocolo *Request*, para que éste lo distribuya automáticamente a los miembros de la plataforma que tienen registrado el proceso de negocio considerado.

Pero nuestro modelo también incorpora la posibilidad de solicitar la lista de los potenciales proveedores con el fin de seleccionarlos de una forma más exigente que tenga en cuenta, entre otros aspectos, sus parámetros operativos. Volviendo a la figura 7.16, esta posibilidad está asociada al botón SOLICITAR LA LISTA DE PROVEEDORES. Si se opta por esta opción, el proveedor del proceso de negocio será una empresa (y no una VE dinámica como ocurría en el caso anterior), lo que nos obligará a establecer el tipo de control sobre cada uno de los procesos de negocio que se encuentran en un nivel inmediatamente inferior al estimado inicialmente.

Obviamente, estas consideraciones tienen su reflejo en la pestaña de control del BP global. Así, en la figura 7.18 aparece la nueva configuración de control que surge tras presionar el botón SOLICITAR LA LISTA DE PROVEEDORES del proceso de negocio *InstalacionDeModulosFotovoltaicos*.



Figura 7.18. Representación gráfica de la configuración de control tras presionar el botón SOLICITAR LA LISTA DE PROVEEDORES del BP *InstalacionDeModulosFotovoltaicos*.

Como se aprecia en la imagen, el icono cliente está directamente asociado con una empresa proveedora del BP *InstalacionDeModulosFotovoltaicos*. Además, aparecen varios iconos que representan los procesos de negocio remotos: aquellos procesos que son finales aparecen representados como empresas (en la figura 7.18 el proceso *CAR - ConexionARed*), mientras que el resto vienen caracterizados por el signo de interrogación.

Nuestra obligación consistirá, por tanto, en establecer para cada uno de esos procesos “signo de interrogación” el tipo de control que queremos definir⁹⁸, de la misma manera que lo hemos venido realizando hasta ahora (rol cliente). Para facilitar esta labor, cada vez que se presiona el botón SOLICITAR LA LISTA DE PROVEEDORES en la ventana GESTIÓN DE PROCESOS aparecen las pestañas de cada uno de estos procesos remotos o de nivel inmediatamente inferior.

En la figura 7.19 se muestra un ejemplo de control de los muchos posibles.

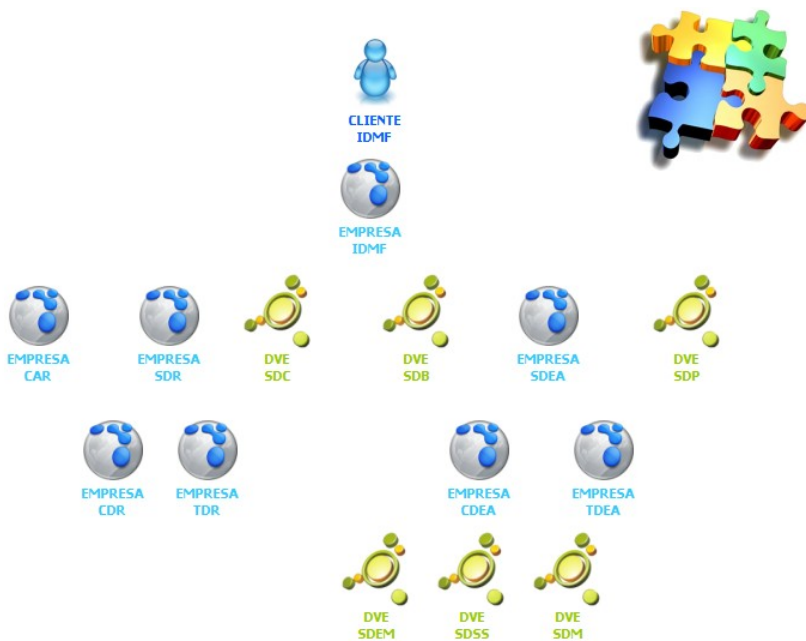


Figura 7.19. Posible ejemplo de configuración de control.

98 Mientras quede algún proceso “signo de interrogación” la aplicación no nos permitirá iniciar el proceso de selección de socios (botón INICIAR EL PROCESO DE SELECCIÓN).

La información relativa a cada configuración de control se almacena en el archivo XML_IDMFConfiguracionesDeControl.xml, que sirve de documento base para iniciar los procesos de selección de socios de cada uno de los BP que aparecen en la representación gráfica. Por este motivo, cada vez que se presiona el botón INICIAR EL PROCESO DE SELECCIÓN, y antes incluso de verificar si se ha definido un tipo de control para cada proceso de negocio representado, la aplicación nos pide que guardemos la configuración (si no lo hemos hecho previamente).

Pero nuestra labor como cliente de un proceso de negocio no se queda aquí. También debemos definir las restricciones y las reglas que guiarán la selección del proveedor: esto lo podemos hacer gracias a las pestañas RESTRICCIONES y REGLAS DE SELECCIÓN. Estas pestañas presentan idéntica apariencia gráfica, donde el componente principal es un editor de texto que muestra el documento “.clp” con la información Jess apropiada. La figura 7.20 muestra su apariencia para el caso de la pestaña RESTRICCIONES.

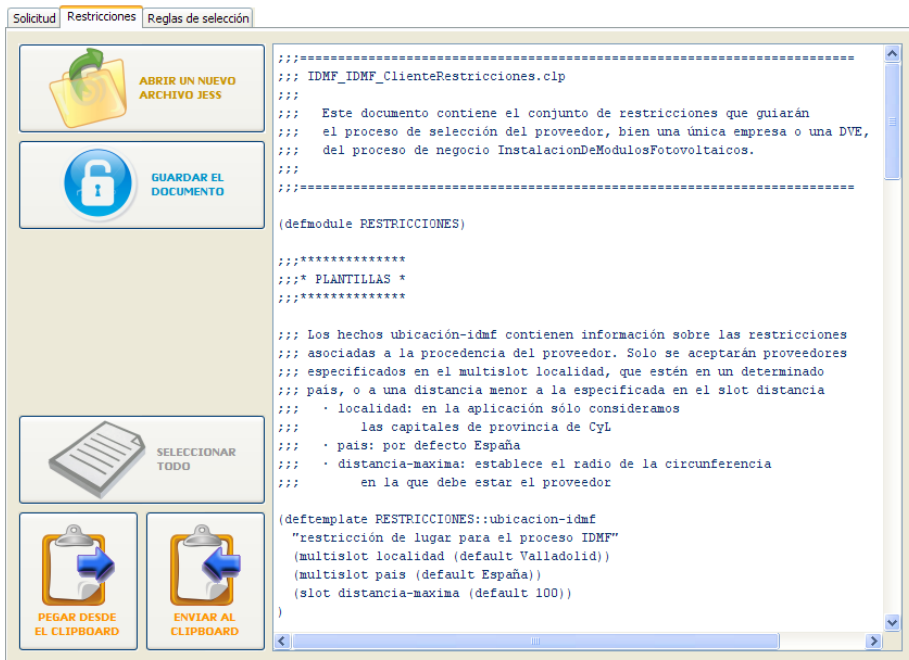


Figura 7.20. Pestaña RESTRICCIONES del proceso de negocio *InstalacionDeModulosFotovoltaicos*.

En la parte izquierda de la pestaña existen, a su vez, cinco botones⁹⁹ con los que vamos a poder realizar varias operaciones sobre el archivo Jess que aparece en el editor de texto.

- El botón ABRIR UN NUEVO ARCHIVO JESS permite mostrar en el editor un documento Jess definido previamente mediante JessDE.
- El botón GUARDAR DOCUMENTO permite almacenar los cambios realizados en el editor (incluido el que se produce cuando se carga un nuevo documento Jess). El texto representado en el editor se almacena en un documento concreto: las restricciones en el documento IDMF_IDMF_ClienteRestricciones.clp y las reglas de selección en el archivo IDMF_IDMF_ClienteReglas.clp.
- El botón SELECCIONAR TODO permite, como su propio nombre indica, seleccionar todo el documento.
- Por último, los botones PEGAR DESDE EL CLIPBOARD Y ENVIAR AL CLIPBOARD permiten, respectivamente, pegar en el documento algún texto almacenado en el *clipboard* y enviar al *clipboard* el texto que actualmente está seleccionado en el editor.

Cuando se accede a la pestaña RESTRICCIONES (figura 7.20), el editor muestra la información almacenada en el archivo IDMF_IDMF_ClienteRestricciones.clp. Este documento contiene las restricciones que utilizaremos de referencia en la selección de los proveedores del servicio, tanto si se tratan de empresas como de VE dinámicas: estas restricciones están representadas como hechos Jess. También contiene las plantillas que definen los hechos junto con su descripción.

En este punto, y antes de proseguir con la descripción de la interfaz, conviene realizar algunas consideraciones sobre los hechos Jess. Como sabemos, el conocimiento en Jess se almacena a través de los hechos; la colección de estas “pepitas de conocimiento” se conoce con el nombre de memoria de trabajo. Lo más importante que debemos saber sobre la memoria de trabajo es que las reglas sólo pueden reaccionar a adiciones, supresiones y a cambios en la memoria de trabajo. No se puede escribir una regla Jess que reaccione ante cualquier cosa. La memoria de trabajo es, por tanto, muy importante.

⁹⁹ No consideramos que sean precisos más botones para la gestión de los archivos Jess. Esta decisión se basa en la creencia de que es más acertado utilizar las características de codificación y depuración del código que ofrece el entorno JessDE, elaborado *ad hoc* para tales documentos, que un editor de texto plano.

Como se ha mencionado, cada hecho tiene una plantilla de la que obtiene su nombre y la lista de sus *slots*. Para crear estas plantillas nosotros hemos usado la construcción `deftemplate`, que es la forma más habitual y potente de crear plantillas. En la imagen 7.21 se muestra la sintaxis de esta construcción:

```
(deftemplate template-name
  ["Documentation comment"]
  [(declare (slot-specific TRUE | FALSE)
            (backchain-reactive TRUE | FALSE)
            (from-class class name)
            (include-variables TRUE | FALSE)
            (ordered TRUE | FALSE))]
  [extends template-name]
  (slot | multislot slot-name
    [(type ANY | INTEGER | FLOAT |
        NUMBER | SYMBOL | STRING |
        LEXEME | OBJECT | LONG)]
    [(default default value)]
    [(default-dynamic expression)]])*)
```

Figura 7.21. Sintaxis de la construcción `deftemplate`.

A modo de ejemplo, en la figura 7.22 se muestra la plantilla de los hechos PROPIEDADES-OPERATIVAS-IDMF. Estos hechos definen las restricciones que permiten filtrar las empresas (no las DVE) que suministran el proceso de negocio en función de sus propiedades operativas.

```
;;; Los hechos propiedades-operativas-idmf contienen información sobre los
;;; valores mínimos para las distintas propiedades operativas de las empresas
;;; que respondan a mi solicitud de provisión de negocio. Un valor de
;;; la propiedad igual a cero indica que no se tiene en cuenta en la selección
;;;   · calidad:
;;;   · fiabilidad:
;;;   · eficiencia:
;;;   · confianza:

(deftemplate RESTRICCIONES::propiedades-operativas-idmf
  "valores mínimos de las propiedades operativas"
  (slot calidad (default 0))
  (slot fiabilidad (default 0))
  (slot eficiencia (default 0))
  (slot confianza (default 0))
)
```

Figura 7.22. Plantilla del hecho PROPIEDADES-OPERATIVAS-IDMF utilizado para filtrar a los proveedores potenciales.

En Jess, hay tres clases de hechos: *ordered facts*, *unordered facts*, y *shadow facts*. La mayoría de los que hemos definido en nuestra aplicación son no ordenados (*unordered facts*) dada la potencia que poseen (la figura 7.23 ejemplifica varios hechos ordenados¹⁰⁰), pero también hemos definido algún hecho sombra¹⁰¹ (*shadow facts*) cuando se ha requerido una mayor integración entre Jess y JADE.

```

;;; La construcción mis-filtros-iniciales contiene las restricciones
;;; que se emplearán para realizar un primer filtro de las empresas
;;; proveedoras del BP InstalacionDeModulosFotovoltaicos

@ (defacts RESTRICCIONES::mis-filtros-iniciales-idmf
  (ubicacion-idmf (localidad Valladolid Burgos Salamanca)
    (distancia-maxima 200))
  (propiedades-operativas-idmf (calidad 90)
    (confianza 95))
  (empresas-vetadas-idmf (aid IMFO_00ac))
)

```

Figura 7.23. Algunas restricciones empleadas en un primer filtrado de las empresas suministradoras del BP.

Una vez definido el conjunto de restricciones, el siguiente paso consiste en establecer las reglas que las usarán para filtrar y seleccionar a nuestros proveedores de entre los disponibles. La pestaña **REGLAS DE SELECCIÓN** permite gestionar fácilmente estas reglas utilizando, para ello, la misma interfaz gráfica presentada en la pestaña **RESTRICCIONES**. En esta ocasión se muestra el archivo `IDMF_IDMF_ClienteReglas.clp` que contiene las reglas que guiarán el comportamiento del agente cuando reciba la lista de potenciales proveedores y las ofertas de otras empresas (ambas cuando el cliente ha solicitado la lista al administrador), así como las ofertas de las DVE que se constituyan (cuando el cliente ha solicitado que se publique la solicitud).

Una regla Jess es similar a la declaración *if... then (si... entonces)* de un lenguaje procedural, aunque no se utiliza de esta manera. Mientras que las declaraciones *si...entonces* se ejecutan en un momento concreto y en un orden específico, conforme a como las escribe el programador, las reglas Jess se

100 Como se puede apreciar, para insertar los hechos en la memoria de trabajo hemos empleado la construcción `defacts`. Esta construcción es, simplemente, una lista de hechos.

101 Estos hechos son solamente *unordered facts* que sirven de puente a objetos Java, de forma que permiten poner cualquier objeto Java en la memoria de trabajo de Jess.

ejecutan siempre que se satisfacen sus partes “si” (la parte izquierda de la declaración o *left-hand-sides* (LHS)), considerando que el motor de inferencia esté funcionando. Esto hace que las reglas Jess sean menos deterministas que un programa procedural típico.

En la figura 7.24 aparece la regla que permite filtrar a los agentes que representan a las empresas proveedoras.

```

;;; La regla filtrar-empresa-idmf permite eliminar de la lista de proveedores
;;; algunas de las empresas que ofrecen el BP InstalacionDeModulosFotovoltaicos.
;;; Para ello utilizamos los valores restrictivos del módulo RESTRICCIONES
;;; (recuerda -> empresa-idmf es un shadow fact)

(=defrule MAIN::filtrar-empresa-idmf
  "filtra a las empresas en función de su ubicación, propiedades operativas
  y las relaciones pasadas"
  ?he <- (RESTRICCIONES::empresa-idmf (aid ?aid) (lugar ?lugar) (distancia ?di)
      (confianza ?co) (calidad ?ca))
  (or (and (not (RESTRICCIONES::ubicacion-idmf (localidad $? ?lu $?)))
    (RESTRICCIONES::ubicacion-idmf (distancia-maxima ?dm& (< ?dm ?di))))
    (RESTRICCIONES::propiedades-operativas-idmf (confianza ?co2& (> ?co2 ?co)))
    (RESTRICCIONES::propiedades-operativas-idmf (calidad ?ca2& (> ?ca2 ?ca)))
    (RESTRICCIONES::relaciones-pasadas-idmf (aid ?aid) (prioridad NULA)))
  =>
  (retract ?he) ;si inclumple un requisito lo elimino de la lista
)

```

Figura 7.24. Ejemplo de una regla de selección del proceso de negocio *InstalacionDeModulosFotovoltaicos*.

Si se comparan los criterios de selección que hemos empleado con los criterios utilizados en otros trabajos (en especial aquellos que abordan el proceso desde un enfoque cualitativo), se comprueba que su número es bastante menor (de hecho la selección final se basa simplemente en el coste y el tiempo de suministro). Sin embargo, nuestro objetivo nunca ha sido proporcionar los múltiples criterios que se pueden utilizar en la selección (existen multitud de trabajos dedicados a ese propósito), sino suministrar una herramienta que permita acercar el proceso de selección a la forma de pensar y actuar de un decisor humano. Gracias al sistema experto Jess logramos este propósito fácilmente. Qué criterios (hechos) o qué reglas de decisión utilice el decisor en el proceso de selección dependerá de los conocimientos de cada uno de ellos, y el sistema¹⁰² se adaptará fácilmente para reflejarlos.

102 Esta es una de las labores del administrador del entorno de gestión: concretar los criterios que las empresas necesitan para tomar las decisiones de selección y trasladarlas a la aplicación multiagente.

Continuando con la descripción de la interfaz del agente empresa, el otro rol que puede ejercer es, precisamente, el de empresa. Para acceder a las opciones de este rol tan sólo deberemos presionar el botón EMPRESA (figura 7.15). De manera inmediata desaparecen las pestañas asociadas al rol cliente y en su lugar se muestran las pestañas REGISTRO y REGLAS DE CONTESTACIÓN que permiten registrar el proceso de negocio en la plataforma y gestionar tanto las características de los distintos servicios que suministra como las reglas que guiarán la contestación cuando se reciban las solicitudes de los clientes para la provisión de un servicio o proceso de negocio concreto.

La figura 7.25 muestra la primera de estas pestañas, REGISTRO, cuya apariencia es bastante similar a la mostrada para las pestañas RESTRICCIONES y REGLAS DE SELECCIÓN del rol cliente.

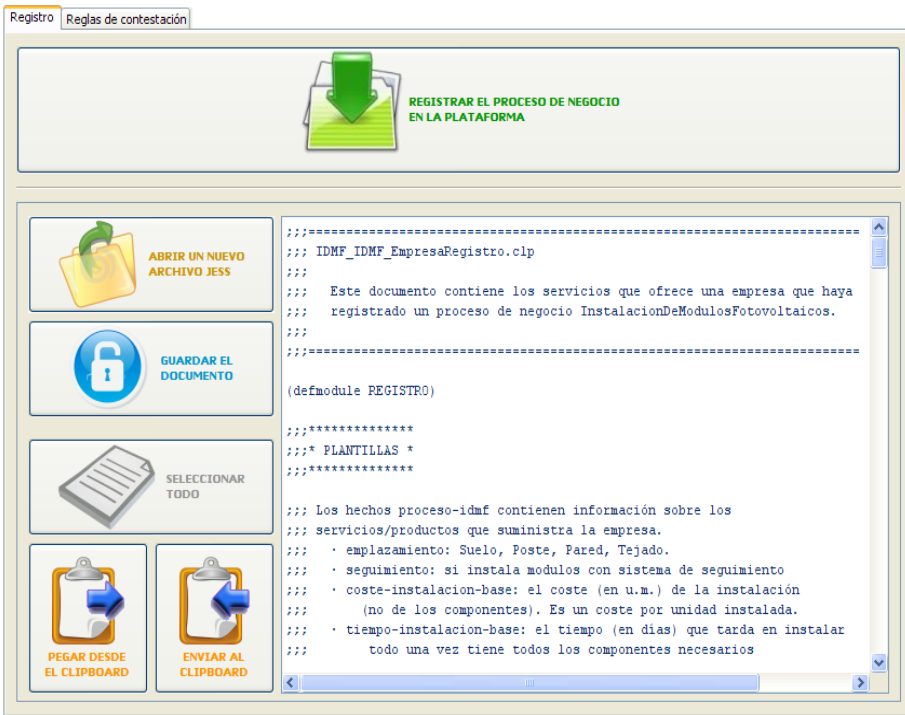


Figura 7.25. Pestaña REGISTRO asociada al proceso de negocio *InstalacionDeModulosFotovoltaicos* que aparece cuando se selecciona el rol empresa.

En la parte superior se encuentra el botón REGISTRAR EL PROCESO DE NEGOCIO EN LA PLATAFORMA que, como su propio nombre indica, permite registrar el BP a través de un protocolo FIPA *Request*, tal y como se especificó en la fase de diseño. Por debajo de este botón, en la sección derecha se muestra el archivo Jess (IDMF_IDMF_EmpresaRegistro.clp) que contiene tanto la descripción de los diferentes servicios o productos que suministra la empresa, representados como hechos Jess, como la plantilla empleada en su definición.

Tras presionar el botón REGISTRAR EL PROCESO DE NEGOCIO EN LA PLATAFORMA, tienen lugar varios acontecimientos. En primer lugar, en la pestaña de control aparece un icono representando a la propia empresa como proveedora del proceso de negocio, y varios iconos asociados a cada uno de los procesos de “nivel inferior” (figura 7.26).



Figura 7.26. Configuración de control tras registrar el proceso de negocio *InstalacionDeModulosFotovoltaicos*.

Cuando uno de estos procesos remotos requiere de la provisión de otros procesos, en la ventana de control aparece representado mediante un “signo de interrogación”, en clara alusión a la obligación de la empresa de establecer el tipo de control que desea sobre él. Por este motivo, nada más presionar el botón de registro, las pestañas asociadas a estos procesos de negocio emergen en la ventana GESTIÓN DE PROCESOS. Haciendo uso del rol cliente asignaremos el tipo de control y gestionaremos el comportamiento del agente.

Un comportamiento emergente similar se da cuando se consideran los BP *ConstruccionDe...*, aunque en este caso el propósito final diverge de forma considerable. Resulta bastante habitual que quien se dedica a fabricar un producto, también se encargue de suministrarlo e incluso transportarlo. Por ese motivo, cuando queremos registrar un proceso de negocio de esta naturaleza, al lado derecho del botón de registro aparecen dos casillas de verificación que hacen referencia a los procesos *SuministroDe...* y *TransporteDe...* asociados al proceso de negocio actual.

En estos casos, si se presiona el botón de registro emergerán, como sabemos, las pestañas de los procesos de “nivel inferior” pero, además, si estas casillas están marcadas también aparecerán las pestañas correspondientes a los BP *SuministroDe...* y *TransporteDe...* asociados. En estas pestañas, y al contrario de lo que sucede para el resto, deberemos hacer uso del rol empresa, lo que exige registrar los procesos en la plataforma y gestionar la información asociada a su provisión.

Por otro lado, cada vez que se registra un proceso de negocio la interfaz de la pestaña de registro cambia. En concreto, el botón de registro actualiza su apariencia y funcionalidad permitiendo al usuario que dé de baja el proceso correspondiente de la plataforma (figura 7.27).



Figura 7.27. El botón REGISTRAR EL PROCESO DE NEGOCIO EN LA PLATAFORMA cambia de apariencia para permitir dar de baja el BP.

Pero además de todo esto, cada vez que se presiona el botón de registro, el hecho en sí es almacenado por el agente empresa. De esta manera, si la empresa quiere solicitar la provisión de un BP de nivel superior¹⁰³, en la solicitud de provisión se incorpora un nuevo *slot* que asocia el *nickname* de la empresa al proceso de negocio que ha registrado. De esta manera, la empresa asegura su participación en la VE dinámica que se constituya para dar respuesta a su solicitud.

Para finalizar con la descripción de la interfaz del agente empresa, sólo nos resta presentar la pestaña REGLAS DE CONTESTACIÓN. Esta pestaña permite gestionar las reglas que guiarán el comportamiento del agente cuando reciba solicitudes tanto de otras empresas (el cliente ha solicitado la lista al administrador) como del administrador (el cliente ha solicitado que se publique la solicitud). En concreto, nos estamos refiriendo al archivo `IDMF_IDMF_EmpresaReglas.clp`.

103 Imaginemos, por ejemplo, que una empresa dedicada a la fabricación de paneles realiza un importante descubrimiento que mejora sus productos. En esta situación, quizás quiera aprovecharse ella misma de las oportunidades de instalación de módulos fotovoltaicos, lo que la llevaría a solicitar ella misma la provisión de tales procesos.

7.3.4 El agente cliente

La interfaz del agente usuario es muy similar a la del agente empresa, aunque obviamente con una menor funcionalidad. Como se podrá suponer, el agente cliente sólo dispone de las opciones asociadas al rol cliente descritas en el apartado anterior, ya que un agente cliente nunca podrá ser proveedor de un proceso de negocio.

7.3.5 Los agentes de servicio

Aunque los agentes de servicio definidos en esta plataforma carecen de interfaz de usuario por motivos de diseño, resulta conveniente volver a recordar que ambos obtienen la información de archivos XML mediante clases traductoras, de forma similar a lo que sucedía para el agente proveedor de información, y no a través de una interfaz de usuario que el administrador de la plataforma pueda y deba manipular.

7.3.6 Javadoc de la aplicación

Los programas en Java pueden tener dos tipos de comentarios: comentarios de implementación y comentarios de documentación. Los comentarios de implementación¹⁰⁴ están destinados a comentar el código (son comentarios sobre la implementación). Los comentarios de documentación (también conocidos como “comentarios Javadoc”¹⁰⁵) tienen por objeto describir la especificación del código, desde una perspectiva independiente de la implementación, para ser leídos por desarrolladores que pueden no tener necesariamente el código fuente a mano.

Los comentarios Javadoc están destinados a describir, principalmente, clases y métodos. Por ello, y como están pensados para que otro programador los lea y utilice la clase (o método) correspondiente, se decidió fijar al menos parcialmente un formato común, de forma que los comentarios escritos por un programador resultaran legibles por otro. Así, los comentarios Javadoc se suelen colocar al comienzo de línea y deben incluir unos indicadores especiales que siempre comienzan por '@'.

104 Están delimitados por `/* ... */` y `//`.

105 Son específicos de Java y están delimitados por `/** ... */`.

Se usan indicadores para el número de versión (*@version*), el autor (*@author*), etc. Es importante observar que los indicadores no son obligatorios; por ejemplo en un método sin parámetros no se incluye obviamente el indicador *@param*. También puede darse que un comentario incluya un indicador más de una vez, por ejemplo varios indicadores *@param* porque el método tiene varios parámetros.

Aparte de obtenerse comentarios más fáciles de leer para otros programadores debido al formato impuesto por los indicadores, la principal utilidad de estos comentarios es que pueden utilizarse para generar la documentación de los programas. Precisamente, este ha sido el propósito principal de los documentos Javadoc (a la postre documentos html) generados a partir de la aplicación. El resultado final es un .html por cada fichero .java y paquete que compone nuestra plataforma.

En particular, la figura 7.28 representa la vista general (*overview*) de la documentación generada a partir del código fuente de la aplicación asociada al usuario empresa.

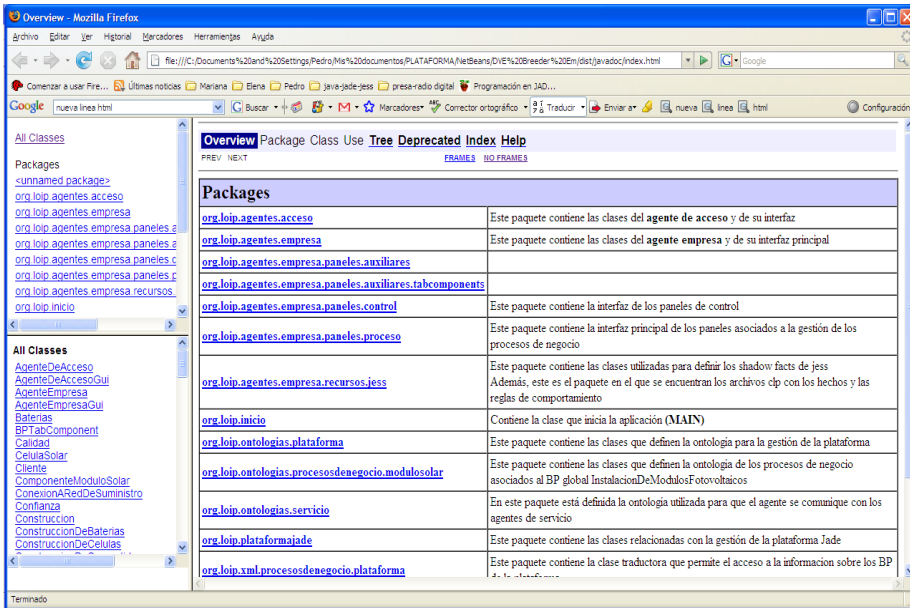


Figura 7.28. Página de inicio del Javadoc generado para DVEBreeder Em.

7.4 RESUMEN

Tras llevar a cabo la planificación, el análisis y el diseño de la aplicación en los capítulos previos, en el presente capítulo se ha abordado la descripción de los aspectos más interesantes asociados a la construcción de la plataforma multiagente. Con objeto de lograr una mejor comprensión del trabajo desarrollado, en primer lugar se han descrito las distintas herramientas empleadas para este propósito. De entre todas las analizadas en el primer apartado destacan la plataforma multiagente JADE y el sistema experto Jess.

- *JADE* nos ha permitido, ante todo, crear los distintos agentes que compondrán la plataforma distribuida *DVEBreeder*. Además de proporcionar un entorno de ejecución donde los agentes pueden “vivir”. *JADE* incorpora una librería de clases que hemos podido utilizar para desarrollar nuestros agentes y un conjunto de herramientas gráficas que permiten administrar y monitorizar la actividad de los agentes activos. Nosotros tan sólo hemos tenido que centrar nuestra atención en definir su comportamiento y apariencia. Para ese propósito (codificación) hemos empleado la *IDE Netbeans*, que posee un gran número de ventajas, entre las que posiblemente destaca la facilidad en la creación de las interfaces de usuario.
- Para dotar a los agentes de “inteligencia” en su comportamiento (en la toma de sus decisiones) hemos recurrido a la API de *Jess* y a la creación de documentos *.clp* en los que hemos definido tanto las reglas que utilizarán los agentes como los hechos que provocarán la activación del motor de inferencia. Para la elaboración de estos documentos *.clp* hemos empleado *JessDE*, un *plugin* de la conocida *IDE eclipse* que suministra la propia distribución de *Jess*.
- Otro aspecto muy importante en la construcción de aplicaciones multiagente es la definición de las ontologías. Esta tarea se ve simplificada gracias a la utilización de una herramienta visual y de gran potencia, *Protégé*, que permite definir ontologías de una manera fácil e intuitiva. Por otro lado, pasar las ontologías a clases Java que puedan ser utilizadas en *JADE*, aunque mecánica, resulta una tarea bastante tediosa y repetitiva: gracias al *plugin BeanGenerator* conseguimos reducir significativamente la construcción de estas clases. Finalmente, debemos mencionar otro *plugin* de *Protégé*, *JessTab*: gracias a esta pestaña hemos podido convertir la información contenida en las ontologías en una estructura simple aceptada por el motor de inferencia *Jess*.

La descripción de las herramientas de desarrollo deja paso a la presentación de los aspectos más interesantes de la aplicación multiagente generada. Ésta se ha realizado desde un punto de vista didáctico y siempre pensando en los usuarios finales. Por ello, se han descrito las interfaces de usuario de los principales agentes involucrados, es decir, el agente administrador, el agente de acceso y el agente empresa (la interfaz del agente cliente es una versión reducida de esta última).

- La interfaz del *agente administrador* contiene herramientas que permiten el acceso a la información de los usuarios de la plataforma (a través de una comunicación con el agente proveedor de información), a los distintos procesos de negocio registrados y a la información que se genera durante la vida de este agente (registro de procesos de negocio de los agentes empresa, comunicación con el agente proveedor de información de la plataforma, etc.). Esta información se distribuye en la GUI por medio de tres ventanas.
- La funcionalidad del *agente de acceso* se reduce simplemente a permitir que el usuario (tanto cliente como empresa) pueda introducir los datos relativos a la plataforma a la que se unirán los agentes, y los datos (identificación y clave de acceso) que permitirán verificar la identidad del agente y su pertenencia a la plataforma.
- El *agente empresa* tiene una interfaz compuesta, como en el caso del agente administrador, por un menú y por diferentes ventanas donde se muestran los procesos de negocio, los hechos y reglas necesarios para la interacción con otros agentes, y la información que resulta de dicha interacción. Así, seleccionando simplemente un proceso de negocio de la ventana homónima tenemos acceso a una pestaña que permite seleccionar el rol de la empresa para dicho proceso. En función de este rol podremos: (cliente) definir las características del proceso que pretendemos nos suministren, establecer el tipo de control, definir las restricciones y las reglas que guiarán la selección de los proveedores, (empresa) registrar el proceso de negocio en la plataforma y concretar tanto las características de los servicios que ofrece la empresa como las reglas de contestación que permitirán seleccionar el mejor servicio para cada solicitud de los clientes.
- La interfaz del *agente cliente* es bastante similar a la del agente empresa aunque sin las funcionalidades asociadas al rol empresa.
- Por último, también se ha hecho mención a la información Javadoc generada durante la construcción de la aplicación, que sin duda constituirá un elemento importante en los desarrollos futuros.

En definitiva, disponemos ya de todos los elementos que compondrán nuestra plataforma multiagente distribuida, por lo que estamos en disposición de iniciar el proceso de validación. Con este propósito, en el próximo capítulo definiremos una serie de escenarios que servirán para comprobar el correcto funcionamiento de la aplicación y verificar, a su vez, la idoneidad de la tecnología empleada y, muy especialmente, del modelo que ha servido de guía en su desarrollo.

CAPÍTULO

8

VALIDACIÓN Y VALORACIÓN FINAL

8.1 INTRODUCCIÓN

Tras la construcción de la plataforma multiagente, llega el momento de estudiar su comportamiento y comprobar si éste responde de forma adecuada a las necesidades que plantea el proceso de selección de socios en las VE dinámicas. Este estudio nos servirá, además, para verificar la idoneidad de la tecnología empleada en su construcción, y para determinar la validez del modelo que durante todo este tiempo ha guiado el desarrollo de nuestra herramienta informática.

Para realizar este estudio se definirán varias situaciones ficticias que tendrán como marco contextual el dominio de aplicación definido en el capítulo 6, que no es otro que la instalación de colectores solares fotovoltaicos. Cada una de estas situaciones diferirá del resto en el valor de ciertos parámetros importantes como son, por ejemplo, los precios de los servicios, el tiempo de suministro, las características operativas de cada proveedor (eficiencia, confianza, etc.), el precio del combustible,... , e incluso las propias reglas.

La primera parte del capítulo se orienta a estudiar la respuesta de la aplicación multiagente en cada uno de estos escenarios, prestando una especial atención a la configuración seleccionada en cada caso y a los valores de coste y tiempo de suministro finales. Obviamente, el resultado más importante que se obtiene en cada escenario no es otro que la formación de una VE dinámica en la que se verán involucradas algunas de las entidades definidas previamente. El estudio del comportamiento de la aplicación ante cada situación nos permitirá verificar su correcto funcionamiento, pero también la adecuación de la tecnología empleada en su construcción y la validez del modelo de partida.

Será el turno, entonces, de valorar cada uno de estos tres aspectos (modelo, tecnología y solución multiagente) con un mayor nivel de detalle, para lo que será necesario un análisis previo de aquellos elementos que definen el contexto de nuestro trabajo. Así, la segunda parte del capítulo comienza con una discusión sobre el estado actual y futuro del paradigma de la DVE, incluido el papel que desempeñan los entornos de gestación como marcos evolutivos, y continúa con un análisis de los aspectos más importantes relativos a la tecnología empleada en la construcción de la plataforma. Finalmente, se afronta el estudio de los logros alcanzados entre los que se encuentran, obviamente, el modelo creado y la herramienta desarrollada.

8.2 VALIDACIÓN

Desde un primer momento, hemos tenido presente que el proceso de validación debía realizarse con un triple propósito: constatar el correcto funcionamiento de la aplicación, verificar la adecuación de la tecnología empleada y, lo más importante, comprobar la idoneidad del modelo que ha servido de base en su construcción. Alcanzar la primera de estas metas es una tarea relativamente fácil ya que, en definitiva, tan sólo exige estudiar su comportamiento y comprobar que el software final verifica las propiedades y funciones establecidas en los requerimientos. En otras palabras, cerciorarse de que todos sus componentes se comportan tal y como se espera.

Es en las otras dos metas donde surgen las complicaciones debido a que no podemos comparar nuestra solución con otras presentes en la literatura. Una primera explicación a este hecho se halla en la naturaleza de la solución planteada. En efecto, desde un principio hemos tratado de conseguir que nuestros agentes se comportasen de forma similar a como lo haría cualquier

directivo humano. Gracias al sistema experto hemos logrado esta meta, pero nos hemos alejado de las soluciones ofrecidas hasta el momento: son un conjunto de reglas y hechos los que definen el comportamiento de nuestros agentes y no complejas funciones matemáticas, por lo que su comparación resulta inadecuada.

A este hecho debemos añadir que en la literatura de investigación más cercana a nuestros propósitos nunca se presenta completamente la solución ni sus resultados, tan sólo una parte reducida de la misma. A este cóctel le podemos sumar un ingrediente más: el dominio de aplicación considerado, la instalación de colectores fotovoltaicos, nunca antes se había planteado como contexto operativo en este tipo de trabajos. Todos estos ingredientes hacen muy difícil una comparación de nuestro trabajo, tanto del modelo como de la aplicación software.

Por otro lado, siempre se pueden comparar algoritmos, o programas de secuenciación, etc., cuando se conoce la solución óptima y tan sólo queremos saber cuál de ellos se acerca más y más rápido a la misma. Sin embargo, ¿cuál es la solución óptima en nuestro problema?, ¿existe realmente?. Lo cierto es que cuando se considera un grupo de agentes con criterios y reglas de comportamientos que pueden variar drásticamente dependiendo de cada situación y cada experto humano, es imposible conocer *a priori* cuál será la mejor solución¹⁰⁶. A lo sumo podremos, y en casos muy concretos, intuir hacia dónde se dirigirán las diferentes elecciones, pero poco más.

En resumen, la validación de la plataforma y del modelo han de realizarse en base exclusivamente al estudio del comportamiento de la aplicación, ya que una comparación con otras soluciones es, a todas luces, imposible.

8.2.1 Nomenclatura

Una característica distintiva de la DVE es que sus miembros son entidades legalmente independientes que se encuentran geográficamente dispersas como consecuencia de la necesidad de contar en cada momento con las entidades mejor preparadas a escala mundial. Nuestra aplicación permite que las empresas estén ubicadas en cualquier parte del mundo, pero para nuestros propósitos de verificación y validación este requisito puede ser relajado, lo

106 Más difícil, si cabe, cuanto mayor sea el número de agentes involucrados.

que nos permite reducir el campo de actuación de los distintos escenarios que planteemos. De hecho, en todos y cada uno de ellos tan sólo hemos considerado que las empresas que forman parte del entorno de gestación se circunscriben al ámbito de la comunidad de Castilla y León (figura 8.1).



Figura 8.1. Ámbito de aplicación de los escenarios propuestos (Comunidad de Castilla y León).

Atendiendo a esta limitación y a la estructura de los procesos de negocio definida para el dominio de aplicación considerado, se ha establecido una nomenclatura especial con la que denominar a las empresas ficticias que participarán en los escenarios considerados (ecuación 8.1), lo que permite un análisis más sencillo e intuitivo de los resultados finales.

$$\text{BPGlobal}_{(\text{niv})}_{(\text{pos})}_{(\text{caprov})}_{(\text{num})} \quad (8.1)$$

donde:

- BPGlobal define el proceso de negocio global,
- niv hace referencia al nivel jerárquico del proceso de negocio,
- pos establece la posición dentro del nivel,
- caprov representa la capital de provincia donde tiene su sede la empresa, y
- num se emplea para numerar las distintas empresas que ofrecen el mismo proceso de negocio en la capital considerada.

Para representar *niv*, *pos* y *num* emplearemos dos dígitos, por lo que para cada uno de ellos podremos encontrar valores comprendidos entre 1 y 99 (no se considera el cero). En el caso de *caprov* utilizaremos las dos primeras letras de la capital de provincia.

La asignación de los valores correspondientes a los parámetros *niv* y *pos* se ha realizado en base a la figura 8.2.

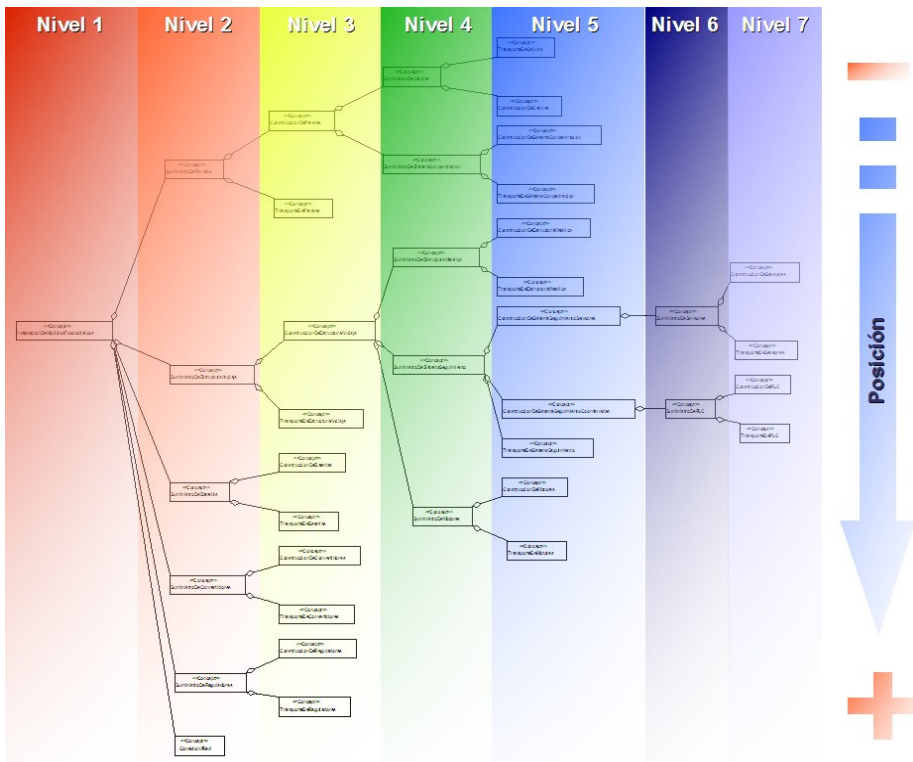


Figura 8.2. Representación gráfica de la asignación de los valores de nivel jerárquico (*niv*) y posición ocupada dentro de cada nivel (*pos*) para los distintos procesos de negocio.

Teniendo presente esta convención de nombrado para las empresas de la plataforma, un nombre como *IDMF_01_01_av_03* nos estaría indicando que se trata de la tercera empresa abulense que suministra el proceso de negocio *InstalacionDeModulosFotovoltaicos*.

Para representar gráficamente los resultados nos valdremos del esquema de colores empleado en el dibujo anterior. Así, la figura 8.3 presenta los procesos que componen el BP global *InstalacionDeModulosFotovoltaicos*, $IDMF_ (niv)_ (pos)$ ¹⁰⁷.

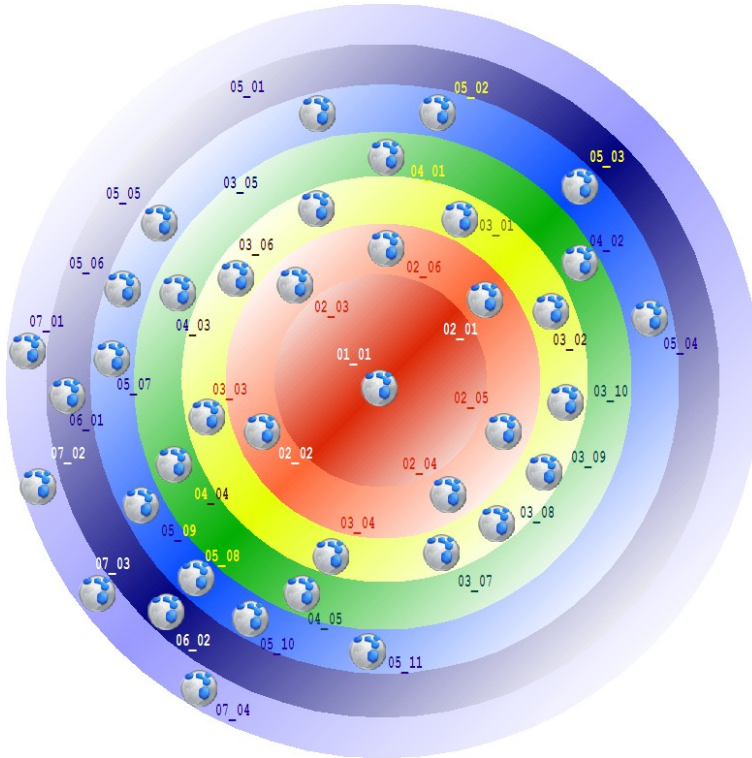


Figura 8.3. Representación gráfica empleada en los resultados.

Por último, a partir de ahora cuando queramos referirnos al tipo de control más laxo que puede ejercer el cliente (el que da lugar a la selección de una VE dinámica) hablaremos de “control DVE”. Si ejerce un control más férreo sobre todos los procesos remotos hablaremos de “control Empresa”, y en el resto de los casos usaremos la denominación “control Mixto”¹⁰⁸.

107 Dado que tan sólo hemos considerado un único proceso de negocio global, en esta representación y en las siguientes omitiremos el término $IDMF_$ a fin de favorecer su visualización y comprensión.

108 Generalmente consideramos la selección de DVE para los BP de primer nivel.

8.2.2 Escenario 1

¿Qué ocurre si todos los proveedores de un mismo proceso de negocio comparten el mismo módulo de decisión, es decir, las mismas reglas y los mismos parámetros de comportamiento?. Obviamente, nos encontramos ante una situación ideal de *competencia perfecta* en la que las distintas empresas suministradoras de un mismo BP ofrecen el mismo producto al mismo precio. No existen empresas vetadas, ni preferidas, ni descuentos, etc., y los valores de sus propiedades operativas (calidad, confianza, etc.) son máximos.

Este es, además, el estado en el que se encuentra la plataforma inicialmente, antes de personalizar los datos para cada una de las empresas. Si el tipo de control que ejerce cada empresa, incluido el cliente, es “control DVE” el resultado final es el que aparece en la figura 8.4.

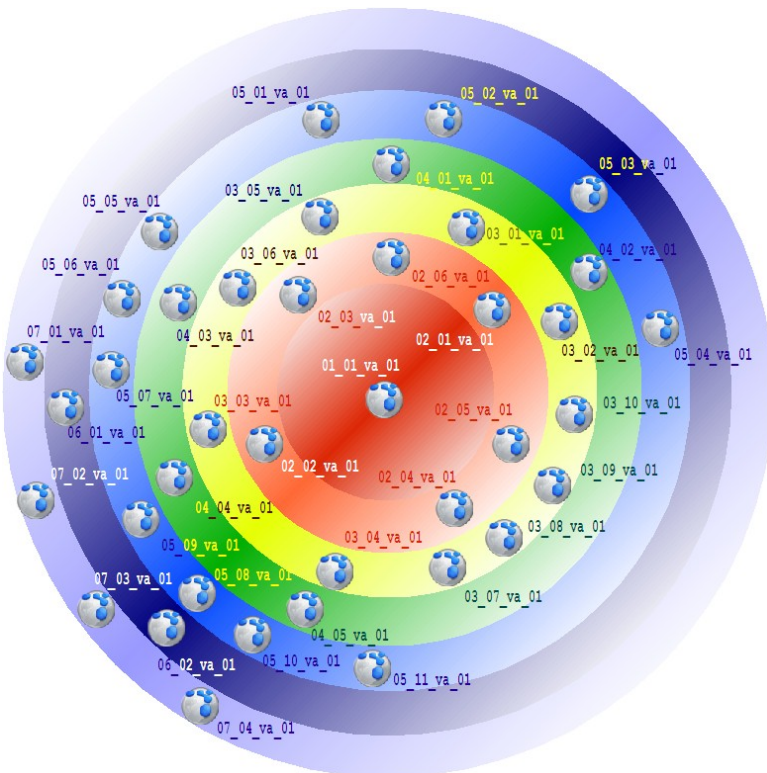


Figura 8.4. Resultado del escenario 1 y control DVE.

En este caso, el cliente que inicia el proceso de selección requiere que se le suministre el proceso de negocio en Valladolid¹⁰⁹, lo que conduce a que todas las empresas seleccionadas tengan su sede en esta localidad. Este comportamiento es fácil de explicar si consideramos que a igualdad de factores operativos, igual producto y precio, etc., las empresas que residen fuera de Valladolid se encuentran con el hándicap del coste de transporte que debe unirse al coste de sus productos-servicios.

Dado que inicialmente en cada ciudad y para cada proceso de negocio se ha considerado que sólo existe un único proveedor se obtiene la misma selección de empresas representada en la figura 8.4 cuando el tipo de control es más restrictivo. Esto incluye situaciones en las que el cliente controla todos y cada uno de los procesos remotos que componen el proceso de negocio global (“control Empresa”), o cualquier combinación posible entre los dos tipos de control anteriores (“control Mixto”).

En todos los casos, el precio final al que adquiere el servicio nuestro cliente vallisoletano asciende a 2395 u.m. y el tiempo de suministro, el que fija la línea de proceso de negocio más lenta, es de 27 días¹¹⁰.

El escenario descrito en este apartado nos ha servido para entrar en contacto con la aplicación, pero no es representativo de la realidad. En los próximos apartados vamos a ir alejándonos de este comportamiento ideal, lo que supondrá modificar los parámetros de la competencia para reflejar situaciones cada vez más imperfectas.

8.2.3 Escenario 2

Consideramos ahora que todas las empresas vallisoletanas mantienen los valores iniciales de precio y tiempo de suministro mientras que el resto de empresas castellanoleonesas los ven reducidos¹¹¹ de forma aleatoria hasta un máximo del 30%. Considerando un tipo de control DVE la solución final es la representada en la figura 8.5 en la que el precio final es de 1995,28 u.m. y el tiempo de suministro 24,47 días.

109 El cliente requiere todos y cada uno de los procesos de negocio definidos.

110 Recuérdese que los datos de coste y tiempo de suministro que hemos considerado para los distintos procesos de negocio y las distintas empresas son ficticios.

111 Bien por una mano de obra más barata, procesos productivos más eficientes, etc.

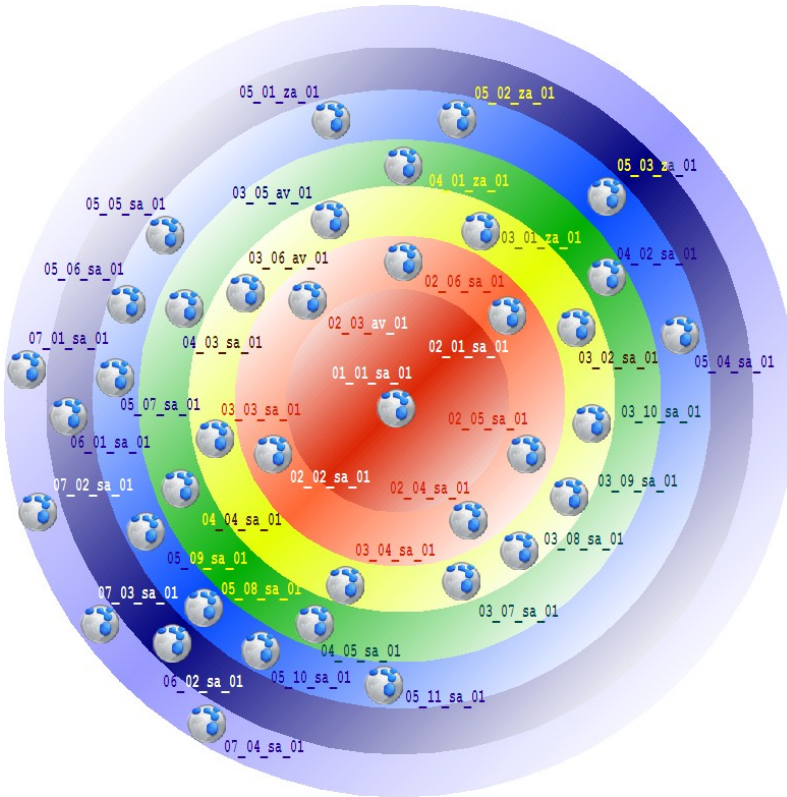


Figura 8.5. Resultado del escenario 2 y control DVE.

Ahora la empresa suministradora del proceso de negocio *InstalacionDeModulosFotovoltaicos* es de Salamanca, lo que condiciona notablemente la procedencia del resto de empresas encargadas de proveer los procesos remotos; de hecho, y como se puede apreciar claramente en la imagen anterior, las empresas que componen la VE dinámica seleccionada tienen ahora su centro de gravedad en Salamanca.

Si se considera un tipo de control Mixto en el que la empresa cliente selecciona directamente el proveedor del BP *InstalacionDeModulosFotovoltaicos* y deja que se configuren DVE para los procesos de primer nivel, la distribución está más diversificada (figura 8.6). En esta ocasión la empresa suministradora del proceso *InstalacionDeModulosFotovoltaicos* vuelve a ser la empresa salmantina, pero para el resto de procesos de negocio

de primer nivel encontramos proveedores de Ávila, Burgos, Palencia, León, Segovia e incluso Soria. Con el nuevo tipo de control el precio final sube ligeramente (2052,84 u.m.) aunque el tiempo de suministro es incluso inferior (23,77 días).

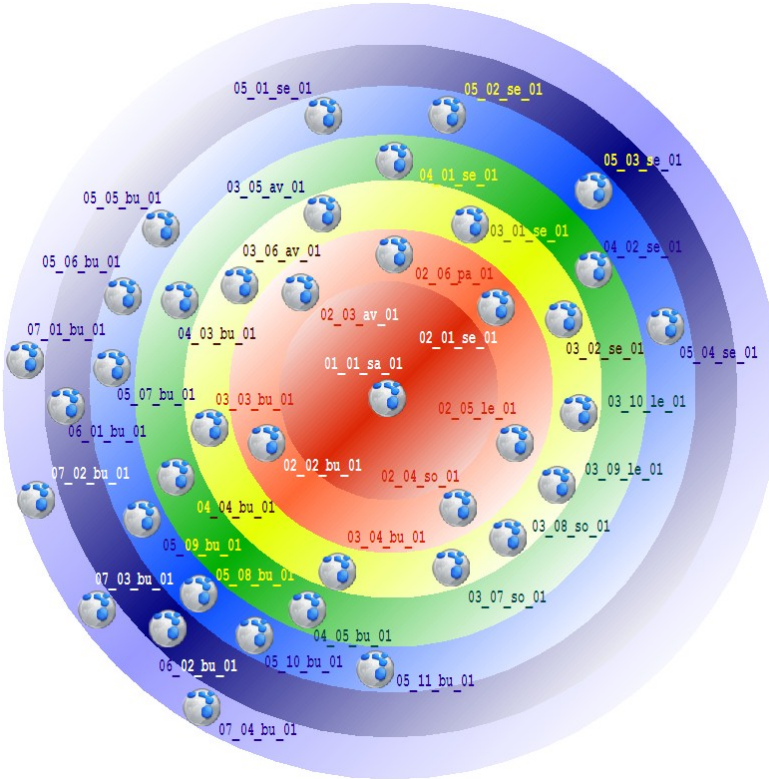


Figura 8.6. Resultado del escenario 2 y control Mixto.

8.2.4 Escenario 3

Consideremos ahora que la empresa salmantina seleccionada en el escenario previo (representado en la figura 8.6) no desempeña correctamente su labor¹¹² y necesitamos una nueva empresa que nos instale los módulos fotovoltaicos (en definitiva, necesitamos una reconfiguración de la red).

¹¹² Vale cualquier razón posible: no cumple los plazos, no emplea personal cualificado, etc.

8.2.5 Escenario 4

Con el funcionamiento normal de la plataforma los valores de las propiedades operativas (las gestionadas por el administrador) varían. Por ello, consideremos ahora que las empresas ven reducidos estos valores de forma aleatoria hasta un 50%. De acuerdo con las reglas de comportamiento establecidas el coste de transporte pierde la relevancia que poseía en los escenarios anteriores, por lo que la configuración resultante deberá estar geográficamente mucho más distribuida, tal y como sucede en la figura 8.8.

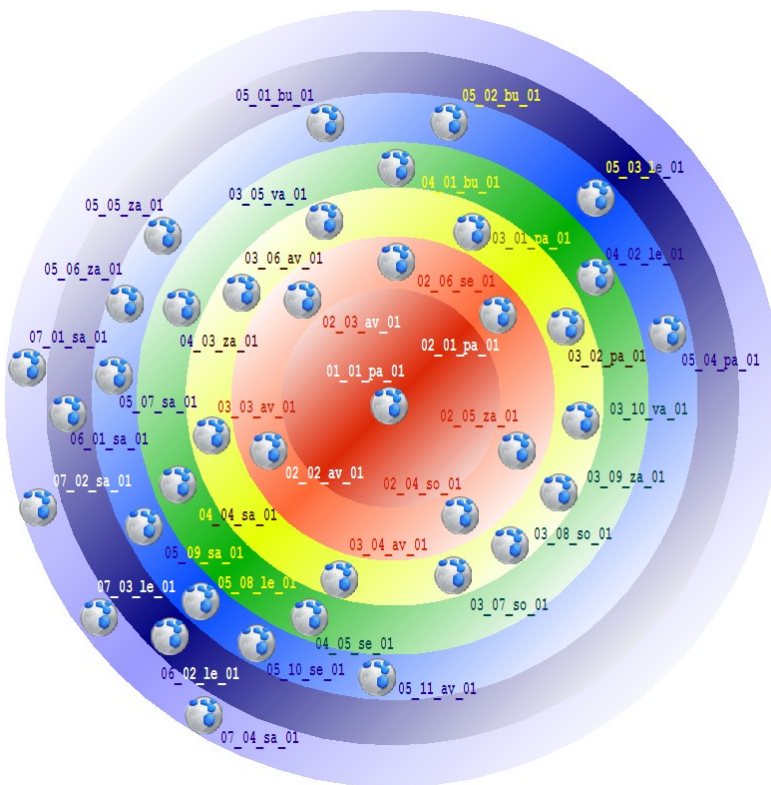


Figura 8.8. Resultado del escenario 4.

Los valores correspondientes al precio (2209,78 u.m.) y tiempo de suministro (25,94 días) se disparan comparándolas con las dos soluciones anteriores, pero lo importante es que estamos seguros de contar con las empresas que mejor lo han estado haciendo desde que entró en operación la plataforma.

8.2.6 Escenario 5

¿Qué ocurre si por circunstancias exógenas el precio del combustible aumenta un 30%?. La primera consecuencia inmediata es una reducción considerable de la dispersión geográfica de la solución, tal y como aparece reflejado en la figura 8.9, ya que con este aumento el coste de transporte vuelve a adquirir una mayor relevancia en el coste final.

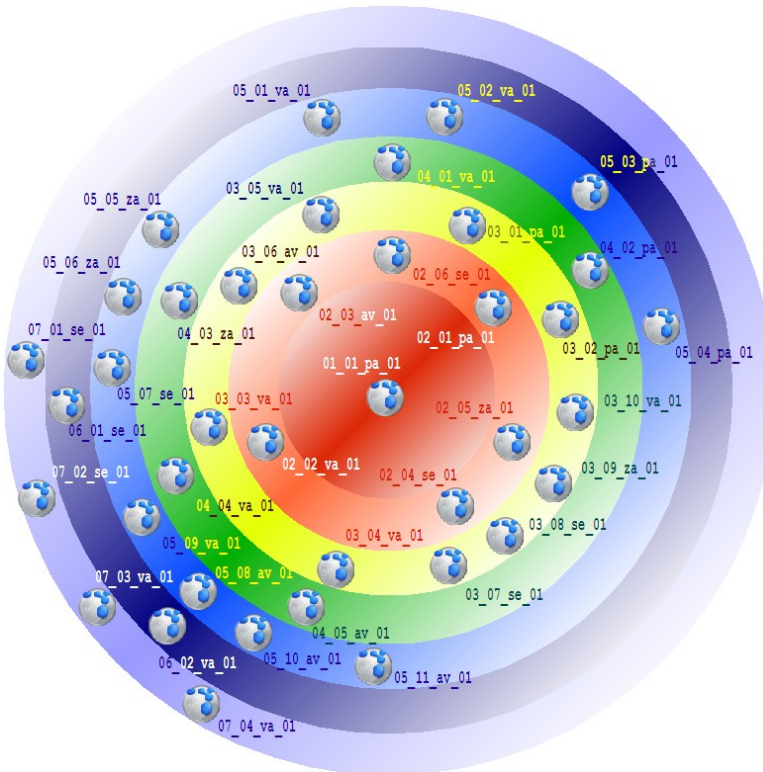


Figura 8.9. Resultado del escenario 5.

Además, ahora aparecen en la solución empresas vallisoletanas en BP no vinculados al transporte. Desde el primer escenario analizado las empresas pucelanas no habían entrado a formar parte de la solución por el elevado precio y tiempo de suministro de sus procesos (recuérdese que estas empresas no vieron reducidos estos valores en el escenario 2): el aumento del precio de combustible las hace competitivas frente al resto.

En este escenario el precio del servicio (2374,2 u.m.) es superior al considerado en la situación previa, aunque la VE dinámica resultante es capaz de proveer el proceso de negocio en un total de 22,17 días. Obviamente, nos decantaremos por esta configuración siempre que el coste de oportunidad que representa la pérdida de la subvención sea superior al ahorro que conlleva seleccionar la DVE del escenario 5, es decir, 106,12 u.m.

8.2.8 Escenario 7

Consideremos ahora que una empresa con sede en Ávila ha realizado un invento revolucionario en la construcción de células fotovoltaicas, de forma que éstas ven aumentada su eficiencia un 25% (y su precio un 15%). Para asegurar su participación en la VE dinámica, deberemos incorporar el nombre local de esta empresa en la solicitud a través del *slot* que hace referencia al proceso *ConstruccionDeCelulas*¹¹⁴.

La configuración del escenario 6 se ve ahora modificada para dar cabida a esta empresa abulense, tal y como aparece representado en la figura 8.11. La instalación del huerto solar supone para el cliente un desembolso ligeramente superior (2437,73 u.m.) como consecuencia, principalmente, del mayor precio de las células. Por lo que se refiere al tiempo de suministro, éste también ve modificado su valor, pasando de los 22,17 a los 22,98 días.

8.2.9 Conclusiones principales

Es evidente que se podrían haber planteado muchos más escenarios a través de la incorporación de nuevas empresas, la modificación de otros criterios y reglas de selección (o de las características internas de las empresas), la selección de tipos de control diferentes, etc. El abanico de posibilidades que se abre es inmenso, aunque para nuestros propósitos resulta innecesario estudiar cada una de ellas. Estamos en disposición, por tanto, de dar un paso hacia delante y comenzar a valorar el trabajo realizado. No obstante, antes de proceder con esta valoración creemos acertado remarcar las conclusiones principales del estudio que se ha pretendido resumir en este apartado.

114 Esto lo podemos hacer fácilmente accediendo a la pestaña del proceso de negocio correspondiente y seleccionando, con el rol cliente, cualquiera de los dos botones disponibles (publicar o solicitar lista).

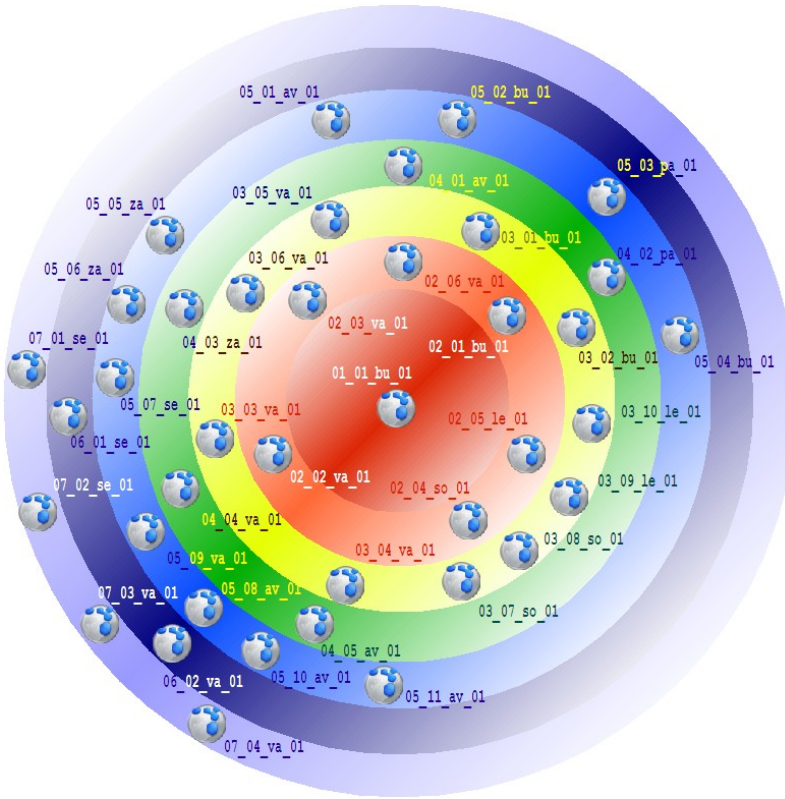


Figura 8.11. Resultado del escenario 7.

- *Funcionamiento correcto de la plataforma.* El laborioso estudio de los escenarios pone de manifiesto que tanto el comportamiento como las funcionalidades de la plataforma responden de forma coherente, ajustándose a la lógica.
- *Adecuación de las tecnologías empleadas.* El uso combinado de los sistemas multiagente y los sistemas expertos ha sido muy acertado: el MAS ha permitido la formación de un sistema distribuido donde cada componente de la plataforma está representado por un agente que muestra una conducta inteligente gracias al ES. Los expertos de las empresas pueden depositar su conocimiento y sus reglas de decisión en el módulo de decisión de los agentes, lo que permite acercar su comportamiento en el proceso de selección a la conducta real de los directivos de las empresas.

- Sin embargo, la principal conclusión del estudio es *la completa validez y eficacia del modelo de partida*. Este modelo multienfoque nos permite crear cualquier tipo de VE dinámica gracias a una simple elección, lo que le convierte en un modelo innovador desde el punto de vista conceptual a la vez que realista y completo. Además, su simplicidad es obvia.

8.3 VALORACIÓN

Tal y como se mencionara al comienzo del capítulo, en este apartado se va a proceder a valorar el trabajo realizado. No obstante, esta valoración estaría incompleta si no incluyésemos un análisis crítico de los principales elementos que lo han hecho posible, incluido el concepto de VE dinámica y la tecnología empleada en la construcción de la aplicación software.

8.3.1 La VE dinámica y su contexto

LA EMPRESA VIRTUAL DINÁMICA

Este modelo de negocio engloba, esencialmente, a participantes de mercados pequeños y altamente especializados que están esperando una oportunidad para ofrecer sus servicios en un dominio específico y trabajar junto a socios complementarios con el fin de satisfacer los requisitos de los clientes. Su principal ventaja es una mayor flexibilidad y adaptabilidad a los cambios que se suceden en los actuales mercados dinámicos y globales. Frente a este modelo, las grandes compañías tradicionales, demasiado lentas para adoptar y desarrollar nuevos conocimientos y habilidades que les permitan continuar en el mercado, están destinadas a sufrir e incluso desaparecer.

Atendiendo a la taxonomía aportada en este trabajo podemos afirmar que una VE dinámica hace un uso intensivo del concepto de metagestión, dado que posee un elevado grado de flexibilidad en su gestión y en la selección de los suministradores que cumplan los requisitos de la organización. Muestra, además, una naturaleza variable y diligente, en la que algunas empresas pueden unirse o abandonar la alianza dinámicamente dependiendo de las fases del proceso de negocio u otros factores de mercado. En otras palabras, el número de socios cambia de acuerdo con la ley de oferta y demanda.

La colaboración se basa puramente en el oportunismo, ya que se establecen para satisfacer una oportunidad específica de negocio y se disuelven una vez finaliza la misma. Los retornos de la inversión suelen darse durante una única transacción, a no ser que se participe en asociaciones a largo plazo como las que representan los VBE. Además, las relaciones dinámicas requieren que los procesos de negocio estén soportados por tecnología de información y que existan interfaces y estándares comunes para el intercambio de datos.

Aunque existen algunas aplicaciones de este concepto prácticamente ninguna muestra una forma pura que satisfaga todas las características analizadas en la taxonomía. Algunos autores ven en la consultoría y en la industria de la construcción ejemplos reales de DVE: sin embargo, en ambos casos las tecnologías como las que se proponen en este trabajo son solamente medios para soportar equipos de proyecto interorganizacionales, aunque no son realmente necesarias.

Por otra parte, la industria informática, tanto software como hardware, está a la vanguardia de los progresos en las tecnologías de la información y comunicación dado que éstas son su *core business* y porque necesitan ser flexibles y adaptarse a los rápidos movimientos del mercado. Ejemplos especiales son los proyectos de software Open Source, que pueden ser vistos como realizaciones puras de empresas virtuales dinámicas, claro está, en un contexto no puramente comercial.

No obstante, y dadas las características actuales de los mercados, las DVE resultan claramente convenientes para aquellas oportunidades de negocio únicas y específicas que sean demasiado complejas, en términos de capacidades y experiencia requeridas, para una sola compañía y en las que el tiempo se erija como factor crítico. Pero la unión de las diferentes empresas puede verse obstaculizada por la falta de confianza: el éxito de la DVE depende, por tanto, del poder de persuasión de sus posibles miembros en lo que se refiere a integridad y honradez.

Una vez asignado un proyecto a un consorcio de DVE debe garantizarse la correcta ejecución y provisión de los servicios estipulados. El esfuerzo se verá recompensado sólo cuando todos los componentes operen como un único elemento. Sin embargo, resulta muy habitual que los socios no se conozcan entre sí, lo que puede provocar suspicacias. Es necesaria, por tanto, una gestión fluida y concisa del proyecto y que todos los puntos potencialmente problemáticos se regulen previamente mediante contratos.

Los *problemas de organización* suelen proceder de la incertidumbre que surge cuando socios diferentes y altamente heterogéneos se unen para perseguir una meta común. Las culturas y actitudes diferentes pueden dar lugar a posiciones conflictivas que deben dejarse a un lado si se quiere lograr la meta de la DVE¹¹⁵. El desafío principal es encontrar los elementos de construcción apropiados para todas las habilidades requeridas, para que los socios comiencen a trabajar juntos de una manera eficaz y para lograr una disolución clara y precisa tras haber cumplido el objetivo del proyecto, manteniendo la interfaz con el cliente si es necesario.

Entre los *problemas tecnológicos* encontramos la integración de entornos IC heterogéneos para compartir datos y lograr la comunicación entre los socios. Los sistemas que colaboren sin intervención humana, y sin los oportunos costes de establecimiento, deben comprender los formatos y la semántica de cada uno de los sistemas de los socios. El principal desafío es conseguir imponer los mismos conceptos ICT (formatos de intercambio de datos, ontologías, etc.) a todos los socios; en definitiva, lograr un acuerdo entre los distintos componentes.

La *seguridad* de los datos es otro aspecto importante a considerar. Se supone que los participantes permiten el acceso a sus datos pero al mismo tiempo desean guardar su conocimiento estratégico y su maestría ocultándolos a los competidores potenciales. Debemos encontrar una solución de compromiso, fácil y concluyente, que proporcione un punto de encuentro entre ambos objetivos. Ambos problemas, heterogeneidad de infraestructuras y seguridad, pueden encontrar una solución eficaz en los entornos de gestión de DVE.

A pesar de estas dificultades, la perspectiva asociada a las VE dinámicas es prometedora considerando los recientes progresos en tecnología de información y comunicación, y la tendencia actual de las compañías de centrarse en sus *core business*. La unión de diferentes entidades con capacidades complementarias para satisfacer una oportunidad única de mercado es concebible si se dan los requisitos previos necesarios y si demuestra ser una alternativa superior a los modelos tradicionales.

115 Siempre debemos tener presente que muchas cosas pueden ir mal durante la vida de una DVE, pero también que solamente hay una ocasión de hacer las cosas correctamente. Cada proyecto es único y no podemos perder tiempo en solucionar conflictos entre los socios: la oportunidad de negocio se nos puede escapar de las manos si no somos capaces de responder con celeridad.

Aún así, las DVE no han emergido aún con fuerza y sólo se observan, como hemos mencionado, en determinados nichos de mercado (principalmente en el campo informático). Los procesos estandarizados y repetitivos se gestionan mejor a través de una compañía jerárquica y establecida, mientras que las industrias que requieren grandes inversiones en capital e I+D+i, como la industria farmacéutica, necesitan una construcción organizacional a largo plazo que les permita recuperar las inversiones realizadas.

Por otro lado, la excelencia del negocio y las ventajas competitivas que constituyen la competencia nuclear de una compañía se guardan como un tesoro y no se comparten fácilmente en un proyecto común. En este contexto, la seguridad podría ayudar a disminuir la amenaza del espionaje industrial. Además, una compañía con una posición de mercado fuerte probablemente dirija su energía a ampliar sus dominios (VE estática) en lugar de colaborar y compartir su poder con agentes externos.

Por último, debemos tener siempre presente que la ventaja de una forma de organización flexible y ágil reside en su capacidad para reaccionar rápidamente ante los problemas y de responder en consecuencia. Las economías de escala no son un factor clave para este modelo de negocio que busca, ante todo, proporcionar soluciones personalizadas y en las que el tiempo es un factor crítico. Sin embargo, pueden llegar a constituir un factor de éxito si las relaciones se prolongan en el tiempo.

ALGUNAS CUESTIONES SOBRE LA INVESTIGACIÓN FUTURA ALREDEDOR DE LA DVE

La empresa virtual dinámica, como forma de organización reciente que ha surgido en un entorno extremadamente turbulento, está supeditada a continuos cambios y mejoras encaminadas a hacer de ella, la estrategia empresarial del siglo XXI. Sin embargo, es muy escaso el número de empresas que hoy en día trabajan bajo esta filosofía debido, en parte, a las razones discutidas anteriormente.

Es previsible que en el futuro surjan un gran número de investigaciones teóricas y empíricas orientadas a resolver algunos de los requisitos de este tipo de empresas y garantizar, así, su aplicabilidad. Estos proyectos se deberán diseñar congregando a investigadores de diversos campos de especialización para responder a las cuestiones abiertas más urgentes desde una visión multidisciplinar.

Y el primer paso ha de consistir en mostrar qué ventajas presenta la DVE en comparación con las compañías tradicionales. Debemos considerar factores como costes de transacción, especificidad del producto, flexibilidad, agilidad, etc., pero sobre todo su capacidad de supervivencia. En segundo lugar, el comportamiento de las personas que trabajan en equipos virtuales debe analizarse desde una perspectiva interorganizacional; los psicólogos pueden encontrar incentivos que simplifiquen la integración de los empleados en culturas organizacionales diferentes con el fin de mejorar la colaboración.

Los proyectos de investigación futuros también tendrán que ocuparse de una gestión de proyectos apropiada y de las implicaciones a largo plazo de la gestión del conocimiento cuando no hay un combinado estable de las personas y sistemas ICT implicados. De la misma manera, deben resolverse muchas de las carencias de las soluciones ICT disponibles, como la interoperabilidad, la seguridad, etc. Por otra parte, se han de proveer y promover estándares globales públicos para eliminar las barreras de entrada, lo que abriría esta forma de hacer negocio a un grupo objetivo más amplio.

Pero, probablemente, lo más importante y necesario antes de que los distintos proyectos puedan iniciarse quizás sea definir claramente qué es la DVE y cuáles son sus componentes. Aunque existen algunos esfuerzos significativos en esta dirección, el objetivo debe ser la creación de un trabajo estándar que proporcione un marco global que ayude a los investigadores, lo que contribuirá a aumentar la aceptabilidad de este campo de investigación. Reunir y elaborar parte de este marco conceptual ha sido, precisamente, uno de los logros significativos de la presente tesis.

Por último, cabe reseñar la necesidad de un lenguaje descriptivo de los procesos de negocio común que permita describir virtualmente cualquier tarea o proceso sin importar a qué industria pertenezca. Subsanan esta carencia plantea un gran desafío que deberá afrontarse con éxito si se quiere garantizar un desarrollo práctico de la VE dinámica.

LOS ENTORNOS DE GESTACIÓN DE VE

De la valoración anterior se desprende que lograr la aplicación práctica a gran escala del concepto de DVE requiere, primero, subsanar un gran número de obstáculos, empezando por aquellos relativos a su formación. Su éxito exige la disponibilidad de información adecuada sobre los socios potenciales, de un

nivel de preparación suficiente (a nivel de empresa, tecnología, etc.), de que sus miembros compartan ciertas metas comunes, de que exista un cierto grado de confianza mutua, de que se hayan establecido infraestructuras informáticas comunes con las que poder interoperar entre sí, de que se hayan acordado ciertas prácticas de negocio y reglas operativas comunes, etc. En definitiva, depende de que los socios potenciales estén preparados para participar en este tipo de colaboración.

Si bien garantizar este conjunto de requerimientos básicos puede llegar a suponer un reto bastante difícil de alcanzar cuando las relaciones de cooperación son temporales, la situación no es tan crítica cuando se consideran procesos de colaboración a largo plazo que no están limitados a una única oportunidad de negocio. Este sería el caso, por ejemplo, de las empresas extendidas (o VE estáticas), los clusters de empresas o las cadenas de suministro tradicionales.

Teniendo presente esta consideración, recientemente han surgido algunos experimentos en los que se crea un club de organizaciones a largo plazo que están preparadas para colaborar y del que surgen coaliciones a corto plazo. De esta forma, diferentes subconjuntos de organizaciones pertenecientes al club se unen, de un modo dinámico, para responder a las oportunidades de negocio. Gracias a estas nuevas formas organizacionales (VBE) muchos de los problemas analizados presentan una solución fácil e intuitiva.

Un VBE tiene como objetivo mejorar el nivel de preparación de sus miembros en lo concerniente a la formación y operación de futuras DVE. Por tanto, este tipo de asociaciones proporcionan un lecho sobre el que establecer, de forma ágil y dinámica, las posibles redes de colaboración que surjan tras la identificación de oportunidades de negocio concretas. Esta realidad hace que sea mucho más rápido, mucho más eficaz y bastante menos costoso, erigir una DVE en el contexto de un VBE que a través de una búsqueda generalizada de socios en un universo abierto de empresas.

Los VBE aplican infraestructuras de información y comunicación efectivas que suministran una base común con la que lograr niveles de colaboración e interacción apropiados entre sus miembros, facilitan y agilizan la configuración de VE ágiles (además de ayudar en su operación), introducen mecanismos para construir confianza y definen una cultura económica basada en la cooperación estableciendo unos principios y valores comunes entre las diferentes organizaciones, independientemente de donde se encuentren.

Los miembros del VBE deben obedecer las reglas generales y las políticas en él definidas pero, a cambio, estas organizaciones acceden y se benefician de múltiples elementos: información, servicios y herramientas comunes que constituyen la bolsa de activos del VBE; mercados y canales de distribución comunes; recursos y factores de trabajo comunes; recursos para compartir el coste de nuevas experiencias; lazos culturales comunes; instalaciones para compartir las lecciones aprendidas; etc.

Un VBE permite reducir el riesgo de grandes pérdidas ocasionadas por el fracaso de alguna de las organizaciones involucradas, facilita las guías generales de colaboración (reglas de conducta, principios de trabajo y cooperación, sistemas de valoración, cultura y ética de colaboración, protección de la información,...), asegura la cobertura de los recursos y competencias necesarias, etc., en definitiva, incrementa las oportunidades de sus miembros de participación en VE dinámicas.

No obstante, y aunque constituyen un contexto de desarrollo apropiado al solucionar muchas de las carencias descritas en los apartados anteriores, debido a su reciente “nacimiento”, tanto conceptual como práctico, son escasos los trabajos de investigación que hacen uso de él. Podemos, por tanto, enorgullecernos de ser uno de los primeros trabajos prácticos que parten de la utilización de los VBE como contexto evolutivo, lo que nos ha conducido a lograr un proceso de formación más real y menos complejo.

Por otro lado, hemos podido apreciar que los VBE existentes en la actualidad sirven a sectores específicos, tienen propósitos particulares y están restringidos a unas regiones concretas. Considerando que los entornos de gestación globales y multisector son los que ofrecen, sin lugar a duda, posibilidades más atractivas para la formación de VE ágiles, es necesario que en el futuro las empresas, Universidades, Estados, etc., realicen un gran esfuerzo en la construcción de este tipo de entornos de gestación.

En este sentido, consideramos que el primer paso que ha de darse consiste en lograr que los diferentes entornos de gestación existentes, tanto regionales como globales, interactúen entre sí permitiendo la configuración de lo que hemos denominado macroentornos de gestación (*VBME*, *VE Breeding Macro Environment*) que soporten la creación eficiente de las futuras VE dinámicas. De esta forma, conseguiremos aproximarnos cada vez más al ideal de VE dinámica que, como se recordará, está constituida por la unión de las entidades “mejor” preparadas a escala mundial.

8.3.2 La tecnología empleada

LOS SISTEMAS MULTIAGENTE Y JADE

La tecnología de agentes/sistemas multiagente está realizando importantes aportaciones en la resolución de problemas en diversos dominios (comercio electrónico, subastas electrónicas, medicina, bolsa, etc.), donde las aproximaciones tradicionales no proporcionan soluciones suficientemente satisfactorias. Las redes colaborativas, y en particular las empresas virtuales dinámicas, son uno de los dominios donde la tecnología de agentes proporciona una forma natural para resolver problemas que son inherentemente distribuidos.

En un MAS un conjunto de agentes heterogéneos se integran optimizando programas con objetivos y procesos diferentes. Dichos agentes son capaces de percibir los cambios en el entorno y se adaptan produciendo cambios en su comunidad para lograr los objetivos establecidos. Su autonomía les permite realizar tareas encaminadas a alcanzar objetivos locales, mientras que la cooperación entre los agentes posibilita la coordinación del sistema para conseguir los objetivos globales manteniendo, a su vez, la flexibilidad necesaria para adaptarse a los cambios.

Los sistemas multiagente proporcionan la base para la creación de arquitecturas menos complejas y, al mismo tiempo, más flexibles, escalables y con una mayor tolerancia a fallos¹¹⁶. Por otra parte, un MAS es inherentemente modular y abierto, lo que permite la integración dinámica de nuevos agentes, eliminación de los existentes o su actualización con diferentes funcionalidades. Además, los agentes operan de forma asíncrona pudiendo realizar trabajos en paralelo, por lo que pueden aumentar la velocidad de respuesta ante nuevas oportunidades.

Sin embargo, y a pesar de sus claras ventajas, hasta hace relativamente poco la construcción de aplicaciones con MAS era una tarea de expertos donde los agentes, la plataforma de comunicación y los protocolos de interacción se programaban de forma *ad hoc* para cada sistema. La necesidad de unificar la implementación de aplicaciones ha permitido la aparición de estándares dentro de la comunidad de investigadores de MAS, algunos tan conocidos

116 Estos sistemas mantienen su funcionalidad a pesar de fallos individuales puntuales.

como los definidos por la organización FIPA. Sin embargo, estos estándares no son suficientes para permitir que un amplio espectro de programadores o usuarios puedan desarrollar sus propias aplicaciones utilizando un sistema multiagente.

Para resolver este problema, en los últimos años se han desarrollado plataformas que facilitan el diseño e implementación de sistemas multiagente, acercando esta metodología a un público más amplio. Las diferentes plataformas de desarrollo existentes coinciden en algunas características básicas, pero sin embargo, poseen capacidades diferentes y están orientadas a distintos escenarios. Además, y a pesar de que su utilización acelera el desarrollo de las aplicaciones, el aprendizaje y uso de estas plataformas no es trivial, y requiere de un tiempo extra considerable que se suma al del desarrollo de la aplicación. Por tanto, elegir la plataforma adecuada resulta crítico.

En este sentido, JADE constituye una de las mejores opciones para el desarrollo de sistemas multiagente. JADE es una completa plataforma de agentes que simplifica el desarrollo de los agentes usando un lenguaje de programación orientado a objetos bien conocido, Java¹¹⁷. JADE implementa el estándar FIPA en muchos aspectos, desde las facilidades que debe proporcionar una plataforma FIPA (AMS, DF, registro,...) hasta la comunicación mediante mensajes ACL, por lo que un agente JADE es capaz de interactuar y comunicarse con cualquier otro agente que soporte FIPA. JADE ofrece, además, una de las mejores y más completa interfaces de comunicación entre agentes.

Otro de los aspectos sobresalientes de la comunicación JADE es que la mayoría de protocolos de interacción definidos por el estándar FIPA están implementados en un conjunto de clases. Para usarlas, sólo es necesario extenderlas, instanciarlas y generar los manejadores (*handlers*) apropiados para los distintos estados. Las plantillas de mensajes que se pasan como parámetro a estas clases permiten filtrar y seleccionar los mensajes que debe atender aquel agente que implemente un comportamiento en el que se haga uso de uno de estos protocolos.

117 El hecho de que JADE y sus agentes no dejen de ser en su totalidad código Java, implica que pueden ser ejecutados en cualquier dispositivo que cuente con una máquina virtual Java, incluidos muchos de los dispositivos de escasa capacidad como teléfonos móviles o PDA (*Personal Digital Assistant*).

Por lo que se refiere al modelo de ejecución, JADE entiende la actividad del agente como un conjunto dinámico de comportamientos que se ejecutan concurrentemente y que, en determinados momentos, pueden suspenderse o finalizarse. Siguiendo este modelo, JADE ofrece una completa familia de comportamientos que resuelven parte de la complejidad de su programación, lo que permite que nos centremos en las actividades que debe realizar el comportamiento sin tener que distraernos en detalles propios del modelo de ejecución adoptado.

Otra gran baza a favor de JADE es la variedad de herramientas que ofrece para la ejecución y depuración del sistema multiagente, y que nos han sido especialmente útiles a lo largo del desarrollo de nuestra aplicación. A parte de las herramientas propias del estándar FIPA (*AMS*, *DF*,...), JADE también ofrece otras como el *RMA*, el *Introspector*, el *Dummy Agent* o el *Sniffer* que simplifican el control y depuración de la plataforma y de sus agentes. Estos agentes han sido especialmente útiles en nuestro trabajo.

En el lado negativo, JADE carece de mecanismos para dotar a los agentes de “inteligencia”, capacidad de planificación o razonamiento. Los agentes JADE no incorporan ninguna actitud mental más allá de sus comportamientos¹¹⁸, e incluso éstos no son más que tareas, trozos de código Java a ejecutar paso a paso para lograr un resultado. El agente JADE no es más que un planificador de estas tareas, pero no como se entiende en el ámbito de la AI sino como se entiende en el entorno de los sistemas operativos. Pero, además, por defecto el agente es un único hilo Java y el planificador es aun más primario que lo que ofrece cualquier sistema operativo moderno.

Si tenemos en cuenta que la racionalidad del agente choca en muchas ocasiones con su eficiencia, y desde luego JADE es muy eficiente, probablemente muchas aplicaciones (o al menos sus agentes) no requieran ser inteligentes o sólo requieran emanar comportamiento inteligente desde una visión más tradicional. En estos casos, cuando los agentes no razonan sino que ejecutan y necesitan de habilidades sociales, JADE destaca de un modo significativo. Sin embargo, cuando la inteligencia y la racionalidad son características deseables, como en nuestro caso, JADE es insuficiente.

118 Los agentes JADE no son inteligentes: la única inteligencia que emana de ellos es la que les pueda dotar el programador a través de los comportamientos, pero ellos por si solos no son capaces de razonar acerca de cuál es la forma de lograr un objetivo o de si su información acerca del mundo es contradictoria.

Para solventar esta necesidad tenemos a nuestra disposición varias opciones. En primer lugar, las características tan notables de esta plataforma han propiciado que sea el punto de partida de otros agentes que funcionan sobre JADE y que sí ofrecen racionalidad y actitudes mentales tal y como se entiende en el ámbito de los agentes. Este es el caso de Jadex, una herramienta de razonamiento basada en el modelo BDI. Por otro lado, el que JADE tenga una base Java le permite interactuar de un modo relativamente fácil con implementaciones Java de Prolog o de sistemas expertos. Precisamente, esta última posibilidad ha sido la opción adoptada en la presente tesis al considerar más acertado separar el módulo de decisión del agente de su propia implementación, lo que simplifica notablemente tanto su gestión (p.e., introduciendo nuevas reglas o cambiando las existentes) como su futura mejora (a través, p.e., de la incorporación de la lógica CBR).

LOS SISTEMAS EXPERTOS Y JESS

Un ES es un sistema informático capaz de emular las prestaciones de un experto humano en un área concreta de conocimiento. En otras palabras, es un sistema capaz de almacenar el conocimiento de un especialista y solucionar problemas mediante la inducción-deducción lógica. Esta capacidad explica nuestra inclinación hacia su uso, ya que nos permite representar el proceso de selección que se da en las empresas de una manera mucho más real que cuando se considera sólo un conjunto de enrevesados algoritmos y funciones matemáticas.

Además, posee ciertas características que se han revelado especialmente útiles. Por ejemplo, el sistema experto permite mejorar la productividad del sistema al resolver los problemas más rápidamente, lo que se traduce en un ahorro de tiempo y recursos considerable. En un entorno tan dinámico y volátil como el que se enfrenta la VE dinámica, en el que la agilidad en la respuesta es una obligación más que una cualidad, puede resultar que sin ese incremento de rapidez que proporciona el sistema experto las soluciones obtenidas fuesen inútiles.

Por otro lado, los valiosos conocimientos de un especialista se guardan y se difunden, de forma que no se pierden aunque desaparezca el especialista. Así, se guarda la esencia de los problemas que se intenta resolver y se programa cómo aplicar los conocimientos para su resolución. Esto permite que personas no especializadas puedan utilizar el ES para resolver problemas y

que incluso sean capaces de aprender de él si lo utilizan regularmente. El beneficio es especialmente útil cuando existen empleados que necesitan conocer la política de la empresa en el proceso de selección.

El sistema experto también permite obtener soluciones más fiables, gracias al tratamiento automático de los datos, y más contrastadas, debido a que se suele tener informatizado el conocimiento de varios expertos. Gracias a ello, se pueden desplazar recursos humanos, los especialistas, hacia otros procesos de la empresa donde sean más necesarios, con lo que logramos optimizar los recursos durante la formación de la empresa virtual dinámica.

El principal escollo que hemos encontrado en la construcción de este tipo de sistemas se halla en que el conocimiento humano es bastante complejo de extraer (las reglas no siempre son obvias), e incluso su representación resulta bastante problemática. A pesar de que eramos conocedores de esta realidad, la pérdida de tiempo que supone la extracción y representación del conocimiento es bastante considerable, si bien las aportaciones que realiza el sistema experto a la plataforma hacen que merezca la pena.

En cuanto a la herramienta empleada, Jess ha demostrado ser un motor de reglas muy rápido. Además, es pequeño, ligero, y su lenguaje de *scripting* permite un fácil acceso a todas las API de Java. Gracias a ello hemos podido construir agentes con capacidad para obtener resultados inteligentes, usando conocimiento expresado en forma de reglas declarativas. La principal pega es que, como sistema basado en reglas, se obtiene un resultado “inteligente”, pero para llegar a él no se realiza ningún tipo de razonamiento humano, simplemente se hacen emparejamientos de reglas preestablecidas.

Para nuestros propósitos este comportamiento nos es totalmente suficiente, ya que desde un primer momento tan sólo hemos pretendido emular el proceso de toma de decisiones de los directivos de las empresas. No obstante, puede argumentarse que un directivo o un experto humano son capaces de decidir de inmediato cuándo se enfrentan a una situación análoga a otra ocurrida en el pasado. Los ES basados en reglas no utilizan este razonamiento por analogía, aunque otros sistemas, como los basados en casos, pueden ser útiles en tales situaciones.

A partir de todo lo analizado anteriormente, se deduce que una gran parte de la potencia que presenta nuestra plataforma procede, en definitiva, de la utilización del ES. En particular, permite que el conocimiento del agente

crezca progresivamente a través de la adicción de nuevos hechos, y que su comportamiento se vea modificado por la incorporación de nuevas reglas o la modificación de las existentes. Creo que la mejor forma de mostrar este potencial es a través del ejemplo que proporciona E.J. Friedman-Hill (figura 8.12) en su tutorial sobre Jess (Friedman-Hill, 2007).

Imagina que estás trabajando en un motor de tasación para ventas online. Se supone que el motor mira cada orden, junto con el historial de compra de un cliente, y aplica varios descuentos y ofertas a la orden. Imagina además que has codificado esto en una clase Java tradicional. Tu jefe entra y dice, “da un descuento del 10% a todos que pasen de \$100”. Fantástico. Agregas una simple declaración si-entonces, recompilas, y de vuelta al trabajo. El jefe vuelve a entrar y dice, “da un descuento del 25% en artículos a los clientes que compren tres o más.” Otros si-entonces, otra recompilación. El jefe vuelve a entrar y dice “revoca la orden del descuento del 10%, haz el 10% si pasan de \$250 en un mes. También, si alguien compra un regrabador de CD, envíale una muestra gratuita de discos CDRW; pero solamente si es un cliente que ha comprado otras veces. Oh, y a propósito, necesitamos tasar el envío en base a la zona geográfica, a excepción de las personas con múltiples localizaciones ...”

Después de algunas semanas así, si has estado utilizando técnicas de programación tradicionales, tu código es un lío además de lento, ya que se deben hacer toda clase de preguntas a la base de datos para cada orden que entre. Si se hubiera escrito esto usando un motor de reglas, tendrías un código limpio y agradable, con una regla para cada política de tasación. Si alguien necesitara saber dónde está implementado la política de tasación “miércoles Wacky”, sería fácil encontrarla. Y si ese motor de reglas es un sistema como Jess, tampoco será lento; el propio motor indexa todas las órdenes y no se requiere ninguna operación de búsqueda; el motor de reglas sólo “sabe” lo qué necesita saber sobre las órdenes pasadas.

Figura 8.12. Utilidad de Jess (Friedman-Hill, 2007).

Estas líneas son plenamente validas también en nuestro caso: podemos vetar empresas, incluir descuentos en función del tipo de cliente, incorporar nuevas restricciones al sistema, establecer preferencias, etc., y todo con la facilidad y limpieza que proporciona, en este caso, Jess.

LAS ONTOLOGÍAS

En esta tesis hemos considerado que una ontología es una entidad computacional, un recurso artificial que se crea para contener el entendimiento común y compartido de un dominio. En este sentido, las ontologías son un elemento fundamental en la comunicación entre agentes ya

que permiten retratar el conocimiento común del dominio considerado. Si los agentes que intervienen en la comunicación utilizan contextos distintos (es decir, ontologías distintas) o términos ambiguos, la comunicación puede ser un desastre¹¹⁹.

Sin embargo, esta visión tan comúnmente aceptada de la ontología como una “formalización de una conceptualización”¹²⁰ es discutible. De hecho, existen ciertos dominios, como el de la empresa, donde la forma de ver el mundo es, básicamente, una declaración de principios sobre cómo deben ser los procesos de negocio y la documentación. En estos casos, una vez diseñada la ontología, el responsable pretenderá que el proceso real de gestión del negocio y de la documentación se ajuste al modelo creado. De esta forma, la ontología se utiliza explícitamente para cambiar el mundo objeto de estudio y no para describirlo.

Una conclusión de esta idea es que las ontologías sobre empresa como por ejemplo ENTERPRISE, TOVE, REA y BMO, entre otros, son difícilmente alineables con ontologías que extraen directamente el conocimiento de una empresa (ONTOBLOGIA web, 2007). Una de las razones más importantes para esta dificultad de conciliación del conocimiento es, quizás, que las bases de datos de la empresa contienen tal cantidad de conocimiento oculto, intangible, sobre la empresa que sólo se puede conocer una vez completado el proceso de integración. Y parte de ese conocimiento es muy importante y específico (tanto que se puede incluso considerar un valor intangible) así como muy útil para la empresa concreta.

En consecuencia, las ontologías para la empresa no las pueden hacer sólo los académicos. Es necesario contar obligatoriamente con un experto en los dos ámbitos (ontologías y empresa). Y lo que es más importante: no es sencilla, y en algunos casos ni siquiera relevante, la relación entre ontologías para la empresa desarrolladas en diversos proyectos (como los mencionados en el párrafo anterior) con una diseñada para una empresa de tamaño medio y con necesidades específicas. Existen, incluso, dificultades para relacionar correctamente algunas clases.

119 Cuando la comunicación está ausente de contexto la interpretación de la performativa intencional del mensaje no está disponible para el receptor (a menos que el emisor lo haga explícito), lo que conduce a una mala interpretación de la información y a posibles discusiones sobre el contenido del mensaje.

120 Donde por conceptualización se puede entender una forma de ver el mundo, centrada en aspectos interesantes para el diseñador.

Las mismas consideraciones se pueden efectuar en relación a la formación y gestión de las VE dinámicas, ya que en última instancia son empresas, están formadas por empresas y comparten muchos de los procesos de negocio de las empresas tradicionales. En consecuencia, es necesario contar con expertos de las diferentes compañías (y sus procesos de negocio) y expertos en ontologías que sean capaces de fusionar sus esfuerzos para mejorar y ampliar las ontologías existentes, adaptándolas tanto a los diferentes tipos de empresas existentes como a la idiosincrasia de la VE dinámica y los VBE. Sólo de este modo conseguiremos disponer de las mejores herramientas en las diferentes fases de su existencia.

8.3.3 Los logros conseguidos

Una vez analizados los elementos que han tenido un papel clave en la presente memoria, y tras haber estudiado la tecnología empleada, llega el momento de discutir los logros alcanzados. Obviamente, éstas metas son consecuencia directa de los objetivos establecidos al iniciar la investigación y, al mismo tiempo, del conjunto de pasos que se definieron con objeto de alcanzar su plena consecución.

EL MARCO TEÓRICO

Uno de los resultados más interesantes del trabajo desarrollado en este tiempo ha sido la elaboración de un marco teórico muy completo sobre el proceso de selección de socios en las VE dinámicas. Efectivamente, a lo largo de los tres primeros capítulos de la memoria se han ido mostrando los aspectos más importantes necesarios para comprender nuestras motivaciones y propósitos. Estos conocimientos se han ido introduciendo de forma progresiva, desde los más globales a los más particulares, tal y como queda reflejado en la figura 8.13.

Así, el primer eslabón de este “estado del arte” ha sido la descripción del contexto que circunscribe la actividad de las empresas de comienzos del siglo XXI, y que ha propiciado la necesidad de nuevos modelos de negocio en los que la adaptabilidad y la agilidad se consideren prioridades competitivas. Estos nuevos modelos surgen básicamente como respuesta a una necesidad propia de cualquier ser vivo, la supervivencia, sobre todo en unos tiempos de elevado dinamismo y de una creciente competitividad y globalización.



Figura 8.13. Progresión del marco teórico expuesto en la presente memoria.

El enorme potencial del modelo organizativo de VE le permite destacar significativamente sobre el resto de conceptos relacionados. Sin embargo, y como suele ocurrir en los inicios de cualquier paradigma, está acompañado de una gran confusión sobre su definición y los elementos que la componen o definen. En esta tesis hemos intentado eliminar esta confusión:

- Para ello, primero hemos diferenciado el modelo de empresa virtual de otros conceptos que aparecen en la literatura (volviéndola más confusa) y que en ocasiones se utilizan como sinónimos, aunque posean un significado distinto e involucren componentes y estrategias de negocio diferentes.
- Por otro lado, y en base al gran número de definiciones existentes sobre este concepto, se ha establecido un conjunto de criterios que permiten dilucidar claramente qué es la VE y, además, qué tipos existen. Para cada uno de ellos se han expuesto sus características y se han presentado ejemplos representativos. De este estudio se ha podido comprobar que las VE dinámicas son las más atractivas para responder a la turbulencia y competitividad propias de nuestros días, aunque también requieren soluciones tecnológicas más avanzadas.

El capítulo 2 nos ha permitido centrar el estudio en este modelo dinámico y, en particular, en su ciclo de vida. De esta forma hemos podido conocer mejor sus características y los mecanismos con los que cuenta. Como no podía ser de otro modo, en este estudio hemos prestado una especial atención a aquellas etapas en las que el proceso de selección de socios constituye un elemento clave, como son la formación y reconfiguración del consorcio. Pero no sólo hemos podido conocer su proceso vital sino también dónde se da o, al menos, dónde hemos considerado que se producirá.

Efectivamente, la existencia de las VE dinámicas tiene lugar dentro de un entorno concreto. Como se recordará, uno de nuestros aciertos más importantes ha sido la utilización de un concepto de muy reciente definición, los entornos de gestación de VE, para representar este contexto evolutivo (figura 8.14). Constituir uno de los primeros desarrollos prácticos que hacen un uso efectivo del mismo es un motivo más de satisfacción. En este capítulo también se han descrito los principales roles involucrados en la existencia de la DVE, incluidos los más representativos de estos VBE.

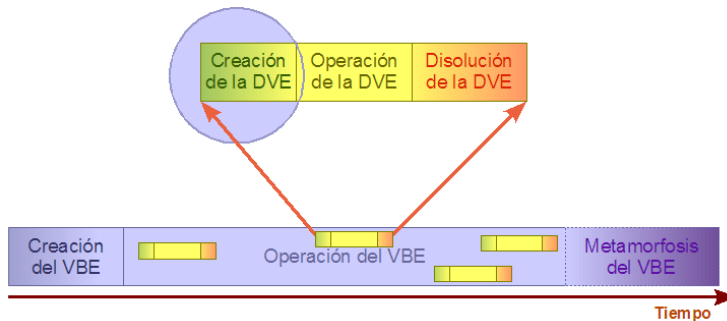


Figura 8.14. La formación de las DVE dentro de los VBE.
(Camarinha-Matos *et al.*, 2005c)

El último de los capítulos se ha dedicado a describir y analizar la forma en que la comunidad investigadora ha afrontado el problema de la selección de socios en los últimos años. Por ese motivo, se ha comenzado con una descripción de los principales proyectos I+D+i con la pretensión de conocer los distintos enfoques empleados. A continuación, se han descrito cada una de estas orientaciones; en concreto, el análisis cuantitativo, los sistemas multiagente y los servicios web. Su estudio ha sido clave en la elección de la tecnología que a día de hoy era más recomendable para verificar la idoneidad del modelo que estábamos a punto de concretar.

EL MODELO

Desde tiempos inmemoriales los investigadores y científicos han pretendido estudiar el comportamiento de los sistemas que les rodean. En este estudio han jugado un papel crucial los “modelos” como instrumentos para poder explicar lo complejo y, en definitiva, simplificar la realidad. Partiendo de esta idea, la presente tesis surgió con la simple motivación de proporcionar un modelo del proceso de selección de socios en las VE dinámicas que constituyese nuestra pequeña aportación al desarrollo de este revolucionario modelo organizativo.

La importancia de definir correctamente este modelo era crucial, pues debía servir como punto de partida en la construcción de herramientas informáticas capaces de responder de manera eficiente a las necesidades propias de la formación y reconfiguración de las VE dinámicas. Por este motivo, se estableció la premisa de construir un modelo innovador, realista y completo (en el sentido de dar cabida tanto a las distintas orientaciones existentes como a otras posibles) pero que, al mismo tiempo, fuese lo más sencillo posible.

Partiendo de un estudio y análisis riguroso de las distintas orientaciones o enfoques presentados hasta el momento (desde el planificador al competidor, pasando por el jerárquico, etc.), de sus ventajas e inconvenientes, de los resultados de su aplicación, etc., hemos sido capaces de ofrecer una solución completa e innovadora, que permite crear VE dinámicas siguiendo los enfoques existentes en la literatura, aunque no se queda sólo ahí. De hecho, nuestro modelo multienfoque es capaz de crear todo tipo de DVE, con el nivel de control que se precise en cada situación.

Pero además, la solución definida es sencilla tal y como sugiere el conocido principio KISS, que plantea la simplicidad como algo imprescindible en cualquier diseño. Este principio, que se basa a su vez en el principio de la navaja de Occam¹²¹ y la máxima de Albert Einstein “*everything should be made as simple as possible, but no simpler*”, nos ha exigido un gran esfuerzo encaminado a reducir nuestro modelo justamente a aquellas piezas que se requerían para hacerlo trabajar, rechazando lo enrevesado e innecesario.

121 La navaja de Occam hace referencia a un tipo de razonamiento que se basa en una premisa muy simple: en igualdad de condiciones, la solución más sencilla es probablemente la correcta.

En resumen, se ha creado un modelo (figura 8.15) sencillo que permite la formación de cualquier tipo de VE dinámica (figura 8.16) al dar cabida a cualquiera de los enfoques existentes en la literatura. Este comportamiento innovador se consigue, sencillamente, al dar la posibilidad a los clientes o a las empresas de publicar sus necesidades en el entorno de gestación o bien solicitar al VBE la lista de los posibles suministradores del servicio demandado.

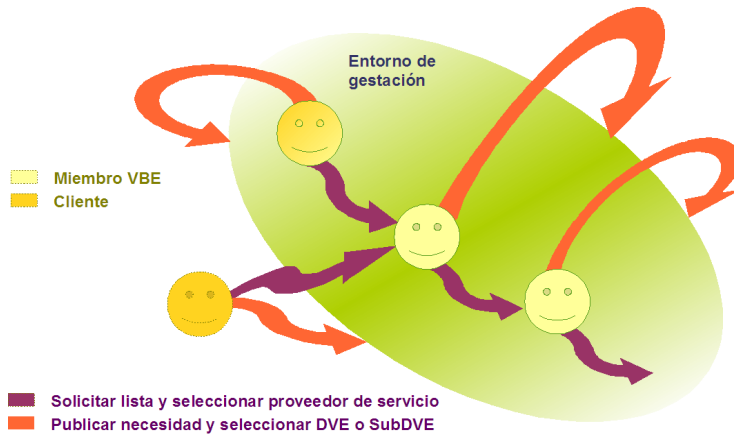


Figura 8.15. El modelo de formación de la DVE desarrollado en la presente tesis.

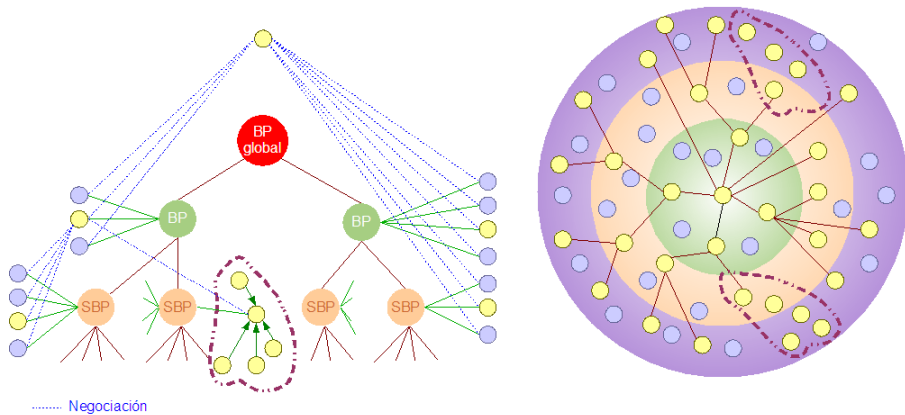


Figura 8.16. Ejemplo de DVE constituida a partir de nuestro modelo multienfoque.

Esta simple elección hace posible la formación de DVE jerárquicas, centralizadas, descentralizadas, etc.; permite crear VE dinámicas que se ajusten a la definición pura, VE dinámicas controladas por un miembro, DVE federadas, VE en las que se establezcan diferentes tipos de control en todos y cada uno de los procesos de negocio, o en unos pocos, etc. En resumidas cuentas, nos da la posibilidad de crear cualquier tipo de VE dinámica que uno se imagine, y con el control que se precise en cada momento.

El resultado parece satisfactorio: un modelo del proceso de selección de DVE multienfoque, global, innovador y sencillo, que amplía las posibilidades de los modelos existentes en la literatura y que reduce considerablemente sus carencias.

LA PLATAFORMA

Para comprobar que el modelo responde como se espera y que permite representar correctamente lo que ocurre durante el proceso de selección de socios hemos desarrollado una aplicación informática. En su elaboración hemos intentado seguir los pasos que definen el ciclo de vida de toda aplicación software: planificación, análisis, diseño, construcción y, finalmente, validación.

Teniendo presente la investigación vinculada al proceso de selección de socios, la planificación nos ha permitido, por un lado, justificar las tecnologías que posteriormente hemos empleado y, además, seleccionar el conjunto de herramientas más adecuadas a nuestros objetivos, motivaciones y principios de actuación. La tecnología multiagente y los sistemas expertos se han impuesto como la mejor solución tecnológica del problema, del mismo modo que JADE y Jess han sido seleccionadas como las herramienta más adecuadas para construir, respectivamente, la aplicación multiagente y el módulo de decisión de los agentes.

A continuación, y siguiendo el conjunto de indicaciones establecidas en la metodología para JADE (Nikraz *et al.*, 2006), hemos logrado clarificar el problema a un nivel de detalle suficiente a través de 6 pasos. El resultado de esta fase de análisis han sido los diagramas de casos de uso, el diagrama de agentes, la tabla de responsabilidades y el diagrama de despliegue de agentes, que convenientemente refinados han constituido la base para la siguiente fase de la metodología, la de diseño.

El primer rasgo distintivo de la plataforma procede, por tanto, de la *conjunción de dos tecnologías* tan innovadoras e interesantes como son los sistemas *multiagente* y los *sistemas expertos*. Aunque desde su nacimiento ambos campos del conocimiento han demostrado un enorme potencial por sí solos, son pocas las ocasiones en las que han mostrado un comportamiento simbiótico, y más pequeño aún es el número de trabajos en los que se hace un uso combinado de las herramientas JADE y Jess. De hecho, este trabajo es el primero que aúna ambas herramientas para dar solución al problema de la selección de socios en los modelos dinámicos de VE. Se trata, por tanto, de una solución realmente *innovadora* desde el punto de vista *tecnológico*.

Pero también es *innovadora* desde el punto de vista *conceptual*. Atendiendo al modelo definido para representar el proceso de selección de socios, los usuarios pueden *definir el control sobre los procesos de negocio remotos*: las empresas y los clientes pueden seleccionar a sus proveedores a partir de la lista que suministra el agente administrador, lo que conduce a un control más estricto y a la selección de una única empresa, o bien pueden optar por publicar sus necesidades en la plataforma, lo que se traduce en un control más laxo y en la selección final de una VE dinámica.

El sistema es, además, *escalable* al permitir la entrada y salida de agentes del sistema. El único requisito para poder acceder a la plataforma consiste en pertenecer al entorno de gestación: de este modo, la empresa comparte información sobre sí misma que cualquier agente podrá utilizar siempre que el desempeño de su labor lo precise. Por otro lado, un agente puede dar de baja algún proceso de negocio en particular o abandonar la plataforma (lo que equivale a dar de baja todos los servicios que suministra).

Tanto los agentes empresa como los cliente presentan un *comportamiento "inteligente"* gracias, como sabemos, a la integración del sistema experto Jess. Cada agente cuenta con una memoria de trabajo y un conjunto de reglas que definen su actuación cada vez que se activa el motor de inferencia. *El comportamiento del sistema construido puede ser fácilmente modificado*, ya que se pueden cambiar tanto las reglas como los hechos que le definen. Además, permite que se tengan en cuenta tanto factores técnicos como subjetivos (confianza, preferencias, vetos, etc.).

La *comunicación* entre los agentes es *asíncrona*, gracias a los mecanismos de comunicación soportados por la plataforma FIPA: el intercambio de mensajes se realiza a través del ACC definido en este estándar. La comunicación

asíncrona es un requisito clave dado que las diferentes entidades no deben tener referencias estáticas entre ellos (como ocurre en cualquier sistema de componentes distribuidos). El contenido del mensaje intercambiado se describe en *ontologías abiertas y flexibles*.

Gracias a la comunicación asíncrona de los agentes se logra otra característica interesante y clave en la gestión dinámica de las DVE: la *autonomía*. En el contexto de esta tesis, la autonomía del sistema se ve profundamente favorecida al permitir que los agentes se comuniquen mediante el intercambio de mensajes que siguen distintos protocolos FIPA. El uso de tecnologías abiertas, interoperables, estándar como FIPA, XML o Java conduce a un aumento de la *transparencia* del sistema.

Otro aspecto interesante es la *modularidad* con la que se ha diseñado y construido la aplicación. Aunque nosotros tan sólo hemos considerado un único dominio de aplicación, la incorporación de nuevos dominios se puede realizar de forma fácil e intuitiva: tan sólo requiere reutilizar muchos de los elementos construidos para el dominio actual (p.e. las clases de la interfaz gráfica), y definir el resto (ontologías, documentos XML, etc.) de forma similar a la descrita en este documento. En definitiva, *la incorporación de nuevos procesos de negocio y dominios de aplicación es trivial*.

Por lo que se refiere a la *interfaz de usuario*, se ha buscado en todo momento que sea lo más *simple* posible, y por ello hemos dotado a los agentes de la funcionalidad básica que precisan. De hecho, las facilidades para la gestión de los documentos Jess son muy reducidas ya que creemos más acertado que dicha gestión se realice mediante una herramienta *ad hoc* como es JessDE. Pero, además, la interfaz es muy *intuitiva* lo que facilita notablemente tanto su utilización como la navegación.

Otra característica interesante de la GUI es que el usuario tiene un *fácil acceso a la información generada* durante el transcurso de la actividad de su agente. Además, el administrador de la plataforma tiene la posibilidad de acceder a los agentes de gestión de JADE en cualquier momento para, por ejemplo, monitorizar el estado de la plataforma o conocer los servicios registrados en el agente páginas amarillas.

En definitiva, se ha diseñado y desarrollado una aplicación *flexible y dinámica* que refleja el dinamismo del concepto de DVE y la agilidad que debe poseer este tipo de modelos organizativos.

USO DE CÓDIGO LIBRE

Son numerosas las organizaciones, instituciones,... , e incluso comunidades autónomas y países, que recomiendan la utilización y el fomento del software libre. Entre las causas de este fenómeno destaca el ahorro que conlleva. De hecho, recientemente se ha publicado que el software libre supone unas pérdidas de 60.000 millones de dólares para las compañías de software propietario como Microsoft¹²². Aunque esta cifra sólo representa un impacto económico del 6% del total estimado del presupuesto de las ICT en software, implica que las empresas están ahorrando 60.000 millones de dólares en informática gracias al software libre, proporcionando gran flexibilidad a las empresas y administraciones.

Teniendo presente este ahorro, cuando definimos los objetivos de la tesis establecimos como principio de actuación la utilización de software libre o, en su defecto, software con licencia para uso académico. Este principio ha guiado nuestro trabajo tal y como pone de manifiesta la figura 8.18 donde se representan los logos de las aplicaciones utilizadas en este proyecto.



Figura 8.18. Algunos logos representativos de las distintas herramientas FS empleadas.

122 Según pone de manifiesto el reciente informe de The Standish Group (StandishGroup web, 2008), del 16 de abril de 2008.

Como se observa, hemos usado la suite ofimática Open Office para la elaboración del presente documento y los manuales; GIMP (*GNU Image Manipulation Program*) para la edición de gráficos e imágenes; el navegador FireFox y el cliente de correos Thunderbird de Mozilla; la IDE Netbeans para la construcción de la aplicación multiagente, usando para ello las librerías que proporcionan tanto JADE como Jess; Protégé para las ontologías, además de los *plugins* JessTab y BeanGenerator; y JessDE, junto con Eclipse, para la elaboración de la memoria de trabajo y las reglas Jess que definirán el comportamiento de los agentes.

De todas ellas, tan sólo Jess necesitaba licencia. Por ese motivo, quiero volver a agradecer a Craig A. Smith (Software Licensing Manager de Sandia National Laboratories) todas las facilidades que nos ha dado para poder utilizar el código para su uso académico y de investigación.

El único pero a este principio de actuación lo constituye el uso de Windows XP como uno de los dos sistemas operativos, el otro ha sido Ubuntu (figura 8.19) que sí es FS, sobre los que hemos desarrollado nuestra actividad. Obviamente, es muy difícil substraerse completamente al sistema operativo más utilizado por los usuarios informáticos (tanto particulares como empresas) y, en consecuencia, el que más se instala en los ordenadores.



Figura 8.19. Anagrama del sistema operativo Ubuntu.

8.4 RESUMEN

Una vez construida la aplicación multiagente, llega el momento de validar tanto el modelo de partida como la plataforma. Dicha validación se ha realizado a lo largo de la primera parte del capítulo, utilizando, para ello, el proceso de negocio ficticio descrito en el capítulo 6. A través de la definición de varios escenarios se ha comprobado el correcto funcionamiento de la plataforma, lo que permite establecer la adecuación de las tecnologías empleadas en su construcción y la validez del modelo que constituía la base de su desarrollo.

Finalmente, en la segunda parte del capítulo se han valorado todos los elementos clave que han intervenido en el resultado final de la memoria, desde la VE dinámica o los entornos de gestación hasta el trabajo realizado, pasando por las tecnologías empleadas en su construcción. La atención principal se ha puesto, como no podía ser de otra manera, en el modelo y la aplicación desarrollada para su validación y valoración, mostrando sus características más distintivas.

En concreto, podemos concluir que hemos desarrollado un modelo muy realista y completo, además de innovador y simple. Basándonos en él hemos desarrollado la herramienta informática con la que responder de forma eficaz al problema de la selección de socios en las DVE. Esta herramienta es flexible, dinámica e innovadora tanto desde el punto de vista tecnológico (al unir dos tecnologías como son los MAS y los ES) como desde el conceptual (fruto de su absoluta conformidad con el modelo inicial). Es escalable, la comunicación es asíncrona, se respetan los estándares FIPA, los agentes son autónomos y muestran un comportamiento inteligente, la interfaz es intuitiva, la reutilización de su código es muy fácil, etc. En definitiva, lo que buscábamos, y mucho más, en una aplicación software de este estilo.

CONCLUSIONES Y EXTENSIONES FUTURAS

INTRODUCCIÓN

En la década de los 80, y como consecuencia de una creciente globalización de los mercados, muchas de las grandes corporaciones integradas empezaron a darse cuenta de que carecían de la flexibilidad y agilidad necesarias para seguir siendo competitivas. Unos años más tarde, a principios de los 90, Internet empezó a utilizarse comercialmente, y numerosos teóricos e investigadores del mundo de los negocios lo abrazaron con entusiasmo al reconocerlo como el motor que conduciría el desarrollo de una nueva generación de modelos de negocio más prometedores.

Todos presagiaban redes electrónicas que posibilitarían un acceso sin precedentes tanto a los mercados como a los suministradores globales en una nueva era de los negocios basada en la información. Desde entonces, la “empresa virtual” y su concepto como red temporal de empresas independientes, geográficamente dispersas y dispuestas a colaborar, y en la que las tecnologías de la información y la comunicación desempeñan un papel clave, empezó a adquirir una creciente difusión.

Sin embargo, la tecnología requerida para crear y configurar este tipo de empresas no estaba disponible en ese momento. Además, y a pesar de los esfuerzos de numerosos investigadores y expertos, el concepto de empresa virtual permanecía bastante difuso y tampoco existía un acuerdo sobre la terminología a utilizar. En contraste, los modelos *Business to Consumer* (B2C) y *Business to Business* (B2B), conocidos como *e-Commerce*, eran alcanzables y prometían asombrosos retornos financieros, lo que se tradujo en una proliferación sin precedentes de las empresas “punto-com” a finales de los noventa.

La mayoría de estos modelos de compra-venta prestaron una excesiva atención a la tecnología creyendo que se trataba de la solución a la mayoría de las cuestiones de negocio. El conocido desplome de las “*dot-com*” vino a desmentir esta creencia y provocó una reorientación de los modelos de negocio comenzando el despliegue de tecnologías basadas en red tanto dentro de la empresa como entre la empresa y sus suministradores y clientes. Los modelos de negocio resultantes incluyen a los modelos de *e-Commerce*¹²³ y se conocen colectivamente como *e-Business* o economía digital.

De entre todos, la VE dinámica se atisba como el modelo de negocio más prometedor del siglo XXI y muchas empresas ya comienzan a encaminarse hacia su adopción. Sin embargo, a día de hoy todavía son más ficción que realidad debido, en su mayor parte, al gran número de problemas, desafíos y riesgos inherentes a este modelo. En un intento por conseguir su aplicación práctica a gran escala, en los últimos años han proliferado los proyectos y trabajos de investigación en los que se trata de dar respuesta a tales desafíos. En este sentido, numerosos autores han identificado la correcta selección de los socios como el factor clave en el éxito de la empresa virtual dinámica.

La manera más eficaz de contribuir a su desarrollo práctico consiste, por tanto, en aportar una solución realista e innovadora a tan complejo problema. Con este fin nació el presente trabajo doctoral, cuyo resultado más significativo ha sido la elaboración de un modelo operativo del proceso de selección de socios que responde de un modo efectivo a sus necesidades aunando las distintas orientaciones propuestas hasta el momento, lo que hace de él un modelo innovador y realista. Además, en base a este modelo se ha construido una plataforma software que hace uso de dos tecnologías tan innovadoras como son los sistemas multiagente y los sistemas expertos.

123 Tratan, por tanto, tanto aspectos operacionales como de compra-venta.

El modelo y la aplicación multiagente constituyen, por tanto, los resultados más interesantes de la presente tesis doctoral, aunque no son los únicos. Por este motivo, se ha definido el apartado “conclusiones” con la clara intención de precisar los logros conquistados en este tiempo y valorar los elementos más importantes que han intervenido en su consecución. En esencia, este apartado recoge y sintetiza los aspectos más importantes de la memoria desde la perspectiva que definen tanto el objetivo inicial de la tesis como el conjunto de pasos definidos para lograr su plena consecución.

Pero el trabajo realizado ha permitido, a su vez, identificar las líneas principales de una interesante y prometedora investigación futura. Estas “extensiones” las hemos agrupado en tres grandes bloques atendiendo, principalmente, a su propósito y alcance.

- Así, primero están aquellas actuaciones encaminadas a mejorar la plataforma actual y a dotarla de una mayor funcionalidad. Entre estas se encuentran la incorporación de nuevos agentes de servicio a la arquitectura distribuida así como la mejora de los ya existentes, la definición de nuevas reglas, el uso de los módulos Netbeans, etc.
- A continuación, se encuentran aquellas extensiones orientadas hacia el uso de nuevas herramientas o tecnologías. En este sentido, tienen una especial relevancia la tecnología JADE-Leap, el razonamiento basado en casos y la unión de los MAS con los servicios web.
- Finalmente, se aborda la intención de buscar una mayor aplicación práctica del trabajo ya no sólo en el ámbito empresarial, integrando nuestra investigación con el resultado de una tesis presentada recientemente, sino incluso en el ámbito social, donde ha surgido una colaboración para favorecer el trabajo en red dirigido a detectar, prevenir y solucionar la actual problemática del *bullying*.

CONCLUSIONES

Como se recordará, la presente tesis nació con la clara intención de construir un modelo operativo y realista del proceso de selección de socios en las VE dinámicas a partir del cual poder desarrollar herramientas de soporte eficientes, innovadoras y plenamente satisfactorias. Con este propósito se definieron una serie de pasos que delimitaban el camino a seguir y que, al mismo tiempo, nos van a servir de guía para establecer las principales conclusiones del trabajo desarrollado.

- a) *Confección de un “estado del arte” sobre la empresa virtual dinámica y la problemática asociada al proceso de selección de socios, estudiando los diferentes enfoques planteados para su resolución.*

Los investigadores en el campo de la VE siempre han considerado que uno de los requisitos más importantes que debe cumplirse para conseguir que el concepto de VE dinámica llegue a tener éxito a gran escala consiste, sencillamente, en definir de forma clara qué es la DVE y cuáles son sus ventajas, componentes, mecanismos, etc. Por este motivo, nuestros esfuerzos iniciales se orientaron hacia la concepción de un marco teórico lo más completo posible sobre la DVE en general, y sobre el proceso de selección de socios en particular, como un medio para contribuir de forma significativa a aumentar la aceptación de este campo de investigación.

Y en este sentido, podemos concluir que uno de los resultados más interesantes del trabajo desarrollado ha sido la elaboración de este marco teórico. Precisamente, este “estado del arte” se ha erigido en un elemento clave de nuestra investigación, no sólo porque favorece una mejor comprensión de nuestras motivaciones y propósitos, sino porque ha servido de guía tanto en la confección de un modelo real y operativo del proceso de selección como en la construcción de la herramienta software que debía verificar su idoneidad.

Dentro de este marco teórico conviene resaltar la inclusión de un concepto de reciente aparición, los entornos de gestación de DVE (estos consorcios constituyen el contexto en el que desarrollan su actividad las VE dinámicas y del que se nutren de socios para realizar su labor). Tanto en nuestro modelo como en la aplicación desarrollada este concepto ha sido un elemento fundamental. Y dado que nuestro trabajo constituye uno de los primeros desarrollos prácticos que emplean este concepto de un modo efectivo, podemos considerarle como pionero.

También hemos establecido la conveniencia de lograr en el futuro que los diferentes entornos de gestación existentes, tanto regionales como globales, interactúen entre sí permitiendo la configuración de lo que hemos denominado macroentornos de gestación (VBME, *VE Breeding Macro Environment*) que soporten la creación eficiente de las futuras VE dinámicas. Sólo a través de estos macroentornos será posible un mayor acercamiento al ideal de VE dinámica constituida, como se recordará, por la unión de las entidades “mejor” preparadas a escala mundial.

Por último, el estudio de los principales trabajos y proyectos de investigación que han abordado la problemática de la selección de socios en los últimos años, ha sido clave en la elección de la tecnología que, a día de hoy, era más recomendable para construir una herramienta informática con la que verificar la idoneidad de nuestro modelo. En efecto, esta investigación permitió descubrir que el análisis cuantitativo del problema constituía un enfoque obsoleto y alejado de la realidad, y que los servicios web presentan multitud de limitaciones para poder aplicarse eficientemente a la selección de socios, lo que situaba a los sistemas multiagente como la mejor opción.

b) Elaboración de un modelo realista y funcional del proceso de selección de socios en las VE dinámicas.

Los “modelos” constituyen para el ser humano una herramienta clave con la que lograr una mejor comprensión de la realidad y de su complejidad. En este sentido, la presente tesis nació con el propósito de proporcionar un modelo del proceso de selección de socios en las DVE que sirviese de punto de partida en la construcción de soluciones informáticas más afines a la realidad de las VE dinámicas. El objetivo puede considerarse alcanzado mediante la construcción de un modelo innovador, realista, global y operativo, que constituye nuestra pequeña aportación al desarrollo de este revolucionario paradigma organizativo.

Partiendo del estudio y análisis riguroso de las distintas orientaciones o enfoques presentados hasta el momento (desde el planificador al competidor, pasando por el jerárquico, etc.), de sus ventajas e inconvenientes, de los resultados de su aplicación, etc., hemos sido capaces de ofrecer un modelo global a partir del cual concebir y crear cualquier tipo de VE dinámica y con el nivel de control necesario en cada situación lo que, en definitiva, nos permite englobar no sólo los distintos enfoques existentes en la literatura sino también muchos otros.

Pero, además, la solución definida es sencilla, tal y como sugiere el conocido principio KISS que plantea la simplicidad como algo imprescindible a la hora de diseñar cualquier modelo, proceso, etc. Basar nuestra actuación en este principio nos ha exigido un gran esfuerzo encaminado a reducir nuestro modelo justamente a aquellos elementos imprescindibles y a rechazar los que son innecesarios o enrevesados. Este trabajo ha permitido que nuestro modelo sea fácil de comprender y aplicar.

En resumen, se ha creado un modelo sencillo que permite la formación de VE dinámicas no sólo siguiendo los distintos enfoques presentados hasta la fecha sino cualquier otro (incluidas las distintas combinaciones de los anteriores). Y esta mayor adecuación a la realidad del proceso de selección se logra permitiendo que los clientes o empresas del VBE puedan bien publicar sus necesidades en el entorno de gestación o bien solicitar una lista de los posibles suministradores de un servicio.

Esta simple elección nos da la posibilidad de formar DVE jerárquicas, centralizadas, descentralizadas, etc.; permite crear VE dinámicas que se ajusten a la definición pura de DVE, VE dinámicas controladas por un miembro, DVE federadas, VE en las que se establezcan diferentes tipos de control en todos y cada uno de los procesos de negocio, o en unos pocos, etc. En resumidas cuentas, permite crear cualquier tipo de VE dinámica y con el control que se precise.

El resultado puede considerarse satisfactorio: un modelo del proceso de selección de DVE multienfoque, realista, innovador y sencillo, que amplía las posibilidades de los modelos hallados en la literatura y que reduce considerablemente sus carencias.

c) Realizar una justificación de la tecnología más adecuada para la construcción del sistema software, y seleccionar las especificaciones, metodologías y herramientas que mejor se ajusten a nuestros propósitos de entre aquellas que tengan una mayor aceptación.

La investigación realizada en torno al proceso de selección de socios en las empresas virtuales dinámicas permitió orientar esta etapa de la tesis hacia la utilización de una tecnología en auge: los sistemas multiagente. De hecho, las redes colaborativas, y en particular las empresas virtuales dinámicas, han demostrado ser uno de los dominios donde la tecnología de agentes proporciona una forma natural para resolver problemas que son inherentemente distribuidos.

En un MAS un conjunto de agentes heterogéneos se integran optimizando programas con objetivos y procesos diferentes. Dichos agentes son capaces de percibir los cambios en el entorno y adaptarse produciendo cambios en su comunidad para alcanzar los objetivos establecidos. Su autonomía les permite realizar tareas encaminadas a alcanzar objetivos locales, mientras que la

cooperación entre los agentes posibilita la coordinación del sistema para alcanzar los objetivos globales logrando, a su vez, la flexibilidad necesaria para adaptarse a los cambios.

En los últimos años se han desarrollado plataformas que facilitan el diseño e implementación de sistemas multiagente, acercando esta metodología a un público más amplio. En este sentido, JADE constituye una de las mejores opciones para la construcción de estos sistemas, pues simplifica de manera considerable el desarrollo de los agentes. JADE implementa el estándar FIPA en muchos aspectos, desde las facilidades que debe proporcionar una plataforma FIPA (AMS, DF, registro,...) hasta la comunicación mediante mensajes ACL, por lo que un agente JADE es capaz de interactuar y comunicarse con cualquier otro agente que soporte FIPA.

También ofrece una de las mejores y más completas interfaces de comunicación entre agentes e implementa, además, la mayoría de protocolos de interacción definidos por el estándar FIPA. Otra gran baza a favor de JADE es la cantidad de herramientas que ofrece para la ejecución y depuración del sistema multiagente, y que nos han sido especialmente útiles a lo largo del desarrollo de nuestra aplicación. A parte de las herramientas propias de la especificación FIPA (AMS, DF,...) ofrece otras como el RMA, el Introspector, el Dummy Agent o el Sniffer que simplifican el control y depuración de la plataforma y sus agentes.

En el lado negativo, JADE carece de mecanismos para dotar a los agentes de “inteligencia”, capacidad de planificación o razonamiento. Los agentes JADE no incorporan ninguna actitud mental más allá de sus comportamientos, e incluso éstos no son más que tareas, trozos de código Java a ejecutar paso a paso para lograr un resultado. Esta circunstancia es especialmente negativa sobre todo cuando pretendemos que el MAS se comporte como lo haría un decisor humano. Para resolver este problema nos hemos aprovechado de que JADE tenga una base Java, lo que le permite interactuar de un modo relativamente fácil con implementaciones Java de sistemas expertos.

Un ES es un sistema informático capaz de emular las prestaciones de un experto humano en un área concreta de conocimiento. Esta capacidad explica nuestra inclinación hacia su uso, ya que la resolución de los problemas sin tener que recurrir a enrevesados algoritmos permite representar mejor el proceso de selección que se da en las empresas. Posee, además, algunas características especialmente útiles.

Así, mejora la productividad de la aplicación al resolver los problemas más rápidamente, lo que se traduce en un ahorro considerable de tiempo y recursos. Por otro lado, los conocimientos se guardan y se difunden, de forma que no se pierden aunque desaparezca la fuente (el especialista). También permite obtener soluciones más fiables (gracias al tratamiento automático de los datos) y más contrastadas (debido a que se suele incorporar el conocimiento de varios expertos). El principal escollo que hemos encontrado en la construcción de este sistema se halla en que el conocimiento humano es difícil de extraer (las reglas no siempre son obvias), e incluso su representación resulta bastante problemática.

En cuanto a la herramienta empleada, Jess ha demostrado ser un motor de reglas muy rápido. Es pequeño, ligero, y su lenguaje de *scripting* nos ha permitido un fácil acceso a todas las API de Java. Gracias a ello, hemos podido construir agentes con capacidad para obtener resultados “inteligentes”. El principal inconveniente reside en que, como sistema basado en reglas declarativas, no se realiza ningún tipo de razonamiento humano, simplemente se hacen emparejamientos de reglas preestablecidas. Además, no incluye ningún mecanismo de aprendizaje automático.

Otro elemento interesante e imprescindible en la construcción de nuestro sistema multiagente son las ontologías. Las ontologías son un elemento fundamental en la comunicación entre agentes ya que permiten retratar el conocimiento común del dominio considerado: si los agentes que intervienen en la comunicación utilizan contextos distintos (es decir, ontologías distintas) o términos ambiguos, la comunicación puede ser un desastre. Pero, al mismo tiempo, las ontologías pueden constituir una declaración de principios sobre cómo deben ser los procesos de negocio: en este sentido, no sólo permiten describir el mundo objeto de estudio sino también cambiarlo.

Las ontologías definidas en esta tesis¹²⁴ constituyen un elemento clave en el éxito de la herramienta desarrollada. Como cualquier ontología poseen la funcionalidad básica de descripción del conocimiento, pero también permiten, sobre todo en el caso de la ontología del dominio de aplicación, que el proceso real de negocio y la documentación desarrollada se ajusten al modelo creado.

124 Tanto las empleadas para la gestión de la plataforma y la comunicación con los diferentes agentes de servicio como aquella definida para describir y representar nuestro dominio de aplicación.

En una aplicación real de la plataforma multiagente esta particularidad exigiría contar con expertos de las diferentes empresas (y sus procesos de negocio) y expertos en ontologías capaces de fusionar sus esfuerzos para mejorar y ampliar las ontologías existentes, adaptándolas tanto a los diferentes tipos de empresas como a la idiosincrasia de la empresa virtual dinámica y los entornos de gestación. Lograr este compromiso constituye, por tanto, uno de los retos más importantes a superar para lograr el éxito de la VE dinámica.

d) Análisis, diseño y construcción de una plataforma software que ofrezca una solución innovadora y eficiente al problema de la selección de socios en las VE dinámicas.

Para comprobar que el modelo responde como se espera y que permite representar correctamente lo que ocurre durante el proceso de selección de socios, hemos desarrollado la aplicación informática. En su elaboración hemos intentado seguir las fases características del ciclo de vida de toda aplicación software: planificación, análisis, diseño, construcción y, finalmente, validación.

La elección de JADE como plataforma de desarrollo multiagente nos ha condicionado para la utilización de las pautas establecidas en la metodología para JADE (Nikraz *et al.*, 2006). Gracias a los 6 pasos que define esta metodología para la fase de análisis, hemos logrado identificar los requisitos del dominio de aplicación y clarificar el problema con un nivel de detalle suficiente. El resultado de esta fase de análisis han sido los diagramas de casos de uso, el diagrama de agentes, la tabla de responsabilidades y el diagrama de despliegue de agentes, que convenientemente refinados han constituido la base para la fase del diseño.

Precisamente, la fase de diseño nos ha servido para desarrollar la solución con el suficiente nivel de detalle como para tener una transición relativamente directa a la construcción de la plataforma, es decir, a su codificación real. Para ello, hemos especificado diferentes aspectos como son las interacciones y los protocolos de interacción, las plantillas de los mensajes, la descripción de los servicios que deben ser registrados/buscados en el agente DF, las interacciones agente-recurso y agente-usuario, los comportamientos internos de los agentes, la definición de las ontologías y el lenguaje de contenido.

El resultado ha sido una herramienta capaz de dar respuesta a las necesidades que plantea la formación de DVE dentro del contexto de los VBE. Una aplicación en la que los agentes (en representación de los miembros del VBE y los clientes) interactúan para formar DVE que respondan a los requisitos de los clientes de manera ágil. Además, dicha formación está sustentada no en complejas funciones matemáticas sino en un razonamiento similar al que realizan habitualmente los directivos de las empresas.

Entre las principales características de esta plataforma multiagente *distribuida* podemos definir las siguientes:

- *Conjunción* de dos *tecnologías* software: los sistemas multiagente y los sistemas expertos. De hecho, este trabajo es pionero en su uso conjunto, y el primero que combina JADE y Jess para dar solución al problema de la selección de socios en las DVE. Se trata de una solución realmente *innovadora* desde el punto de vista *tecnológico*.
- Atendiendo al modelo especificado, la aplicación *permite definir el control sobre los procesos de negocio remotos*, bien seleccionando proveedores a partir de la lista que suministra el agente administrador, o bien publicando sus necesidades en la plataforma. Por tanto, también es *innovadora* desde el punto de vista *conceptual*.
- El sistema es *escalable*, permitiendo fácilmente la inclusión y eliminación de agentes en el sistema.
- Los agentes presentan un *comportamiento “inteligente”* gracias a la integración del sistema experto Jess. Los agentes empresa y cliente cuentan con una memoria de trabajo y un conjunto de reglas que definen su conducta cada vez que se activa el motor de inferencia.
- La selección de socios se realiza considerando tanto *factores cuantitativos como cualitativos*, tanto técnicos como subjetivos, etc.
- *El comportamiento del sistema construido puede ser modificado fácilmente*, ya que en cualquier momento se pueden cambiar tanto las reglas como los hechos que lo definen.
- La *comunicación* entre los agentes es *asíncrona*, gracias a los mecanismos de comunicación soportados por la plataforma FIPA.
- El contenido de los mensajes se describe a través de *ontologías abiertas y flexibles*, que hacen posible la comunicación.
- Los *agentes* son *autónomos* gracias, sobre todo, a la comunicación asíncrona y al uso de distintos protocolos FIPA.
- El uso de tecnologías abiertas, interoperables, estándar como FIPA, XML o Java aumenta la *transparencia* del sistema.

- *La incorporación de nuevos procesos de negocio es trivial* debido, en gran parte, a la *modularidad* con la que se ha diseñado y construido la aplicación.
- La *interfaz de usuario* es *simple e intuitiva* lo que facilita su utilización.
- El usuario tiene un *fácil acceso a la información generada* durante el transcurso de la actividad de su agente.
- El administrador de la plataforma tiene la *posibilidad de acceder a los agentes de gestión de Jade* en cualquier momento para las operaciones de gestión y monitorización que precise.

En definitiva, se ha diseñado y desarrollado una aplicación *flexible y dinámica* que refleja claramente el dinamismo del concepto de DVE y la agilidad que deben poseer este tipo de modelos organizativos. Estas características no se hubiesen logrado sin un modelo adecuado.

e) *Planteamiento y construcción de diferentes escenarios dentro del dominio de aplicación, con el propósito de verificar la adecuación del modelo propuesto y de la tecnología empleada.*

Se han establecido una serie de escenarios ficticios donde empresas Castillanoleonesas han actuado como proveedoras de los BP necesarios en la instalación de colectores fotovoltaicos. Partiendo de un modelo perfectamente competitivo en el que todos los proveedores de un mismo BP tienen idénticas características y comportamiento, se han ido introduciendo variaciones asociadas tanto a las características de los servicios (tanto prestados como demandados), condiciones externas, criterios de selección y contestación, tipo de control, etc., que han dado como resultado la formación de diferentes tipos de DVE. El estudio de estos escenarios nos ha permitido verificar tanto la adecuación del modelo inicial como lo acertado de combinar la tecnología multiagente y los sistemas expertos en la solución.

Por último, y antes de pasar a analizar las posibles líneas de actuación que deja abiertas nuestro trabajo, quisiera resaltar la utilización casi total de software libre en su elaboración (a excepción de Jess, que precisa licencia para uso académico y de investigación, y Windows XP, que junto a Ubuntu han sido los dos sistemas operativos empleados). Por tanto, hemos sido capaces de acercarnos bastante al principio de actuación que adoptamos al definir los objetivos de la presente tesis.

LÍNEAS DE TRABAJO FUTURO

Aunque el trabajo presentado en esta memoria intenta proporcionar una solución coherente, eficiente e innovadora al problema de la selección de socios en las DVE, somos conscientes de que existen ciertos aspectos sujetos a posibles mejoras y nuevas investigaciones. Surgen, además, atractivas posibilidades de investigación futura, no sólo en el ámbito empresarial sino en el campo de las ciencias sociales.

Pero antes de pasar a comentar estas posibles líneas de actuación, conviene destacar que no se han considerado mejoras del modelo de partida. A fin de cuentas, creemos haber conseguido el objetivo principal establecido al comienzo de la memoria, ofreciendo un modelo operativo, realista y sencillo del proceso de selección. Sin embargo, es posible que el trabajo posterior a esta tesis nos dirija hacia mejoras, e incluso ampliaciones, en su diseño.

Mejoras en la plataforma

MEJORA DE LOS AGENTES DE SERVICIO EXISTENTES...

En la plataforma hemos considerado dos agentes “auxiliares” cuya labor consiste, fundamentalmente, en suministrar información al agente empresa: nos estamos refiriendo al *agente de servicio divisas* y al *agente de servicio transporte*. Gracias a estos agentes se ha logrado mejorar el servicio de la plataforma, dado que suministran una información valiosa para aquellos agentes empresa que soliciten sus servicios. Con esta información los agentes son capaces de confeccionar ofertas realistas, actualizadas y adaptadas a las exigencias de las solicitudes.

Estos agentes se encargan, respectivamente, de gestionar los cambios de divisas y de proporcionar información asociada a la logística del transporte como son la distancia y el coste (esta última en función de un gran número de variables entre las que se incluyen el punto de origen y destino, el medio de locomoción, el tipo de combustible, etc.). En ambos casos, y con el propósito de simplificar la construcción de la plataforma, se ha considerado que la información a la que acceden se encuentra disponible en el servidor del administrador, quien se encarga de actualizarla a través de otras herramientas externas a la plataforma.

Sin embargo, esta solución no es operativa ni realista, por lo que no es satisfactoria. En efecto, para lograr un comportamiento óptimo de estos agentes son necesarias herramientas que permitan una actualización automática de los datos. Así, por ejemplo, para el caso del *agente de servicio divisas* necesitaríamos un comportamiento similar al que ofrecen gran multitud de páginas web en las que se puede obtener el valor monetario de una determinada cantidad expresada en una divisa diferente a la inicialmente considerada.

Por lo que se refiere al *agente de servicio transporte* el razonamiento es muy similar. En el mercado existen portales cartográficos, como Vía Michelin (Vía Michelin web, 2007) o Mappy (Mappy web, 2007), capaces de calcular la distancia entre dos puntos o el coste de transporte en función de los puntos de origen y destino, el tipo de travesía (la más rápida, la más económica, la más corta, etc.), el medio de locomoción, el coste del combustible, las aduanas, los peajes, el tamaño de los vehículos, etc.

Sería conveniente, en ambos casos, que nuestros agentes de servicio fuesen capaces de interactuar (comunicarse) con algún portal de este tipo donde la información disponible (tipo de cambio, coste de la gasolina, rutas actuales, peajes, etc.) es actualizada periódicamente. De esta forma, la gestión de estos datos sería una cuestión totalmente ajena al administrador, y las empresas del VBE sabrían en todo momento que la información empleada se encontraría perfectamente actualizada. Obviamente, el administrador debe garantizar la confiabilidad de estos portales y la existencia de agentes con los que nuestros propios agentes de servicio puedan interactuar.

... E INCORPORACIÓN DE ALGUNOS NUEVOS

Pero también podemos incorporar nuevos agentes de servicio que den respuesta a otras necesidades de la plataforma de selección y, como no podía ser de otro modo, de los entornos de gestión. En este sentido resulta significativo el trabajo realizado por Cardoso y sus colaboradores (Cardoso *et al.*, 2005), donde se introduce el concepto de institución electrónica. Las instituciones electrónicas son marcos completos que puedan ayudar eficazmente en la labor de cooperación de las organizaciones virtuales. En concreto, Cardoso centra su esfuerzo en crear servicios basados en las ontologías y la contratación electrónica en el contexto de las instituciones electrónicas.

Por un lado, los servicios de contratación vía web proporcionan la especificación automática de los acuerdos de negocio mediante su formalización a través de contratos electrónicos (*e-contracting*), además de su cumplimiento. Por otro, los servicios basados en ontologías permiten la interoperabilidad entre los agentes que representan a organizaciones que utilizan ontologías diferentes. Los servicios de ontología ofrecen consejos útiles sobre la forma de negociar temas específicos, lo que se traduce en conversaciones apropiadas y en que se materialicen los acuerdos.

Fruto de la introducción de este concepto, ha surgido la posibilidad de colaborar con este autor en la Universidad de Oporto, lo que se podría traducir en la incorporación a la plataforma de agentes intermediarios entre estos servicios institucionales y los agentes representantes de los usuarios. Además, esta colaboración podría ser incluso más beneficiosa dado que el profesor Cardoso también está trabajando en aspectos relacionados con los sistemas expertos usando la combinación JADE-Jess, lo que es otro punto en común.

INTEGRACIÓN DE LA INFORMACIÓN DE LA EMPRESA

Como ya sabemos, los servicios que ofrece cualquier empresa de la plataforma se incorporan a la memoria de trabajo en forma de hechos que se definen en el documento Jess correspondiente (*...EmpresaRegistro.clp*). En la aplicación actual es la propia empresa la que se encarga de establecer tales hechos de forma manual, con el consiguiente consumo de tiempo y recursos que eso conlleva, por no hablar de los errores que puede dar lugar si se introducen los datos de forma incorrecta.

No obstante la solución a este problema es conceptualmente sencilla y tan sólo requiere interconectar nuestra aplicación con los recursos de la empresa. Para ello, tan sólo sería necesario incorporar un agente auxiliar capaz de acceder a la información de los productos-servicios que oferta la empresa, bien a través de bases de datos o mediante su interacción con el software que la gestiona. Este agente transformaría la información necesaria en hechos que incorporaría a la memoria de trabajo cada vez que el agente empresa se lo solicitase como consecuencia de la recepción de una oferta.

Sólo de esta forma estaremos en disposición de ofrecer una oferta más acorde a la realidad y plenamente actualizada.

MEJORA DEL COMPORTAMIENTO DE LOS AGENTES

Somos conscientes de que el comportamiento de nuestros agentes (a partir de las reglas y la memoria de trabajo) puede resultar bastante básico si se compara con algunos de los sistemas desarrollados en la actualidad para otros propósitos. Sin embargo, el conjunto de reglas y hechos definidos para los distintos agentes sirve claramente a nuestros propósitos iniciales, ya que nos proporciona el comportamiento deseado.

Sin embargo, lograr una aplicación más ajustada a la realidad de las empresas, y en concreto a la de las VE dinámicas, exige que los hechos y la memoria de trabajo sea modificada. Además, es necesario disponer de mecanismos más eficientes e intuitivos que permitan a los usuarios modificar fácilmente tanto los hechos como las reglas sin que esto les obligue a un conocimiento profundo de los entresijos de Jess.

Por tanto, otra posible opción de futuro pasa por migrar el actual formato de los hechos y reglas Jess a XML. Esto es posible gracias a que Jess ya soporta un lenguaje de reglas basado en XML, *JessML*. Este lenguaje ha sido diseñado con una estructura similar al lenguaje Jess, pero dotado de ciertas especificaciones que hacen posible el tratamiento automatizado de la información. No obstante, JessML presenta ciertos aspectos que precisan de un mayor desarrollo, aunque es muy probable que dichas mejoras se vayan incorporando en los próximos lanzamientos de Jess.

MEJORA DE LA INTERFAZ DE CONTROL

Otro elemento de la aplicación que debe mejorar es la representación gráfica de las configuraciones de control. Dadas las características del proceso de negocio considerado para la realización de los experimentos, la actual representación es suficiente, pero lo cierto es que cuando se valoren BP más complejos la actual pestaña de control resultará poco funcional.

Deberemos, entonces, suministrar un visor de la configuración capaz de, por ejemplo: realizar *zoom*, delimitar claramente los distintos niveles jerárquicos y las conexiones entre los BP, permitir su edición para distribuir la estructura de forma más acorde a las necesidades del usuario, etc., y en el que los iconos estarán vinculados a las pestañas de los BP para permitir un fácil y rápido acceso a la información correspondiente.

LOS MÓDULOS NETBEANS

Un módulo *NetBeans* es básicamente un archivo .jar que contiene clases Java escritas para interactuar con la API de NetBeans. Los módulos NetBeans se pueden utilizar para ampliar la funcionalidad del IDE con nuevas características o bien para construir aplicaciones enriquecidas (*rich-client application*). En este segundo caso, se puede usar el núcleo de la IDE como una plataforma sobre la que desarrollar aplicaciones de escritorio independientes.

Al basar la aplicación en la plataforma NetBeans, se puede ahorrar mucho tiempo de desarrollo, ya que se pueden reutilizar características existentes de la plataforma tales como menús, barras de herramientas y los sistemas de ventanas. Sin embargo, hasta la fecha han surgido ciertos conflictos entre esta tecnología y las librerías JADE que no hemos podido subsanar. Sin embargo, su potencial es muy interesante y creemos que es la forma más adecuada para construir futuras aplicaciones.

Nuevas tecnologías

JADE LEAP Y LOS DISPOSITIVOS HANDHELD

Como consecuencia de la introducción de las redes inalámbricas (GPRS, UMTS, WLAN) y del continuo crecimiento de dispositivos de pequeño tamaño como las PDA y teléfonos móviles, se está produciendo una integración progresiva de los entornos inalámbricos y de cable. En este escenario la necesidad de desplegar aplicaciones que estén distribuidas parcialmente en la red fija y parcialmente en estos dispositivos “de mano” (*handheld*) es cada vez más importante.

Esta urgencia es especialmente visible en la formación de las DVE. Como seguramente se recordará, las oportunidades de colaboración que activan la formación de este tipo de consorcios pueden surgir en cualquier momento, por lo que las empresas no pueden permitirse el lujo de estar ligadas a una red fija. Esta circunstancia se ve agravada en aquellos casos en los que el socio potencial es una persona física, un autónomo. Necesitamos, por tanto, ofrecer una herramienta más completa que tenga en cuenta las posibilidades asociadas a estos “pequeños” dispositivos.

En 1999, Sun Microsystems anunció la aparición de *Java 2 Micro Edition* (J2ME) con el propósito de permitir que aplicaciones Java se ejecutasen en dispositivos con potencia de procesamiento limitada, como teléfonos móviles, PDA, y otros. Supone, en definitiva, una solución que responde a la amplia difusión que están teniendo estos dispositivos en los últimos años y a la demanda de usuarios y proveedores de servicios de recibir/ofrecer nuevas aplicaciones para aumentar las funcionalidades que aportan estos pequeños dispositivos.

Considerando que Java es el lenguaje de programación en el que más aplicaciones de agentes móviles se han desarrollado, las posibilidades de desarrollar plataformas de agentes en estos dispositivos se presentan alcanzables. De hecho, existen varias propuestas que pretenden involucrar a dispositivos limitados J2ME en plataformas de agentes móviles; de entre todas ellas, destaca significativamente el proyecto IST LEAP (LEAP web, 2006).

El proyecto LEAP (*Lightweight Extensible Agent Platform*) engloba un consorcio de compañías entre las que se encuentran Motorola, British Telecom y Siemens. El objetivo del proyecto es el desarrollo de una plataforma de agentes en Java conforme al estándar FIPA que pueda operar tanto en dispositivos limitados (teléfonos móviles, PDA, etc.) con J2ME, como en PC con J2SE. Para ello, han diseñado una arquitectura modular, con una parte obligatoria, común a todos los tipos de dispositivos, y otra opcional; mediante un instalador se puede componer la plataforma según las limitaciones del dispositivo en el que se instale.

La plataforma JADE incorpora un *add-on* basado en este proyecto, el *LEAP add-on*, que permite a los agentes FIPA ejecutarse en dispositivos de capacidad limitada. Cuando se combina con JADE, este *add-on* reemplaza algunas de las partes del *Kernel* de JADE formando un entorno de ejecución JADE modificado que puede ser desplegado en una gran variedad de dispositivos que van desde los servidores hasta los teléfonos móviles basados en Java.

Haciendo uso de este *add-on* seremos capaces de ofrecer una solución más realista al problema de la selección de socios, y su gestión. Podremos, por un lado, ofrecer una herramienta independiente de la ubicación física y geográfica de los usuarios y, al mismo tiempo, ampliaremos el abanico de posibles socios o colaboradores.

RAZONAMIENTO BASADO EN CASOS

El enfoque tradicional de la IA se basa en construir un sistema de la nada, tratando de resolver los problemas mediante la construcción de soluciones basadas en principios generales (o reglas) y en un conocimiento profundo de los expertos del tema. Sin embargo, los peligros de la adquisición de los conocimientos de un experto humano han sido bien documentados y han dado lugar a la expresión "...la adquisición de conocimiento es el cuello de botella..." acuñada por Feigenbaum. En contraste, el razonamiento basado en casos (CBR) recuerda casos "pasados" similares al problema actual y transforma la solución correspondiente para resolver el nuevo problema.

Por tanto, el uso de este tipo de razonamiento puede ser una solución muy interesante para el futuro próximo de la plataforma. Podemos sustituir el módulo de decisión basado en Jess (reglas) por un módulo basado en el uso de sistemas expertos de "segunda generación" o, lo que es incluso más interesante, combinar ambos tipos de razonamiento (es decir, reglas y casos) en un mismo módulo de decisión.

Mientras que las reglas contienen el conocimiento que ha sido codificado a lo largo de los años por expertos, los casos contienen el conocimiento acerca de un tema de una manera menos procesada. Ninguna de las dos tecnologías excluye a la otra, ya que en innumerables ocasiones las reglas codificadas no se pueden representar correctamente con un conjunto de casos y la idiosincrasia del conocimiento de los casos muchas veces no puede ser capturada por reglas conocidas.

Pero el uso conjunto de reglas y casos no sólo permite que la arquitectura tome ventaja de un mayor dominio del tema sino que, además, favorece la introducción de innovaciones en la tecnología basada en casos que pueden conducir a resultados más prometedores que los generados por cualquiera de las tecnologías de forma aislada:

- En primer lugar, las reglas pueden utilizarse para indexar los casos. En este sentido, podemos unir directamente los casos con las reglas utilizando los antecedentes de estas últimas, lo que proporciona las claves apropiadas para la recuperación de los casos. Además, con este comportamiento evitamos tener que analizar el dominio para identificar un vocabulario que ya está disponible de manera implícita en las reglas.

- Otra forma de utilizar las reglas es aplicarlas al problema para obtener una respuesta aproximada y así diseñar analogías de los casos que permitan cubrir las excepciones a las reglas.

Obviamente, el obstáculo clave para la utilización del razonamiento basado en casos reside en la necesidad de una gran “base de casos” de la que extraer los casos antiguos que servirán de base en la construcción de la nueva solución. Sin embargo, por diversas razones esa base de datos puede ser bastante difícil de encontrar. Además, también debe existir una buena comprensión del dominio por lo que no conviene emplearla, por ejemplo, cuando se abordan nuevas áreas en las que hay información incompleta o existe un conocimiento incorrecto o controvertido.

Por tanto, un primer paso antes de hacer uso de esta lógica de razonamiento consistirá en proporcionar la correspondiente base de casos de partida y en delimitar claramente el dominio y los conceptos asociados.

JADEx

Otra posible extensión en relación al módulo de razonamiento de los agentes consiste en la utilización de Jadex. Se trata de un motor de razonamiento para agentes inteligentes basado en el modelo BDI. Por lo tanto, y al igual que ocurre con otros agentes BDI, los agentes Jadex están formados principalmente por un conjunto de creencias, planes (entendidos como recetas), objetivos y eventos.

Posee una API que facilita el acceso a los conceptos de Jadex cuando se programan los planes (construidos en Jadex como una clase Java predefinida que proporciona métodos útiles para gestionar las creencias). Para definir un agente, además de los planes hay que crear un archivo XML que contenga las creencias, deseos y planes iniciales del agente. Este marco brinda también ciertas funcionalidades que pueden facilitar la definición de las capacidades del agente.

Jadex es, además, adaptable a distintas plataformas (Jadex web, 2008). Aunque originalmente fue diseñado para ser usado con JADE, el deseo de sus autores por limitar al máximo sus dependencias con respecto a la plataforma que lo ejecute ha permitido que en la actualidad Jadex pueda utilizarse de manera autónoma con otras plataformas y sistemas (esta separación de

sistema de razonamiento y plataforma lo hace muy atractivo). Como motor de razonamiento Jadex permite realizar agentes BDI utilizando XML y Java. Mediante el adaptador¹²⁵ pertinente, el agente tiene acceso a la plataforma, pudiendo registrarse en su AMS, ofrecer servicios en el DF o utilizar los servicios de mensajería que ésta le proporcione.

LOS SERVICIOS WEB

Un servicio web es básicamente una interfaz que describe una colección de operaciones a las que se puede acceder a través de una red (o de Internet) mediante mensajes XML estandarizados. Esta interfaz oculta los detalles de implementación del servicio, lo que permite que pueda ser utilizado con independencia tanto de la plataforma hardware o software como del lenguaje de programación que se emplee.

A pesar de su reciente popularidad, existen multitud de limitaciones para aplicar eficientemente este paradigma a la selección de socios en las DVE. Sin embargo, también es cierto que han empezado a aparecer los primeros trabajos en los que se intenta integrar ambas tecnologías buscando dar una mejor respuesta a los problemas que rodean a las organizaciones virtuales y, en particular, a las DVE. En todas ellas se considera que los servicios web representan un paradigma de computación prometedor para lograr la integración de aplicaciones a través de Internet y, por lo tanto, para facilitar la formación de DVE entre diferentes plataformas software y hardware.

Estos trabajos constituyen el germen de una prometedora investigación y, por tanto, un posible camino por el que transcurra nuestra investigación futura. No obstante, hay que tener presente que todavía son múltiples y muy variadas las limitaciones que plantea la tecnología actual de los servicios web en relación con los procesos que se han de llevar a cabo en la formación y operación de las VE dinámicas, lo que limita considerablemente su aplicabilidad en este dominio.

125 Con Jadex se proporciona el adaptador *standalone* que permite su ejecución de manera aislada y autónoma, y también es posible descargar el adaptador para la plataforma JADE. Existe, por último, un tercer adaptador a otra plataforma en fase experimental. Por tanto, el número de plataformas sobre las que actualmente puede correr Jadex aún es muy pequeño para un motor que se anuncia como independiente de la plataforma.

Hacia una mayor aplicación práctica

INTEGRACIÓN DE PROYECTOS

Una posible vía de investigación futura, que a nuestro juicio puede resultar muy interesante y prometedora, consiste en unir el trabajo de investigación aquí descrito con el realizado por el Dr. José Alberto Araúzo Araúzo en su tesis doctoral “*Control distribuido de sistemas de fabricación flexibles: un enfoque basado en agentes*” (Araúzo, 2007), defendido recientemente en el departamento de Organización de Empresas y Comercialización e Investigación de Mercados de la Universidad de Valladolid.

Dicha tesis permite mostrar la adecuación de los sistemas multiagente al control de la fabricación en situaciones donde las técnicas utilizadas hasta el momento no han solucionado el problema satisfactoriamente. Para ello, el autor desarrolla un prototipo de sistema de control de planta basado en agentes JADE para sistemas de fabricación con configuraciones *job-shop* flexibles.

La integración de este sistema multiagente con el aquí presentado mejoraría significativamente la funcionalidad y aplicabilidad de nuestra plataforma: aquellas empresas que hiciesen uso de sistemas de fabricación flexibles en sus procesos, en particular con un control de tipo *job-shop*, serían capaces de proporcionar una respuesta mucho más realista a las necesidades de los clientes. A fin de cuentas, sólo con un mayor conocimiento de los recursos y disponibilidades de sus procesos de fabricación se pueden ofrecer datos de costes y tiempos de elaboración más acordes con el funcionamiento real.

La integración eficiente de ambas herramientas se verá favorecida por el hecho de que ambas utilizan JADE como plataforma de desarrollo multiagente¹²⁶. En cualquier caso, el objetivo final que se persigue con este tipo de integraciones es el de suministrar una herramienta que tenga en cuenta todos y cada uno de los flujos de información existentes¹²⁷ en las empresas del entorno de gestación. Sólo así podremos ofrecer una solución plenamente eficiente y real al proceso de selección de socios en las DVE.

126 Ésta era una de las razones que justificaban la elección de la plataforma JADE.

127 De acuerdo con el modelo CIM (*Computer Integrated Manufacturing*) de la NBS (*National Bureau of Standards*).

CONCRETAR LOS DOMINIOS DE APLICACIÓN

La instalación de módulos fotovoltaicos constituye un campo de aplicación e investigación muy interesante, a la vez que amplio y complejo. En nuestro trabajo tan sólo hemos intentado recrear parte de este dominio, por lo que tanto la descomposición final de los BP como su descripción constituyen una mera simplificación de la realidad. Esta decisión se tomó en su momento con la intención de facilitar el trabajo de construcción y validación de la plataforma, sabedores que las conclusiones finales no iban a depender ni de la dimensión ni de la naturaleza del dominio de aplicación.

Sin embargo, cuando se desea dotar de mayor carga práctica a la solución este hecho constituye una laguna en nuestro trabajo que deberemos subsanar. Obviamente, se necesita contar con información más completa y real de este dominio (o de cualquier otro), para lo que es preciso superar dos grandes dificultades:

- Por un lado, es necesario que un gran número de empresas se pongan de acuerdo para compartir información, tecnología, reglas de actuación, etc., en definitiva para formar parte de un entorno de gestación. Este requisito es, a día de hoy, una auténtica utopía para la mayor parte de las empresas actuales.
- Además, las empresas del mundo real, sobre todo las de pequeño y mediano tamaño, normalmente no disponen de los recursos ni de la voluntad suficiente para colaborar en este tipo de proyectos.

Estos y otros obstáculos ponen de manifiesto la dificultad que conlleva dar este paso dentro del ámbito empresarial. No obstante, la aplicación de nuestro trabajo al ámbito de la prevención y solución del *bullying* (que analizaremos en breve) puede constituir un magnífico punto de partida en esta tarea.

MEJORAR LA SEGURIDAD DE LA PLATAFORMA

Un requisito clave en el éxito de las empresas virtuales dinámicas, y que contribuiría a aumentar la participación de las empresas en nuestra investigación, es la seguridad. Sin embargo, este tema apenas ha sido considerado en la construcción de la plataforma (a excepción del agente de acceso) dada la escasa relevancia que posee en relación a los objetivos establecidos en un primer momento.

No obstante, su importancia es crucial cuando se considera la aplicación real de esta herramienta a gran escala; es, por tanto, un aspecto que deberemos considerar en breve. Por ese motivo, y como primer paso, será conveniente hacer uso del *add-on* Jade-s de la plataforma JADE. Esta herramienta proporciona el soporte de seguridad para los sistemas multiagente JADE, favoreciendo una administración segura de la plataforma.

PROTOCOLO CONTRACT-NET MULTIRONDA

En nuestra plataforma multiagente los acuerdos entre los distintos agentes (bien cliente-empresa o empresa-empresa) se logran mediante una única negociación en la que se emplea el protocolo Contract-Net de FIPA. Es el propio módulo de decisión de los agentes el encargado de seleccionar las mejores ofertas a partir de los criterios y las reglas que los expertos o directivos humanos hayan establecido previamente.

Sin embargo, este comportamiento podría verse mejorado permitiendo que los agentes pudieran verse inmersos en varias rondas de negociación, de tal forma que en cada interacción se emplease el mismo protocolo Contract-Net. De este modo, los agentes con las mejores ofertas en cada ronda dispondrían de mayor información que podrían emplear para mejorar sus ofertas en las sucesivas interacciones. Cada ronda permitiría filtrar las ofertas, y en consecuencia los agentes proveedores, hasta llegar a una última ronda en la que se realizaría la selección final.

De este modo, las entidades solicitantes se verían beneficiadas ya que recibirían ofertas mucho más ajustadas y competitivas, tanto desde un punto de vista económico como temporal. No obstante, la gran beneficiada con esta manera de proceder no sería otra que la propia plataforma multiagente, ya que habríamos logrado que poseyera un comportamiento más realista y eficiente.

APLICACIÓN EN LA PREVENCIÓN DEL MALTRATO INFANTIL

Por último, ha surgido la posibilidad de aplicar nuestra plataforma multiagente a un problema real, aunque no dentro del ámbito empresarial, a raíz de la próxima celebración del IX Congreso Estatal de Infancia Maltratada que se desarrollará próximamente en Valladolid.

La finalidad de este congreso es potenciar un modelo de trabajo en red desde todos los ámbitos y organismos involucrados en la prevención, detección e intervención de situaciones de maltrato infantil. Con este planteamiento se intenta sensibilizar sobre la trascendencia que tiene el trabajo coordinado y la implicación de todos los profesionales afectados.

Las dificultades y necesidades que esta atención conjunta plantea a los profesionales exige nuevas alternativas y enfoques que potencien respuestas más eficaces y actuaciones más coherentes. Nuestra plataforma software abre una nueva vía de actuación que facilitará una mejora de la respuesta de los organismos públicos e instituciones al incorporar nuevos criterios de coordinación.

A través de la colaboración con REA¹²⁸, Asociación Castellano-leonesa para la Defensa de la Infancia y la Juventud, estamos trabajando en la personalización de nuestra herramienta multiagente para facilitar este trabajo en red necesario a la hora de detectar, prevenir y solucionar la problemática del maltrato entre iguales (*bullying*), aunque posteriormente se pretende que abarque a todos los tipos de maltrato infantil.

La plataforma DVEBreeder permitiría la formación de *organizaciones* virtuales dinámicas de una forma fácil e intuitiva según las necesidades del problema. De esta forma, los centros educativos, centros de salud, servicios sociales, servicios de menores, organismos judiciales, cuerpos de seguridad,... , (en definitiva, los miembros del entorno de gestación) dispondrán de una herramienta para comunicarse, intercambiar información relevante y coordinar su respuesta de forma eficaz.

Obviamente, la actividad de adecuación a este nuevo dominio constituirá una primera aplicación real de nuestra herramienta (y, en consecuencia, una forma de testear su verdadero potencial) pero, al mismo tiempo, supondrá una evolución de nuestro trabajo hacia la fase operativa de las organizaciones virtuales dinámicas.

128 Tanto REA como la Junta de Castilla y León son pioneras en programas de prevención del maltrato infantil. Los programas de la asociación han sido editados y utilizados por diferentes estamentos de la Junta.

BIBLIOGRAFÍA

REFERENCIAS BIBLIOGRÁFICAS

- (Aamodt y Plaza, 1994) Aamodt A.; Plaza E., (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approach. *AI Communications IOS Press*, vol 7, nº1, pp. 39-59.
- (Adams *et al.*, 2001) Adams W.M.; Wallas, R.M.; Sengupta, A., (2001). Collaborative Commerce: The Agile Virtual Enterprise Model. *Pushing the Digital Frontier: Insights into the Changing Lands of E-Business*. Penn State eBusiness Research Center, pp. 242-262.
- (Aerts *et al.*, 2002) Aerts, A.T.M; Szirbik, N.B.; Goossenaert, J.B.M., (2002). A Flexible, Agent-Based ICT Architecture for Virtual Enterprises. *Computers in Industry*, vol. 49, nº 3, pp. 311-327.

- (Afsarmanesh y Camarinha-Matos, 2005) Afsarmanesh, H.; Camarinha-Matos, L.M., (2005). A Framework for Management of Virtual Organization Breeding Environments. *Collaborative Networks and their Breeding Environments*, (PRO-VE'05), Springer, Valencia, Spain.
- (Afsarmanesh, 2005) Afsarmanesh, H. (ed.) (2005). *Key Components, Features, and Operating Principles of the Virtual Breeding Environment*. ECOLEAD Deliverables. En (VE-forum web, 2006).
- (Ambroszkiewicz *et al.*, 1998) Ambroszkiewicz, S.; Cetnarowicz, K.; Radko, B., (1998). Enterprise Formation Mechanisms based on Mobile Agents. *Proceedings of Workshop on Intelligent Agents in Information and Process Management, KI'98*, pp. 23-34.
- (Amor, 2002) Amor, D. (2002). *The e-Business (r)evolution*. Prentice Hall Books.
- (Appel y Behr, 2000) Appel, W.; Behr, R., (2000). Towards the Theory of Virtual Organizations: A Description of their Formation and Figure. *Electronic Journal of Organizational Virtualness*, Newsletter vol. 2, nº 2, pp 35-48.
- (Applegate *et al.*, 1996) Applegate, L. M.; Holsapple C. W.; Kalakota R.; Radermacher F. J.; Whinston A. B., (1996). Electronic Commerce: Building Blocks of New Business Opportunity. *Journal of Organizational Computing and Electronic Commerce*, vol. 6, nº 1, pp. 1-10.
- (Araújo, 2003) Araújo, J.A. (2003). *Sistemas multiagentes para el control de plantas de fabricación*. Trabajo para la obtención de la suficiencia investigadora. Universidad de Valladolid.
- (Arias, 2005) Arias, J., (2005). *Definición de un modelo para la verificación formal de procesos de negocio*. Tesis doctoral. Departamento de Ingeniería telemática. Universidad Carlos III de Madrid.
- (Babic y Plazibat, 1998) Babic, Z.; Plazibat, N., (1998). Ranking of Enterprises based on Multicriteria Analysis. *International Journal on Production Economics*, vol. 56-57, pp. 29-35.
- (Backhouse y Burns, 1999) Backhouse, C.J.; Burns, N.D., (1999). Agile Value Chains for Manufacturing - Implications for Performance Measures. *International Journal of Agile Management Systems*, vol. 1, nº 2, pp. 76-82.

- (Barata, 2005) Barata, J.A., (2005). FETISH-ETF: Federated European Tourism Information System Harmonization - Engineerink Task Force. *Virtual Organizations – Systems and Practices* (Camarinha-Matos, Luis and Afsarmanesh, Hamideh, eds.), Springer Science, pp. 301-302.
- (Bauer *et al.*, 2001) Bauer, B.; Müller, J. P.; Odell, J., (2001). Agent UML:A Formalism for Specifying Multiagent Interaction. *International Journal of Software Engineering and Knowledge Engineering*, vol. 11, n°. 3, pp. 91-103.
- (Bauer y Köszegi, 2003) Bauer, R.; Köszegi, S., (2003). Measuring the Degree of Virtualization. *Electronic Journal of Organizational Virtualness*, vol. 5, n° 2, pp. 26-46.
- (Beer *et al.*, 1990) Beer, M.; Eisenstat, R.A.; Spector, B., (1990). Why Change Programs don't Produce Change. *Harvard Business Review*, vol. 68, n° 6, pp. 158-166.
- (Bellifemine *et al.*, 1999) Bellifemine, F., Rimassa, G., Poggi, A., (1999). JADE: A FIPA Compliant Agent Framework. *Proceedings 4th international conference and exhibition on the Practical Applications of Intelligent Agents and Multi-Agent systems* (PAAM 99), London.
- (Bellifemine *et al.*, 2001) Bellifemine, F.; Poggi, A.; Rimassa, G., (2001). JADE: a FIPA2000 Compliant Agent Development Environment. *Proceedings of the fifth International Conference on Autonomous Agents*, ACM.
- (Bellifemine *et al.*, 2007) Bellifemine, F.; Caire, G.; Trucco, T.; Rimassa, G.; Mungenast, R., (2007). *Jade Administrator's Guide*. En (Jade web, 2008)
- (Bellifemine *et al.*, 2008) Bellifemine, F.; Caire, G.; Trucco, T.; Rimassa, G., (2008) . *Jade Programmer's Guide*. En (Jade web, 2008).
- (Bhattachryya y Rahman, 2002) Bhattachryya, S.K.; Rahman, Z., (2002). Virtual. Organisation:. A. Stratagem. *Singapore Management Review*, vol. 24, n° 2, pp. 29-45.
- (Bichler *et al.*, 1999) Bichler, M.; Kumar, M.; Jhingran, A., (1999). *Multi-attribute competitive bidding*. IBM T.J. Watson Research Report.

- (Billington, 1994) Billington, C., (1994). Strategic Supply Chain Management. *OR/MS Today*, vol. 21, nº 2, pp. 20-27.
- (Binstock, 2000) Binstock, A., (2000). *Virtual Enterprise Comes of Age*. Information Week.com
- (Bischofberger, 2004) Bischofberger, T. (2004). *IT-Infrastructure for Dynamic Virtual Enterprises*. Master Thesis. Department of Informatics of the University of Zurich.
- (Blanco y Pavón, 2000) Blanco, D.; Pavón, J., (2000). *Zeus, una herramienta para el desarrollo de sistemas multiagente*. Informe del Grupo de Agentes del Departamento de sistemas Informáticos y Programación, Universidad Complutense de Madrid.
- (Bloch y Pigneur, 1996) Bloch, M.; Pigneur, Y., (1996). On the Road of Electronic Commerce: a Business Value Framework, Gaining Competitive Advantage and Some Research Issue. *Tronic Commerce for Trade Efficiency and Effectiveness*, Bled, pp. 91-112.
- (Bolcer y Kaiser, 1999) Bolcer, G. A., Kaiser, G., (1999). SWAP: Leveraging the Web to Manage Workflow. *IEEE Internet Computing*, vol. 3, nº 1, pp. 85-88.
- (Borghoff *et al.*, 1997) Borghoff U.M., Bottoni, P., Mussio, P., Pareschi, R., (1997). Reflective Agents for Adaptive Workflows. *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM97)*, London, pp 405-420.
- (Botia, 2005) Botia, J.A., (2005). *Tutorial básico de JADE*. Escuela de Primavera de Agentes, patrocinado por Agentcities.es. Imaginática 2005, Universidad de Sevilla.
- (Boudreau *et al.*, 1998) Boudreau M.C.; LochDaniel K.; Straud R.D., (1998). Going Global: Using Information Technology to Advance the Competitiveness of the Virtual Transnational Organisation. *Academy of Management*, vol. 12, nº 4, pp. 120-28.
- (Bratman, 1987) Bratman, M.E., (1987). *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA.

- (Bremer *et al.*, 1999) Bremer, C.; Mundim, A.; Michilini, F.; Siqueira, J.; Ortega, L., (1999). A Brazilian Case of VE Coordination. *Proceedings of the IFIP TC5 WG5.3 / PRODNET Working Conference on Infrastructures for Virtual Enterprises: Networking Industrial Enterprises*, vol. 153, pp. 377-386.
- (Browne y Zhang, 1999) Browne J.; Zhang J., (1999), Extended and Virtual Enterprises - Similarities and Differences. *International Journal of Agile Management Systems*, vol. 1, n° 1, pp. 30-36.
- (Bussman y Müller, 1993) Bussman S.; Müller H.J., (1993). A Communication Architecture for Cooperating Agents. *Computers and Artificial Intelligence*, vol. 12, n° 1, pp. 37-53.
- (Bussmann *et al.*, 2004) Bussmann, S.; Jennings, N.R.; Wooldridge, M., (2004). *Multiagent Systems for Manufacturing Control*. Springer-Verlag.
- (Byrne *et al.*, 1993) Byrne, J.A.; Brandt, R.; Port, O., (1993). The Virtual Corporation: The Company of the Future Will Be the Ultimate in Adaptability (Cover Story). *International Business Week*, vol. 8, pp. 36-40.
- (Byrne, 1999) Byrne, C., (1999). FACTS and Fiction: Agents in the Electronic Travel Market. *Communicate*, vol 4, n° 2.
- (Caire *et al.*, 2002) Caire, G.; Leal, F.; Chainho, P.; Evans, R.; Garijo, F.; Gomez-Sanz, J. J.; Pavon, J.; Kerney, P.; Stark, J.; Massonet, P., (2002). *MESSAGE - Methodology for Engineering Systems of Software Agents*. Eurescom Technical Information P907.
- (Caire *et al.*, 2002b) Caire, G.; Leal, F.; Chainho, P.; Evans, R.; Garijo, F.; Gomez-Sanz, J. J.; Pavon, J.; Kerney, P.; Stark, J.; Massonet, P., (2002). Agent Oriented Analysis using MESSAGE/UML. *Lecture Notes in Computer Science*, M. Wooldridge, G. Weiss, and P. Ciancarini (Eds.). Springer Verlag, vol. 2222, pp. 119-135.
- (Caire y Cabanillas, 2004) Caire, G.; Cabanillas, D., (2004). *Jade Tutorial - Application-Defined Content Languages and Ontologies*. En (Jade web, 2008).
- (Caire, 2003) Caire, G., (2003). *Jade Tutorial - Jade Programming for Beginners*. En (Jade web, 2008).

- (Camarinha-Matos *et al.*, 1998) Camarinha-Matos, M.; Afsarmanesh, H.; Garita, C.; Lima, C., (1998). Towards an Architecture for Virtual Enterprises. *Journal of Intelligent Manufacturing*, vol. 9, nº 2, pp. 189-199.
- (Camarinha-Matos *et al.*, 2003) Camarinha-Matos, M.; Afsarmanesh, H.; Rabelo, R. J., (2003). Infrastructure Developments for Agile Virtual Enterprises. *Int. Journal of Computer Integrated Manufacturing*, vol. 16, nº 4-5, pp. 235-254.
- (Camarinha-Matos *et al.*, 2005) Camarinha-Matos, L.M.; Silveri, I.; Afsarmanesh, H.; Oliveira, A.I., (2005). Towards a Framework for Creation of Dynamic Virtual Organizations. *Collaborative Networks and their Breeding Environments*, (PRO-VE'05), Springer, Valencia, Spain.
- (Camarinha-Matos *et al.*, 2005b) Camarinha-Matos, L.M.; Afsarmanesh, H.; Ollus, M., (2005). ECOLEAD: a Holistic Approach to Creation and Management of Dynamic Virtual Organizations. *Collaborative Networks and their Breeding Environments*, (PRO-VE'05), Springer, Valencia, Spain.
- (Camarinha-Matos *et al.*, 2005c) Camarinha-Matos, L. M.; Afsarmanesh, H.; Ollus, M., (2005). *Virtual Organizations: Systems and Practices*. Springer.
- (Camarinha-Matos y Afsarmanesh, 1999) Camarinha-Matos, L.M.; Afsarmanesh, H., (1999). The Virtual Enterprise Concept. En Camarinha-Matos, L.M. y Afsarmanesh, H. (eds.), *Infrastructures for Virtual Enterprises: Networking Industrial Enterprises*, Kluwer Academic Publishers, pp. 3-14.
- (Camarinha-Matos y Afsarmanesh, 2001) Camarinha-Matos L. M.; Afsarmanesh, H., (2001). Virtual Enterprise Modeling and Support Infrastructures: Applying Multi-Agent System approaches. *Multi-Agent Systems and Applications, Lecture Notes in Artificial Intelligence LNAI 2086*, pp. 335-364.
- (Camarinha-Matos y Afsarmanesh, 2003) Camarinha-Matos, L.M.; Afsarmanesh, H., (2003). Elements of a Base VE Infrastructure. *Journal of Computers in Industry*, vol. 51, nº 2, pp. 139-163.
- (Camarinha-Matos y Cardoso, 1999) Camarinha-Matos, L.M.; Cardoso, T., (1999). Selection of Partners for a Virtual Enterprise. *Infrastructures for Virtual Enterprises - Networking Industrial Enterprises*, Kluwer Academic Publishers, pp. 259-278.

- (Camarinha-Matos y Lima, 2000) Camarinha-Matos, L.M.; Lima, C., (2000). Supporting Business Process Management and Coordination in Virtual Enterprises. *Advances in Networked Enterprises*, Kluwer Academic Publishers, pp. 3-14.
- (Camarinha-Matos, 2002) Camarinha-Matos, L. M., (2002). Multi-Agent Systems in Virtual Enterprises. *Proceedings of AIS'2002- International Conference on AI, Simulation and Planning in High Autonomy Systems*, SCS publication, pp. 27-36, Lisboa, Portugal.
- (Camarinha-Matos, 2003) Camarinha-Matos, L.M., (2003). Infrastructures for Virtual Organizations - Where We Are. *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, vol. 2, pp. 405- 414.
- (Cardoso *et al.*, 2005) Cardoso, H.L.; Malucelli, A.; Rocha, A.P.; Oliveira, E., (2005). Institutional Services for Dynamic Virtual Organizations. *Collaborative Networks and their Breeding Environments*, (PRO-VE'05), Springer, pp. 521-528.
- (Cardoso *et al.*, 2006) Cardoso, H.; Rocha, A.P.; Oliveira, E., (2006). Virtual Organization Support through Electronic Institutions and Normative Multi-Agent Systems. *Handbook of Research on Nature Inspired Computing for Economy and Management*, vol 2, pp. 786-804.
- (Carrillo, 1995) Carrillo, F.J., (1995). El perfil emergente de la empresa virtual. *Transferencia*, año 8, número 31, pp. 24-26.
- (Castillo *et al.*, 1997) Castillo, E.; Gutierrez, J.; Hadi, A., (1997) . *Expert Systems and Probabilistic Network Models*. Springer, New York.
- (Castillo *et al.*, 1998) Castillo, E.; Gutiérrez, J.M.; Hadi, A.S. (1998). *Sistemas expertos y modelos de redes probabilísticas*. Monografías de la Academia de Ingeniería, DL. Madrid.
- (Castillo, 2003) Castillo, E., (2003). *Sistema de ejecución de workflows adaptable para la construcción de aplicaciones de comercio electrónico*. Tesis Universidad de las Américas.
- (Cernuzzi y Rossi, 2001) Cernuzzi, L.; Rossi, G., (2001). On the Evaluation of Agent oriented Methodologies. *Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies*.

- (Chaffey, 2002) Chaffey, D. (2002). *E-Business and E-Commerce Management*. Prentice Hall - Financial Times.
- (Chantal, 2002) Chantal, M. (2002). Explotación de los corpórea textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento. *Red iris*, vol. 18.
- (Charte, 1999) Charte, F. (1999). *Programación avanzada con C++ Builder 4* Editorial Anaya Multimedia.
- (Chesbrough y Teece, 1996) Chesbrough H.W., Teece, D. J. (1996). *Organizing for Innovation: When is Virtual Virtuous?*. Harvard Business Review.
- (Chess *et al.*, 1998) Chess, D., Harrison, C.G., Kershenbaum, A., (1998). Mobile Agents: Are They a Good Idea?. *G. Vigna (Ed.), Mobile Agents and Security, LNCS 1419*, Springer Verlag, pp. 25-47.
- (Chess, 1995) Chess, D., (1995). Itinerant Agents for Mobile Computing. *IEEE Personal Communications Magazine*, vol. 2, nº 5, pp. 34-59.
- (Chituc y Azevedo, 2005) Chituc, C.M.; Azevedo, A.L., (2005). Multi-Perspective Challenges on Collaborative Networks Business Environments. *Collaborative Networks and Their Breeding Environments*, Kluwer, pp. 25-32.
- (Choy *et al.*, 1999) Choy, S., Breugst, M., Magedanz, T., (1999). Beyond Mobile Agents with CORBA – Towards Mobile CORBA Objects. 6th ACTS Conference on Intelligence in Services and Networks (IS&N), pp. 168-180.
- (Christie y Levary, 1998) Christie, M. J.; Levary, R., (1998). Virtual Corporations: Recipe for Success. *Industrial Management*, vol. 40, nº 4, pp. 7-11.
- (Chu *et al.*, 2002) Chu, X.N.; Tso, S.K.; Zhang, W. ; Li, Q., (2002). Partnership Synthesis for Virtual Enterprises. *The International Journal of Advanced Manufacturing Technology*, vol. 19, pp. 384-391.
- (Clements *et al.*, 1997) Clements, P.E.; Papaioannou, T.; Edwards, J., (1997). Aglets: Enabling the Virtual Enterprise. En Proceedings of Mesela '97 - First International Conference on Managing Enterprises - Stakeholders, Engineering, Logistics and Achievement. p. 425.

- (Cliffè, 2004) Cliffè, O., (2004). *Basic Aspects of JADE Programming*. En (Jade web, 2008).
- (Cohen y Levesque, 1990) Cohen, P.R.; Levesque, H.J., (1990). Intention is Choice with Commitment. *Artificial Intelligence*, vol. 42, pp. 213-261.
- (Collis y Ndumu, 1999) Collis, J C.; Ndumu, D.T., (1999). The Role Modelling Guide. *Applied Research and Technology, BT Labs*. En (Zeus web, 2007).
- (Colmenauer y Roussel, 1996) Colmerauer, A.; Roussel, P. (1996). The Birth of Prolog. *History of Programming Languages*, ACM Press / Addison-Wesley, pp. 331-367.
- (COM, 2001) Comisión de las comunidades europeas, (2001). *El impacto de la economía electrónica en las empresas europeas: análisis económico e implicaciones políticas*. Comunicación de la comisión al consejo y al parlamento europeo.
- (Corchado y Molina, 2002) Corchado J. M.; Molina J.M., (2002). *Introducción a la Teoría de Agentes y Sistemas Multiagente*. Publicaciones Científicas, Editorial Catedral Publicaciones. Salamanca, España.
- (Cost *et al.*, 1998) Cost, R.; Finin, T.; Labrou, Y.; Luan, X.; Peng, Y.; Soboro, I., (1998). Jackal: A Java-based Tool for Agent Development. Working Notes of the AAAI '98 Workshop on Software Tools for Developing Agents.
- (Cost *et al.*, 2001) Cost, R. S.; Labrou, Y.; Finin, T., (2001) Agent Coordination Using an Agent Communication Language. *Coordination of Internet Agents: Models, Technologies and Application*, Springer-Verlag.
- (Cuenca *et al.*, 2002) Cuenca, F.A.; Gaitán, J.J.; Hlavickova, V., (2002). Prácticas internacionales integradas de e-business. VIII Jornadas de Enseñanza Universitaria de la Informática, pp. 59-66.
- (Cuesta, 1998) Cuesta, F. (1998). *La empresa virtual. La estructura cosmos*. Serie McGraw-Hill de management. McGraw-Hill. Madrid.

- (Dang, 2004) Dang, V.D., (2004). *Coalition Formation and Operation in Virtual Organisations*. PhD, School of Electronics and Computer Science, University of Southampton.
- (Davidow y Malone, 1992) Davidow, W.H.; Malone, M.S., (1992). *The Virtual Corporation: Structuring and Revitalizing the Corporation for the 21st Century*. Harper-Collins Publishers.
- (Davidrajuh y Deng, 2000) Davidrajuh, R.; Deng, Z.Q., (2000). Identifying Potential Supplier for Formation of Virtual Manufacturing Systems. En Proceedings of 16th IFIP World Computer Congress 2000, vol. ITBM, Beijing, China.
- (Davidrajuh y Deng, 2000b) Davidrajuh, R.; Deng, Z.Q., (2000). An Autonomous Data Collection System for Virtual Manufacturing Systems. *International Journal of Agile Management Systems*, vol 2, nº 1, pp. 7-15.
- (Davis y Smith, 1983) Davis, R.; Smith, R., (1983). Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*, vol. 20, pp. 63-109.
- (Dean y Schniederjans, 1991) Dean, B.; Schniederjans, M. (1991). A Multiple Objective Selection Methodology for Strategic Industry Selection Analysis. *IEEE Transactions on Engineering Management*, vol. 38, pp. 53-62.
- (Dennis y Wixom, 2000) Dennis, A.; Wixom, B.H., (2000). *Systems Analysis and Design: An Applied Approach*. John Wiley and Sons.
- (Dignum y Dignum, 2002) Dignum, V.; Dignum, F., (2002). Towards an Agent-Based Infrastructure to Support Virtual Organisations. En L.M. Camarinha-Matos, ed., *Collaborative Business Ecosystems and Virtual Enterprises*, Kluwer Academic Publishers, pp. 363-370.
- (Do et al., 2000) Do, V.T.; Halatchev, M.; Neumann, D., (2000), A Contextbased Approach to Support Virtual Enterprises, Proceedings of the 33rd Hawaii International Conference on System Sciences, vol. 1, nº 4-7.
- (Doz y Hamel, 1998) Doz, Y.L.; Hamel, G., (1998). *Alliance Advantage. The Art of Creating Value Through Partnering*, Harvard Business School Press.

- (Durking, 1994) Durkin, J., (1994). *Expert Systems: Design and Development*. Maxwell Macmillan.
- (Dyer, 1990) Dyer, J. S., (1990). Remarks on the Analytic Hierarchy Process. *Management Science*, vol. 36, nº 3, pp. 249-258.
- (Efstathiou y Rajkovic, 1979) Efstathiou, J.; Rajkovic, V., (1979). Multiattribute Decision Making Using a Fuzzy Heuristic Approach. *IEEE Transactions on Systems, Man, and Cybernetics SMC-9*. pp. 326-333.
- (Evans, 2001) Evans, R., (ed.), (2001). *MESSAGE: Methodology for engineering systems of software agentes. Final*. Technical Report P907-T11, EURESCOM.
- (Feigenbaum y Feldman, 1963) Feigenbaum, E.; Feldman, J., (1963). *Computers and Thought*. McGraw-Hill.
- (Feigenbaum, 1977) Feigenbaum, E.A., (1997). The Art of Artificial Intelligence: Themes and Case Studies in Knowledge Engineering. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'77), pp. 1014-1029.
- (Feng y Yamashiro, 2003) Feng, D. Z.; Yamashiro, M., (2003). A Pragmatic Approach for Optimal Selection of Plantspecific Process Plans in a VE. *Production Planning & Control*, vol. 14, nº 6, pp. 562-570.
- (Ferber, 1999) Ferber, J., (1999). *Multi-Agent Systems. An Introduction to Distributed Artificial Inteligence*. Addison-Wesley.
- (Fernández y Gómez-Pérez, 2002) Fernández M.; Gómez-Pérez A., (2002). Overview and Análisis of Methodologies for Building Ontologies. *The Knowledge Engineering Review*, vol. 17, nº 2, pp. 129-156.
- (Field y Hoffner, 2002) Field, S.; Hoffner, Y., (2002). In Search of the Right Partner. In Camarinha-Matos, L. M., editor, *Collaborative Business Ecosystems and Virtual Enterprises*, Kluwer Academic Publishers, pp. 55-62.
- (Fielding *et al.*, 1998) Fielding, R.T.; Whitehead, E.J.; Anderson, K.M.; Bolcer, G.A.; Oriely, P.; Taylor, R.N., (1998). Support for the Virtual Enterprise: web-based Development of Complex Information Products. *Communications of the ACM*, vol. 41, nº 8, pp. 84-92.

- (Finin *et al.*, 1997) Finin, T., Labrou, Y., Mayfield, J., (1997). KQML as an Agent Communication Language. *Software Agents*. MIT Press, Cambridge, pp. 291-316.
- (Fischer *et al.*, 1996) Fischer, K.; Heimig, I.; Org, J.; Uller, P., (1996). Intelligent Agents in Virtual Enterprises. First International Conference on the Practical Application of Intelligent Agents and Multi Agent Technology, pp. 205-223.
- (Fischer *et al.*, 2004) Fischer, M.; Jähna, H.; Teich, T., (2004). Optimizing the Selection of Partners in Production Networks. *Robotics and Computer-Integrated Manufacturing*, vol. 20, pp. 593-601.
- (Flores-Mendez, 2000) Flores-Mendez, R.A., (2000). Towards a Standardization of Multi-Agent System Frameworks. *ACM Crossroads, Special Issue on Intelligent Agents*, vol. 5, n° 4, pp. 18-14.
- (Foner, 1993) Foner L. N., (1993). *What's an Agent, Anyway? A Sociological Case Study*. Agents Memo 93-01, Agents Group, MIT Media Lab.
- (Forbairt, 1996) Forbairt, (1996), *Virtual Corporation Defined*. Summary Section for Forbairt Internet Report, Forbairt, Ireland.
- (Forgy y McDermott, 1977) Forgy, C.; McDermott, J., (1977). OPS: a Domain-Independent Production System Language. Proceedings of the 5th International Joint Conference on Artificial Intelligence, pp. 933-939.
- (Foster *et al.*, 2001) Foster, I.; Kesselman, C.; Tuecke, S., (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Lecture Notes in Computer Science*, vol. 2150, pp. 1-4.
- (Fox *et al.*, 1993) Fox, M.S.; Chionglo, J.F.; Fadel, F.G., (1993). A Common Sense Model of the Enterprise. Proceedings of the Second Industrial Engineering Research Conference. Norcross GA: Institute for Industrial Engineers, pp. 425-429.
- (Franklin y Graesser, 1997) Franklin S.; Graesser A., (1997). Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Intelligent Agents III*, Berlin. Springer Verlag, pp. 21-35.

- (Frederix, 1998) Frederix, F.L.M., (1998). Agility and Human Factors in the Virtual Enterprise. Proceedings of the Conference on Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility, and the Virtual Enterprise, pp. 737-748.
- (Friedman-Hill, 2007) Friedman-Hill, E.J., (2007). *Jess. The Rule Engine for the Java Platform*. En (Jess web, 2008).
- (Galeano *et al.*, 2005) Galeano,G; Apolinar, D.; Zubiaga, G.; Atahualpa, J.; González, I.; Molina, A., (2005) . Virtual Breeding Environment: A First Approach to Understand Working and Sharing Principles. *Interoperability of Enterprise Software and Applications*, Springer, pp. 99-110.
- (García y Pavón, 2000) García, D.; Pavón, J., (2000). *Introducción al Estándar FIPA*. Informe Técnico UCM-DSIP 98-00, Departamento de sistemas informáticos y Programación de la UCM.
- (García-Serrano y Ossowski, 1998) García-Serrano, A.; Ossowski S., (1998). Inteligencia Artificial Distribuida y Sistemas Multiagente. *Revista Iberoamericana de Inteligencia Artificial*. Número 6, Monografía.
- (Garijo, 2005) Garijo, F., (2005). Tecnología de agentes: orígenes, conceptos y características. En *Agentes software y sistemas multiagente : conceptos, arquitecturas y aplicaciones*. Prentice Hall, pp. 1-28.
- (Garneau y Delisle, 2002) Garneau, T., Delisle, S., (2002). Programmation orientée-agent: evaluation comparative d'outils et environnements. JFIADSMA 2002, Lille, France.
- (Genesereth *et al.*, 1994) Genesereth, M.R.; Singh, N.P.; Syed M.A., (1994). A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation. *International Journal of Cooperative Information Systems*, vol. 4, nº 4, pp. 339-367.
- (Genesereth y Ketchpel, 1994) Genesereth M.R.; Ketchpel, S.P., (1994). Software Agents. *Communication of the ACM*, vol. 37, nº 7, pp. 48-ff.
- (Georgakopoulos, 1998) Georgakopoulos, D., (1998). Collaboration Management Infrastructure for Comprehensive Process and Service Management. International Symposium on Advanced Database Support for Workflow Management, Enschede, The Netherlands.

- (Georgeff *et al.*, 1999) Georgeff, M.; Pell, B.; Pollack, M.; Tambe, M.; Wooldridge, M., (1999). The Belief-Desire-Intention Model of Agency. *Lecture Notes In Computer Science*, vol. 1555, pp. 1-10.
- (Georgeff y Rao, 1991) Georgeff, M.P.; Rao, A.S., (1991). Modeling Rational Agents Within a BDI-Architecture. Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann publishers Inc., pp. 473-484.
- (Geppert *et al.*, 1998) Geppert, A.; Kradolfer, M.; Tombros, D., (1998). Market-Based Workflow Management. *Lecture Notes in Computer Science*, vol. 1402, pp. 179-187.
- (Giarratano, 2001) Giarratano, J., (2001). *Sistemas expertos: principios y programación*. Ed. Thompson.
- (Giménez y Lourenço, 2004) Gimenez, C.; Lourenço, H.R., (2004). *E-Supply Chain Management: Review, Implications and Directions for Future Research*. UPF Working Paper #769.
- (Giret *et al.*, 2005) Giret, A.; Julian, V.; Botti, V., (2005). Aplicaciones Industriales de los Sistemas Multiagente. *Agentes Software y Sistemas Multi-Agente. Conceptos, Arquitecturas y Aplicaciones*, pp. 186-203.
- (Gisela y Korsunsky, 2003) Gisela, A.; Korsunsky, L. (2003). *Análisis de los proyectos de la unión europea para la creación y soporte de redes de empresas virtuales y supply chain management*. Fondazione Istituto per il Lavoro - working paper.
- (Goldman *et al.*, 1995) Goldman, S.L.; Nagel, R.N.; Preiss, K. (1995). *Agile Competitors and Virtual Organizations: Strategy for Enriching the Customer*. Van Nostram Reinhold.
- (Gómez y Bosque, 2004) Gómez, M.; Bosque, J., (2004). Aplicación de análisis de incertidumbre como método de validación y control del riesgo en la toma de decisiones, *GeoFocus* (Artículos), nº 4, pp. 179-208.
- (Gómez-Pérez *et al.*, 2003) Gómez-Pérez, A.; Fernández-López M.; Corcho, O., (2003). *Ontological Engineering: Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer-Verlag.

- (Gómez-Pérez, 1999) Gómez-Pérez, A. (1999). Ontological Engineering: A State of the Art. *Expert Update, British Computer Society*, vol. 2, nº 3, pp. 33-43.
- (Gómez, 2002) Gómez, J.J., (2002). *Modelado de sistemas multiagente*. PhD thesis, Departamento de Sistemas Informáticos y Programación. Universidad Complutense de Madrid.
- (Gómez, 2003) Gómez, J.J., (2003). Metodologías para el desarrollo de sistemas multi-agentes. *Revista Iberoamericana de Inteligencia Artificial*, nº 18, pp. 51-63.
- (Goranson, 1999) Goranson, H.T., (1999). *The Agile Virtual Enterprise: Cases, Metrics, Tools*. Quorum Books.
- (Gornev et al., 1997) Gornev, V.F.; Rarassov, V.B.; Soenen, R.; Tahon, K., (1997). Virtual Enterprise : Reasons, Sources and Tools. *Proceeding of MCPL '97*, Campinas, Brazil. pp. 53-58.
- (Green et al., 1997) Green, S.; Hurst, L.; Nangle, B.; Cunningham, P.; Somers, F.; Evans, R., (1997). *Software Agents: A Review*. Intelligent Agents Group Report.
- (Grefen et al., 2000) Grefen, P.; Aberer, K.; Hoffner, Y.; Ludwig, H., (2000). CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises. *International Journal of Computer Systems Science & Engineering*, vol. 15, pp. 277-290.
- (Gutiérrez, 2001) Gutiérrez, J.M., (2001). *Sistemas Expertos Basados en Reglas*. En (Citeseer web, 2008).
- (Haddadi y Sundermeyer, 1996) Haddadi, A.; Sundermeyer, K., (1996). Belief-desire-intention agent architectures. *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, pp. 169-185.
- (Hagel y Singer, 1999) Hagel J.; Singer M., (1999). Unbundling the Corporation. *Harvard Business Review*, pp. 133-141.
- (Hales y Barker, 2000) Hales, K.R.; Barker, J.R., (2000). *Searching for the Virtual Enterprise*. Technical Report 00/01. School of Information Technology, Bond University. Australia.

- (Hales y Barker, 2003) Hales, K.R.; Barker, J.R., (2003). *Value Creation and the Virtual Enterprise*. Technical Report 03/04. School of Information Technology, Bond University. Australia.
- (Hales, 2005) Hales, K.R., (2003). *Value Creation in a Virtual World*. Tesis doctoral, Bond University. Australia.
- (Hamel y Prahalad, 1994) Hamel, G.; Prahalad, C.K., (1994) *Compiendo por el futuro*. Ariel.
- (Hammer, 2000) Hammer, M., (2000). The Rise of the Virtual Enterprise. *Information Week*, vol. 778, n° 3, pp. 152.
- (Hampton *et al.*, 1997) Hampton, L.; Martin, R. C.; Pitt, F. G.; Ottinger, T., (1997). *A Critique of Use Cases*. En: <http://ootips.org/use-cases-critique.html>.
- (Handy, 1989) Handy, C., (1989). *The Age of Unreason*. Harvard School Press.
- (Hannus *et al.*, 2004) Hannus, M.; Anastasiou, M.; Bifulco, A.; Findeisen, P.; Karvonen, I.; Löh, H.; Plüss, A.; Ollus, M.; Weidemann, M., (2004). *Guidelines for Virtual Organizations*, VOSTER project consortium. Published by VTT Technical Research Centre of Finland. En (VE-Forum web, 2008).
- (Hao *et al.*, 2003) Hao, Q.; Shen, W.; Wang, L.; Lang, S., (2003). Cooperative Scheduling for Inter-Enterprise Manufacturing Resources Sharing. *COVE Newsletters*, n° 5, pp. 7-16.
- (Hao *et al.*, 2005) Hao, Q.; Shen, S.; Wang, L., (2005). Towards a Cooperative Distributed Manufacturing Management Framework. *Computers in Industry*, vol. 56, n° 1, pp.71-84.
- (Hayes-Roth *et al.*, 1983) Hayes-Roth, F.; Waterman, D. A.; Lenat, D. B., (eds.) (1983). *Building Expert Systems*. Addison-Wesley, Reading, MA.
- (Heuvel y Maamar, 2003) Heuvel, W., Maamar, Z., (2003). Moving toward a Framework to compose Intelligent Web Services. *Communications of the ACM*, vol. 46, n° 10, pp. 103-109.

- (Hoffmann, 2001) Hoffmann, J., (2001). Virtual Enterprise. *GMD Research Series*, nº 18.
- (Hoffner y Crawford, 1997) Hoffner, Y.; Crawford, B.,(1997). Using Interception to Create Domains in Distributed Systems. Proceedings of the IFIP/IEEE international conference on Open distributed processing and distributed platforms, pp. 251-263.
- (Hoffner y Schade, 1998) Hoffner, Y.; Schade, A., (1998). Co-operation, Contracts, Contractual Match-Making and Bindings. Proceedings Second International Enterprise Distributed Object Computing Workshop, vol. 3, nº3, pp. 75-86.
- (Hoffner, 1999) Hoffner, Y., (1999). Supporting Contract Match-Making. Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises, IEEE Computer Society, p. 64.
- (Ibañez *et al.*, 2004) Ibáñez, M.; Rosell, J.I.; Rosell, J.R. (2004). *Tecnología solar*. Mundi-Prensa. Madrid.
- (Iglesias *et al.*, 1999) Iglesias, C.A.; Garijo, M.; González, J.C.; Velasco, J.R., (1999). Analysis and Design of Multiagent Systems using MASCommonKADS. *Lecture Notes in Artificial Intelligence. Intelligent Agents IV. Agents Theories, Architectures and Language*, vol. 1365, pp. 1-16.
- (Iglesias, 1997) Iglesias, C.A., (1997). *Fundamentos de los agentes inteligentes*. Informe Técnico UPM/DIT/GSI 16/97, Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid.
- (Iglesias, 1998) Iglesias, C. A., (1998). *Definición de una metodología para el desarrollo de sistemas multiagente*. Tesis Doctoral. Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid.
- (Ingham, 1999) Ingham, J., (1999) *What is an Agent*. Technical Report #6/99. Centre for Software Maintenance of the University of Furham. Durham.

- (Ip *et al.*, 2003) Ip, W.H.; Huang, M.; Yung, K.L.; Wang, D., (2003). Genetic Algorithm Solution for a Risk-based Partner Selection Problem in a Virtual Enterprise. *Computers & Operations Research*, vol. 30, pp. 213-231.
- (Ip *et al.*, 2004) Ip, W. H.; Yung, K. L.; Wang, D., (2004). A Branch and Bound Algorithm for sub-Contractor Selection in Agile Manufacturing Environment. *International Journal of Production Economics*, vol. 87, nº 2, pp. 195-205.
- (ISO/IEC, 1998) International Organization for Standardization / International Electrotechnical Commission (1998). *9596, Information Technology, Open Systems Interconnection, Common Management Information Protocol (CMIP) - Part 1: Specification*. En (ISO web, 2005).
- (Jacobson *et al.*, 2000) Jacobson, I.; Booch, G.; Rumbaugh, J., (2000). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.
- (Jagers *et al.*, 1998) Jagers, H.; Jansen, W.; Steenbakkens, W., (1998). Characteristics of Virtual Organisations. *Organizational Virtualness, Proceedings of the First VoNet- Workshop*, vol. 2, nº 1, pp. 65-76.
- (Jansen *et al.*, 1999) Jansen, W.; Steenbakkens, W.; Jäger, H., (1999). Electronic Commerce and Virtual Organizations. *The Electronic Journal of Organizational Virtualness*, vol. 1, nº 1, pp. 54-68.
- (Jarimo y Salo, 2006) Jarimo, T.; Salo, A., (2006). *An Optimisation Model for Partner Selection in Virtual Organisations*. VTT Technical Research Centre of Finland, VTT, Finland, Working Paper.
- (Jarimo, 2006) Jarimo, T., (2006). *Partner Selection and Utility Sharing in Collaborative Networks*. Licentiate's Thesis. Helsinki University of Technology.
- (Jennings *et al.*, 1996) Jennings, N.R., Faratin, P., Johnson, M.J., O'Brien, P., and Wiegand, M.E., (1996). Using Intelligent Agents to Manage Business Processes. Proceedings of the First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM96), London, pp. 345-360.

- (Jennings *et al.*, 1998) Jennings R.; Sycara K.; Wooldridge M., (1998). A Roadmap of Agent Research and Development. *Autonomous Agents and Multiagent Systems*, vol 1, nº 1, pp. 275-306.
- (Jennings y Bussmann, 2003) Jennings, N. R.; Bussmann, S., (2003). Agent-Based Control Systems. Why Are They Suited to Engineering Complex Systems. *IEE Control Systems Magazine*, vol. 23, nº 3, pp. 61-73.
- (Jennings y Wooldridge, 1998) Jennings, N.R.; Wooldridge, M.J., (eds.), (1998). *Agent Technology: Foundations, Applications and Markets*. Springer-Verlag.
- (Jennings, 1993) Jennings, N. R., (1993). Commitments and Conventions: the Foundations of Coordination in Multiagent Systems *Knowledge Engineering Review*, vol. 8, nº 3, pp. 223-250.
- (Jennings, 1995) Jennings, N. R., (1995). Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions. *Artificial Intelligence*, vol. 75, nº 2, pp. 195-240.
- (Julián y Botti, 2000) Julián, V.J.; Botti, V., (2000). Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial. *Revista Novatica*, pp. 95-99.
- (Julián y Botti, 2003) Julián, V.J.; Botti, V.J., (2003). Estudio de métodos de desarrollo de sistemas multiagente. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, nº 18, pp 65-80.
- (Kaihara, 1999) Kaihara, T. (1999). Supply Chain Management Based on Market Mechanism in Virtual Enterprise. PRODNET Working Conference on Infrastructures for Virtual Enterprises: Networking Industrial Enterprises, pp. 399-408.
- (Kandel, 1992) Kandel, A., (1992). *Fuzzy Expert Systems*. CRC Press.
- (Kanet *et al.*, 1999) Kanet, J.J.; Faisst, W.; Mertens, P., (1999). Application of Information Technology to a Virtual Enterprise Broker: the Case of Bill Epstein. *International journal of production economics*, vol. 62, pp. 23-32.

- (Kaslingam y Lee, 1996) Kaslingam, R.; Lee, C., (1996). Selection of Vendors: a Mixed Integer Programming Approach. *Computers and Industrial Engineering*, vol. 31, pp. 347-350.
- (Katzy y Dissel, 2001) Katzy, B.R.; Dissel, M.C., (2001). A Toolset for Building the Virtual Enterprise. *Journal of Intelligent Manufacturing*, vol. 12, nº 2, pp. 121-131.
- (Katzy y Loeh, 2003) Katzy, B.R.; Loeh, H. (2003). Virtual Enterprise Research State of the Art and Ways Forward. *International Conference on Concurrent Engineering*, Espoo, Finland, pp. 343-354.
- (Katzy y Miralles, 1999) Katzy, B.R.; Miralles, F., (1999). *The Virtual Enterprise - Thriving on Turbulent Change*. Barcelona Management Review.
- (Kelly *et al.*, 1996) Kelly, S.; Lyytinen, K.; Rossi, M., (1996). MetaEdit+ A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment. *Lecture Notes in Computer Science*, vol. 1080, pp. 1-21.
- (Kendall, 1998) Kendall, E., (1998). Agent Roles and Role Models: New Abstractions for Multiagent System Analysis and Design. Proceedings of the International Workshop on Intelligent Agentes in Information and Process Management, Bremen, Germany.
- (Kinny *et al.*, 1997) Kinny, D.; Georgeff, M.; Rao, A., (1997). A Methodology and Modelling Technique for Systems of BDI Agents. Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Alemania, vol. 1038.
- (Kinny y Georgeff, 1997) Kinny, D.; Georgeff, M., (1997). *Modelling and Design of Multi-Agent Systems*. Springer Verlag.
- (Klen, 1999) Klen, A., (1999). Distributed Business Process Management. PRODNET Working Conference on Infrastructures for Virtual Enterprises: Networking Industrial Enterprises, pp. 241-258.
- (Kolodner, 1993) Kolodner, J., (1993). *Case-Based Reasoning*. Morgan Kaufmann Publishers.
- (Kotler, 1995) Kotler, P., (1995). *Dirección de Marketing*. Prentice-Hall.

- (Kotler, 2000) Kotler, P., (2000). *Marketing Management: Analysis, Planning, Implementation and Control*. Prentice-Hall.
- (Kowalski, 1974) Kowalski, R.A., (1974). *Logic for Problem Solving*. DCL Memo 75, Department of Artificial Intelligence, University of Edinburgh, Scotland.
- (Kowalski, 1974b) Kowalski, R.A., (1974). Predicate Logic as a Programming Language. *Proc. IFIP 74*. North Holland, Amsterdam, pp. 569–574.
- (Labrou y Finin, 1998) Labrou, Y.; Finin, T., (1998). Semantics for an Agent Communication Language. *Intelligent Agents IV: Agent Theories, Architectures and Languages, Lecture Notes in Computer Science*, Springer Verlag, vol. 1365, pp. 209-214.
- (Lambert *et al.*, 1998) Lambert, D. M.; Cooper, M. C.; (1998). Supply Chain Management: Implementation issues and research opportunities. *The International Journal of Logistics Management*, vol. 9, nº 2, pp. 1-19.
- (Lange y Oshima, 1999) Lange, D. B. y Oshima, M., (1999). Seven Good Reasons for Mobile Agents. *Communications of the ACM*, vol. 42, nº 3, pp. 88-90.
- (Lau y Wong, 2001) Lau, H.C.W.; Wong, T.T., (2001). Partner Selection and Information Infrastructure of a Virtual Enterprise Network. *International Journal of Computer Integrated manufacturing*, vol. 14, nº 2, pp. 186-195.
- (Laubacher y Malone, 1997) Laubacher, R.J.; Malone, T.W.; the MIT Scenario Working Group, (1997). *Two Scenarios for 21st Century Organizations: Shifting Networks of Small Firms or All-Encompassing Virtual Countries?*. MIT Initiative on Inventing the Organizations of the 21st Century, Working Paper N°. 001.
- (Laudon y Traver, 2002) Laudon, K.C.; Traver, C.G., (2002). *E-Commerce - business, technology and society*. Addison Wesley.
- (Lee, 1998) Lee, R.M., (1998). Towards Open Electronic Contracting Using Electronic Trade Scenarios. *Journal of Electronic Markets Special Issue on Electronic Contracting*, vol. 2, nº 1.

- (Li *et al.*, 2000) Li, Y.; Huang, B.Q.; Liu, W. H.; Wu, C.; Gou, H.M., (2000). Multi-Agent System for Partner Selection of Virtual Enterprises. Proceedings of 16th IFIP World Computer Congress 2000, Vol. ITBM.
- (Lipnack y Stamps, 1993) Lipnack, J.; Stamps, J., (1993). One Plus One Equals Three. *Small Business Reports*, vol. 18, nº 8, pp. 49-58.
- (Lizárraga, 2002) Lizárraga, C., (2002). *Web Services*. Laboratorio de Física Interdisciplinaria. Departamento de Física. Universidad de Sonora.
- (Luger y Stubblefield, 1989) Luger, G. F.; Stubblefield, W. A., (1989). *Artificial Intelligence and the Design of Expert Systems*. Benjamin/Cummings Co.
- (Magedanz, 1995) Magedanz, T., (1995). On the Impacts of Intelligent Agent Concepts on Future Telecommunication Environments. *Lecture Notes on Computer Science 998- Bringing Telecommunication Services to the People - IS&N'95*, pp. 396-414.
- (Malone y Rockart, 1991) Malone, T. W.; Rockart, J. F., (1991). Computers, Networks, and the Corporation. *Scientific American*, vol. 265, nº 3. pp. 128-136.
- (Malucelli *et al.*, 2005) Malucelli, A.; Lopes, E.; Oliveira, E.C., (2005). Enriching a MAS Environment with Institutional Services. *Environments for Multiagent Systems II, Lecture Notes in Computer Science 3830*, pp. 105-120.
- (Malucelli y Oliveira, 2004) Malucelli, A., Oliveira, E., (2004). Ontology-Services Agent to Help in the Structural and Semantic Heterogeneity. *Virtual Enterprises and Collaborative Networks*, Kluwer Academic Publishers, pp. 175-182.
- (Margulis y Sagan, 1996) Margulis, L.; Sagan, D., (1996). *¿Qué es la vida?*. Tusquets Editores.
- (Martinez *et al.*, 2001) Martinez, M.T.; Fouletier, P.; Park, K.H.; Favrel, J., (2001). Virtual Enterprise - Organisation, Evolution and Control. *International Journal Production economics*, vol. 74, nº 1-3, pp. 225-238.

- (Martins *et al.*, 2004) Martins, L.L.; Gilson, L.L.; Maynard, M.T., (2004). Virtual Teams: What Do We Know and Where Do We Go from Here?. *Journal of Management*, vol. 30, nº 6, pp. 805-835.
- (Mas, 2005) Mas, A. (Coord.) (2005). *Agentes Software y Sistemas Multiagente. Conceptos, Arquitecturas y Aplicaciones*. Pearson-Prentice Hall.
- (Matskin *et al.*, 2005) Matskin, M. ; Kungas, P.; Rao, J.; Sampson, J.; Petersen, S.A., (2005). Enabling Web Services Composition with Software Agents. Proceedings of Internet and Multimedia Systems, and Applications, IMSA 2005, pp.93-98.
- (McCaffer y Garas, 1999) McCaffer, R.; Garas, F. (eds), (1999). eLSEwise: European Large Scale Engineering Wide Integration Support Effort. *Engineering Construction and Architectural Management*, Special Issue, vol. 6, nº 1.
- (Meade *et al.*, 1997) Meade, L.; Liles, D.; Sarkis, J., (1997). Justifying Strategic Alliances and Partnering: a Prerequisite for Virtual Enterprising. *Omega*, vol. 25, nº 1, pp. 29–42.
- (Medjahed *et al.*, 2003) Medjahed, B.; Benatallah, B.; Bouguettaya, A.; Ngu, A.; Elmagarmid, A.K., (2003). Business-to-Business Interactions: Issues and Enabling Technologies. *The VLDB Journal The International Journal on Very Large Data Bases*, vol. 12, nº 1, pp. 59–85.
- (Mejia y Molina, 2002) Mejia, R.; Molina, A., (2002). Virtual Enterprise Broker: Processes, Methods and Tools. Third Working Conference on Infrastructures for Virtual Enterprises: Collaborative Business Ecosystems and Virtual Enterprises, pp. 81-90.
- (Mertens *et al.*, 1998) Mertens, P.; Griese, J.; Ehrenberg, D., (1998). *Virtuelle Unternehmen und Informationsverarbeitung*. Springer Verlag.
- (Mikhailov, 2002) Mikhailov, L., (2002). Fuzzy Analytical Approach to Partnership Selection in Formation of Virtual Enterprises. *Omega*, vol 30, nº 5, pp. 393-401.
- (Miles y Snow, 1984) Miles, R.E.; Snow, C.C. (1984). Fit, Failure and the Hall of Fame. *California Management Review*, vol. 26, nº 3, pp. 10-28.

- (Miles y Snow, 1986) Miles, R.E.; Snow, C.C. (1986). Organizations: New Concepts for New Forms. *California Management Review*, vol. 28, pp. 62-73.
- (Miller, 1995) Miller, G., (1995). WordNet: A Lexical Database for English. *Communication of ACM*, vol. 38, nº 11, pp. 39-41.
- (Milojicic *et al.*, 1998) Milojicic, D.S.; Breugst, M.; Busse, I.; Campbell, J.; Covaci, S.; Friedman, B.; Kosaka, K.; Lange, D.B. Kouichi; Oshima, M.; Tham, C.; Virdhagriswaran, S.; White, J. (1998). MASIF: The OMG Mobile Agent System Interoperability Facility. MA'98: Mobile agents. International workshop N°2, vol. 1477, pp. 50-67.
- (Mixteo *et al.*, 2004) Mitxco, J.; Idigoras, I.; Vicente, A., (2004). Los clusters como fuente de competitividad: el caso de la Comunidad Autónoma del País Vasco. *Revista Cuadernos de Gestión*, vol. 4, nº 1, pp. 55-67.
- (Molina *et al.*, 2004) Molina, J.M.; García, J.; Bernardos, A.M., (2004). Agentes y Sistemas Multiagente. *Tecnologías y Servicios para la Sociedad de la Información*, Consejo Social de la Universidad Politécnica de Madrid, capítulo 4.
- (Mowshowitz, 1997) Mowshowitz, A., (1997). Virtual Organization. *Communications of the ACM*, vol. 40, nº 9, pp. 30-37.
- (Mowshowitz, 2000) Mowshowitz, A., (2000). *Virtual Organization*. Quorum Books.
- (Nagalingam y Lin, 1999) Nagalingam, S.V.; Lin, G.C.I., (1999). Latest Developments in CIM. *Robotics and Computer Integrated Manufacturing*, vol. 15, nº 6, pp. 423-430.
- (Namin *et al.*, 2006) Namin, A.S.; Shen, W.; Ghenniwa, H., (2006). Web Services /Agent-Based Model for Inter-Enterprise Collaboration. *Emerging Solutions for Future Manufacturing Systems*, vol. 159, pp. 231-240.
- (Nayak *et al.*, 2001) Nayak, N.; Bhaskaran, K.; Das, R., (2001). Virtual Enterprises. Building Blocks for Dynamic e-Business. *Australian Computer Science Communications*, vol. 23, nº 6, pp. 80-87.

- (Nayak *et al.*, 2001b) Nayak, N.; Chao, T.; Li, J.; Mihaeli, J.; Das, R.; Derebail, A.; Hoo, J.S., (2001). Role of Technology in Enabling Dynamic Virtual Enterprises. Proceedings of International Workshop on Open Enterprise Solutions: Systems, Experiences and Organizations, Rome, Italy.
- (Nikraz *et al.*, 2006) Nikraz, M.; Caire, G.; Bahri, P.A., (2006). A Methodology for the Analysis and Design of Multi-Agent Systems Using JADE. *International Journal of Computer Systems Science and Engineering*.
- (Norman *et al.*, 2004) Norman, T.J.; Preece, A.; Chalmers, S.; Jennings, N.R.; Luck, M.; Dang, V.D.; Nguyen, T.D.; Deora, V.; Shao, J.; Gray, A.; Fiddian, N., (2004). Agent-Based Formation of Virtual Organisations. *International Journal of Knowledge Based Systems*, vol. 17, n° 2-4, pp.103-111.
- (Noy y McGuinness, 2001) Noy, N.F.; McGuinness, D.L., (2001). *Ontology Development 101: a Guide to Creating your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.
- (Nwana *et al.*, 1999) Nwana, H.S.; Ndumu, D.; Lee, L.; Collis, J., (1999). ZEUS: A Toolkit for Building Distributed Multi-Agent Systems. *Applied Artificial Intelligence Journal*, vol. 13, n° 1, pp. 129-186.
- (Nwana y Wooldridge, 1997) Nwana, H.S.; Wooldridge, M., (1997). Software Agent Technologies. *Software Agents and Soft Computing Towards Enhancing Machine Intelligence*, vol. 1198/1997, pp. 59-78.
- (Nwana, 1996) Nwana, H.S., (1996). Software Agents: an Overview. *The Knowledge Engineering Review*, vol. 11, n° 3, pp. 205-244.
- (Nydicck y Hill, 1992) Nydicck, R.; Hill, R., (1992). Using the AHP to Structure the Supplier Selection Procedure. *Journal of Purchasing and Materials Management*, vol. 25, n° 2, pp. 31-36.
- (Odell *et al.*, 2000) Odell, J.; Van Dyke Parunak, H.; Bauer, B., (2001). Extending UML for Agents. Proc.of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence, pp. 3-7.

- (Odell *et al.*, 2001) Odell, J.; Van Dyke Parunak, H., Bauer, B., (2001). Representing Agent Interaction Protocols in UML. *First International Workshop, AOSE 2000 on Agent-Oriented Software Engineering* (Secaucus, NJ, USA), pp. 121–140.
- (Odell, 2000) Odell, J., (2000). Objects and agents: how do they differ?. *Journal of Object-Oriented Programming*, vol. 13, nº 6, pp. 50-53.
- (Oliveira y Camarinha-Matos, 2005) Oliveira, A.I.; Camarinha-Matos, L.M. (eds.) (2005). *Requirements and mechanisms for VO planning and launching*. ECOLEAD Deliverables. En (VE-forum web, 2006).
- (Oliveira y Rocha, 2000) Oliveira, E.; Rocha, A.P., (2000). Agents Advanced Features for Negotiation in Electronic Commerce and Virtual Organisations Formation Process. *Agent Mediated Electronic Commerce: The European AgentLink Perspective, LNAI*, pp. 78-97.
- (Ong y Narasimhalu, 1998) Ong, L.; Narasimhalu, A., (1998). Case-Based Reasoning. *The Handbook of Applied Expert Systems*. CRC Press LCC, Capítulo 11.
- (Orfali *et al.*, 1996) Orfali, R.; Harkey, D.; Edwards, J., (1996). *The Essential Distributed Object Survival Guide*. John Wiley & Sons Inc.
- (Osorio y Barata, 2001) Osorio, A. L.; Barata, M.M., (2001). Reliable and Secure Communication Infrastructure for Virtual Enterprises. *Journal of Intelligent Manufacturing*, vol. 12, nº 2, pp. 173-181.
- (Ossadnik, 1996) Ossadnik, W., (1996). AHP-Based Synergy Allocation to Partners in a Merger. *European Journal of Operational Research*, vol. 88, nº 1, pp. 42-49.
- (Osterle *et al.*, 2000) Osterle, H.; Fleisch, E.; Alt, R., (2000). *Business Networking: Shaping Enterprise Relationships on the Internet*. Springer.
- (Ouzounis, 2001) Ouzounis, E. K., (2001). *An Agent-Based Platform for the Management of Dynamic Virtual Enterprises*. Tesis doctoral de la Facultad de Electrónica e Informática de la Universidad de Berlin.

- (Park y Favrel, 1999) Park, K.H.; Favrel, J., (1999). Virtual Enterprise. Information System and Networking Solution. *Computer & Industrial Engineering*, vol. 37, nº 1-2, pp. 441-444.
- (Partovi *et al.*, 1989) Partovi, F.; Burton, J.; Banerjee, A., (1989). Application of the AHP in Operations Management. *International Journal of Operations and Production Management*, vol. 10, pp. 5-19.
- (Pavón y Gómez, 2005) Pavón, J.; Gómez, J., (2005). Métodos y Herramientas. En *Agentes Software y Sistemas Multiagente: Conceptos, Arquitecturas y Aplicaciones*, Prentice Hall, pp. 145-183.
- (Pelechano *et al.*, 2005) Pelechano, E.; García, F.; Soriano, I., (2005). Las organizaciones virtuales frente a los sistemas tradicionales de coordinación de la actividad económica. *Revista Madri+d*, nº 28.
- (Peng *et al.*, 1998) Peng, Y.; Finin, T.; Labrou, Y.; Chu, G.; Long, J.; Tolone, W.J.; Boughannam, (1998). A Multi-Agent System for Enterprise Integration. Proceedings of the Third International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, pp 155-169.
- (Pérez, 1996) Pérez, C. (1996). Nueva concepción de la tecnología y sistema nacional de innovación. *Cuadernos de CENDES*, Año 13, nº 31, pp. 9-33.
- (Petersen *et al.*, 2002) Petersen, S. A.; Rao, J.; Tveit, A., (2002). Virtual Enterprises: Challenges in Selecting and Integrating Computational and Human Resources. 1st International Workshop on Challenges in Open Agent Systems.
- (Petersen *et al.*, 2003) Petersen, S.A.; Rao, J.; Matskin, M., (2003). Virtual Enterprise Formation with Agents - an Approach to Implementation. IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03), p. 527.
- (Petersen, 2003) Petersen, S.A., (2003). *An Agent-based Approach to Support the Formation of Virtual Enterprises*. Tesis Doctoral de la Universidad de Ciencia y Tecnología de Noruega.
- (Petrie y Bussler, 2003) Petrie, C.; Bussler, C., (2003). Service Agents and Virtual Enterprises: A Survey. *IEEE Internet Computing*, vol. 7, nº 4, pp. 68-78.

- (Picard, 2005) Picard, W., (2005). Modeling Structured non-Monolithic Collaboration Processes. *Collaborative Networks and their Breeding Environments*, (PRO-VE'05), Valencia, Spain, pp. 379-386.
- (Porter, 1985) Porter, M.E., (1985). *Competitive advantage - Creating and sustaining superior performance*. The Free Press.
- (Porter, 1990) Porter, M.E., (1990). *The Competitive Advantage of Nations*. Free Press.
- (Posadas, 2000) Posadas, J.L., (2000). *Arquitecturas de procesos en sistemas reactivos distribuidos. Agentes inteligentes*. Trabajo para la obtención de la suficiencia investigadora, Departamento de Informática de Sistemas y Computadores de la Universidad Politécnica de Valencia.
- (Prahalad y Hamel, 1990) Prahalad, C.; Hamel, G., (1990). *The Core Competence of the Corporation*, Harvard Business Review.
- (Pressman, 1982) Pressman, R.S., (1982). *Software Engineering: A Practitioner's Approach*. Series in Software Engineering and Technology. McGraw-Hill, Inc.
- (Rabelo *et al.*, 1998) Rabelo, R.; Camarinha-Matos, L.M.; Afsarmanesh, H., (1998). Multiagent Perspectives to Agile Scheduling. Proceedings of IEEE/IFIP International Conference on Balanced Automation Systems, pp. 51-66.
- (Rabelo *et al.*, 2000) Rabelo, R.; Camarinha-Matos, L.M.; Vallejos, R. (2000). Agent-based Brokerage for Virtual Enterprise Creation in the Moulds Industry. *E-business and Virtual Enterprises*, Kluwer Academic Publishers, pp.281-290.
- (Rabelo *et al.*, 2000b) Rabelo, R.; Afsarmanesh, H.; Camarinha-Matos, L.M., (2000). Federated Multi-Agent Scheduling in Virtual Enterprises. *E-business and Virtual Enterprises*, Kluwer Academic Publishers, pp. 145-156.
- (Rabelo y Camarinha-Matos, 1994) Rabelo, R.; Camarinha-Matos, L.M., (1994). Negotiation in Multi-Agent based Dynamic Scheduling. *Int. Journal on Robotics and CIM*, vol. 11, nº 4, pp. 303-309.

- (Ramoneda, 2006) Ramoneda, J.(2006). *El diario de Ramoneda*. Hora 25, Cadena Ser. 14 de marzo de 2006.
- (Rao y Georgeff, 1995) Rao, A.S.; Georgeff, M., (1995). BDI Agents: from Theory to Practice. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp 312-319.
- (Rao y Petersen, 2003) Rao, J.; Petersen, S. A., (2003). Implementing Virtual Enterprises using AGORA Multi-agent System. *Information Systems for a Connected Society, CAiSE'03 Forum*. Klagenfurt/Velden, Austria, pp. 193-196.
- (Reichert y Dadam, 1998) Reichert, M.; Dadam, P., (1998). ADEPTflex. Supporting Dynamic Changes of Workflows Without Losing Control. *Journal of Intelligent Information Systems (JIIS)*, Special Issue on Workflow Management Systems, vol. 10, n° 2, pp. 93-109.
- (Reid *et al.*, 1996) Reid, L.; Liles, D.H.; Rogers, K.J.; Johnson, M.E., (1996). A Methodology for Engineering Virtual Enterprise. *Proceedings of the 5th Agility Forum Conference, USA*, pp. 329-338.
- (Ricordel y Demanzeau, 2001) Ricordel, P.M.; Demanzeau, Y., (2001). *Programmation Orientée Multi-Agents. Développement et Déploiement de systèmes Multi-Agents Voyelles*". Tesis doctoral. Instituto Nacional Politécnico de Grenoble.
- (Rocha *et al.*, 2005) Rocha, A.P.; Cardoso, H.L.; Oliveira, E., (2005). Contributions to an Electronic Institution Supporting Virtual Enterprises' Life Cycle. *Virtual Enterprise Integration: Technological and Organizational Perspectives*, Idea Group, Inc., pp. 229-246.
- (Rocha y Oliveira, 1999) Rocha, A.; Oliveira, E., (1999). An electronic market architecture for the formation of virtual enterprises. *Infrastructures for Virtual Enterprises*, Kluwer.
- (Rocha, 2001) Rocha, A.P., (2001). *Metodologias de Negociação em Sistemas Multi-Agentes para Empresas Virtuais*. Tesis doctoral. Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia, Universidade do Porto, Porto.
- (Rodríguez-Aguilar, 2001) Rodríguez-Aguilar, J.A., (2001). *On the Design and Construction of Agent-mediated Electronic Institutions*. Ph.D. Thesis, Universitat Autònoma de Barcelona.

- (Roussel, 1975) Roussel, P., (1975). *PROLOG: Manuel d'utilisation*. Groupe d'intelligence Artificielle, Université d'Aix-Marseille, Luminy, France.
- (Russel y Norvig, 1995) Russell, S.; Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Ed. Prentice Hall.
- (Saaty y Alexande, 1989) Saaty T.; Alexande, J.M., (1989). *Conflict Resolution: the Analytic Hierarchy Approach*. Praeger.
- (Saaty, 1980) Saaty, T.L., (1980). *The Analytic Hierarcy Process*. McGraw-Hill.
- (Santesmases, 1993) Santesmases, M., (1993). *Marketing. Conceptos y estrategias*. Ed. Pirámide. Madrid.
- (Sanz *et al.*, 2003) Sanz, P.; Benito, J.J.; Olmo, R., (2003). La empresa virtual: una solución de futuro. V Congreso de Ingeniería de Organización. Valladolid, Spain..
- (Sanz *et al.*, 2004) Sanz, P.; Benito, J.J.; Araújo, A., (2004). La solución organizativa del siglo XXI: la empresa virtual. XIV Jornadas Luso-Españolas de Gestión Científica. Ponta Delgada. Azores.
- (Sanz *et al.*, 2004b) Sanz, P.; Benito, J.J.; Araújo, A.; Olmo, R., (2004). CIM y Empresa Virtual: la integración como forma de supervivencia. VII Congreso de Ingeniería de Organización. Leganés. Spain
- (Sanz *et al.*, 2005) Sanz, P.; Benito, J.J.; Araújo, A., (2005). An agent-based framework for selection of partners in dynamic virtual enterprises. Proceedings of The XV Jornadas Luso-Españolas de Gestión Científica. Sevilla. Spain.
- (Sanz *et al.*, 2005b) Sanz, P.; Benito, J.J.; Araújo, A.; Olmo, R., (2005). Los sistemas multi-agentes en la formación de Empresas Virtuales Dinámicas. Proceedings of The IX Congreso de Ingeniería de Organización. Oviedo. Spain.
- (Sanz *et al.*, 2005c) Sanz, P.; Benito, J.J.; Araújo, A.; Olmo, R., (2005). Partner Selection in Dynamic Virtual Enterprises using Multi-Agent Systems. Proceedings of The 2005 International Conference in Artificial Intelligence (ICAI'05). Las Vegas, Nevada, USA

- (Sanz *et al.*, 2007) Sanz, P.; Benito, J.J.; Araúzo, A.; Olmo, R., (2007). Justificación del modelo organizativo de empresa virtual dinámica interorganizacional a partir de la comparativa entre los modelos de empresa virtual. Proceedings of the International Conference on Industrial Engineering and Industrial Management. Madrid. Spain.
- (Sanz, 2004) Sanz, P., (2004). *Empresas Virtuales Dinámicas. Un Estudio*. Trabajo para la Obtención de la Suficiencia Investigadora. Universidad de Valladolid.
- (Schank, 1982) Schank, R. (Ed.), (1982). *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press.
- (Schreiber *et al.*, 2000) Schreiber, A.; Akkermans, J.; Anjewierden, A., (2000). *Engineering of Knowledge and Management: The COMMONKADS Methodology*. MIT Press.
- (Shehory y Sturm, 2001) Shehory, O.; Sturm, A., (2001). *Evaluation of Modeling Techniques for Agent-based Systems*. ACM press.
- (Shen y Norrie, 1998) Shen W.; Norrie, D.H., (1998). An Agent-based Approach for Distributed Manufacturing and Supply Chain Management. *Globalization of Manufacturing in the Digital Communications Era of the 21st Century*, pp. 579-590.
- (Shen y Norrie, 1998b) Shen, W.; Norrie, D.H., (1998). An Agent-Based Approach for Dynamic Manufacturing Scheduling. Proceedings of Autonomous Agents'98 Workshop on Agent-Based Manufacturing, pp. 117-118.
- (Shen y Norrie, 1999) Shen, W.; Norrie, D.H., (1999). Agent-based Systems for Intelligent Manufacturing: A State-of-the-art Survey. *Knowledge and Information Systems*, vol. 1, n° 2, pp. 129-156.
- (Shoham, 1993) Shoham, Y. (1993). Agent-oriented Programming. *Artificial Intelligence*, vol. 60, n° 1, pp. 51-92.
- (Shoham, 1997) Shoham, Y. (1997). An Overview on Agent-oriented Programming. *Software Agents*, AAAI Press, pp. 271-290.

- (Sieber, 1997) Sieber, P., (1997). Virtual Organizations: Static and Dynamic Viewpoints. *VoNet: The Newsletter*, vol. 1, n° 2, pp. 3-9.
- (Smith, 1980) Smith, R.G., (1980). *The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver*. Kaufmann.
- (Snow *et al.*, 1999) Snow, C.; Lipnack, J.; Stamps, J., (1999). The Virtual Organization: Promises and Payoffs, Large and Small. *Trends in Organizational Behavior*, vol. 6, pp. 15-30.
- (Spinosa *et al.*, 1998) Spinosa, L.M.; Rabelo, R.; Klen, A.P., (1998). High-level Coordination of Business Processes in a Virtual Enterprise. Conference on Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility, and the Virtual Enterprise, pp. 725-736.
- (Stair y Reynolds, 2002) Stair, R.M.; Reynolds, G.E., (2002). *Principles of Information Systems*. Course Technology.
- (Stevens, 1984) Stevens, L., (1984). *Artificial Intelligence. The Search for the Perfect Machine*. Hayden Book Company, Hasbrouck Heights.
- (Stewart, 1997) Stewart, T.A., (1997). *Intellectual Capital: The New Wealth of Organizations*. Doubleday & Company.
- (Strader *et al.*, 1998) Strader, T.J.; Lin, F.R.; Shaw, M.J., (1998). Information Infrastructure for Electronic Virtual Organization Management. *Decision support Systems*, vol. 23, n° 1, pp. 75-94.
- (Stricker *et al.*, 2000) Stricker, C.; Riboni, S.; Kradolfer, M.; Taylor, J., (2000). Market-based Workflow Management for Supply Chains of Services. 33rd Hawaii Int Conference on System Sciences (HICSS-33), Maui, Hawaii.
- (Stylianou *et al.*, 1992) Stylianou, A.C.; Madley, G.R.; Smith, R.D., (1992). Selection Criteria for Expert System Shells: a Socio-Technical Framework. *Communications of the ACM*, vol. 35, n° 10, pp. 30-48.

- (Suguri *et al.*, 2001) Suguri, H.; Kodama, E.; Miyazaki, M.; Nunokawa, H.; Noguchi, S., (2001). Implementation of FIPA Ontology Service. Proceedings of the Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents. Montreal, Canada.
- (Svirskas *et al.*, 2007) Svirskas, A.; Ignatiadis, I.; Roberts, B.; Wilson, M., (2007). Virtual Organization Management Using Web Service Choreography And Software Agents. *Network-Centric Collaboration and Supporting Frameworks*. Springer Boston, vol. 224, pp. 535-542.
- (Swagerman *et al.*, 2001) Swagerman, D.M.; Wassenaar, D.A.; Van Elk M., (2001). Business Reporting in a Virtual Environment based on the Semiotic Framework, *Electronic Journal of Virtual Organization*, vol. 3, n° 7, pp. 80-106.
- (Talluri *et al.*, 1999) Talluri, S.; Baker, R.; Sarkis, J., (1999). A Framework for Designing: eLcient Value Chain Networks. *International Journal on Production Economics*, vol. 62, pp. 133-144.
- (Tansley y Hayball, 1993) Tansley, D.S.W.; Hayball, C.C., (1993). *Knowledge Based systems Analysis and Design a KADS developer's handbook*. Prentice Hall.
- (Terzis *et al.*, 1999) Terzis, S., Nixon, P., Wade, V., Dobson, S., Fuller, J., (1999). *The Future of Enterprise Groupware Applications*. Computer Science Department. Trinity College Dublin.
- (Tombros y Geppert, 2000) Tombros, D.; Geppert, A., (2000). Building Extensible Workflow Systems using an Event-Based Infrastructure. Proc. 12th Conf. on Advanced Information Systems Engineering, Stockholm, Sweden.
- (Tombros, 1999) Tombros, D., (1999). *An Event- and Repository-Based Component Framework for Workflow System Architecture*. PhD thesis, University of Zurich.
- (Van Heijst *et al.*, 1997) Van Heijst, G.; Schreiber, A.T.; Wielinga, B.J., (1997). Using Explicit Ontologies in KBS Development. *International Journal of Human and Computer Studies*, vol. 46, n° 2-3, pp. 183-292.
- (Vaucher y Ncho, 2004) Vaucher, J.; Ncho, A. (2004). *Jade Primer*. En (Jade web, 2008).

- (Venkatraman y Henderson, 1998) Venkatraman, N.; Henderson, J.C., (1998). Real Strategies for Virtual Organizing. *MIT, Sloan Management Review*, vol 40, nº 1, pp. 33-48.
- (Wang, 2000) Wang, S., (2000) Meta-Management of Virtual Organizations: Toward Information Technology Support, *Electronic Networking Applications and Policy*, vol. 10, nº 5, pp. 451-458.
- (Waterman, 1986) Waterman, D.A., (1986). *A Guide to Expert Systems*. Addison-Wesley.
- (Watson, 1997) Watson, I., (1997). *Applying Case Based Reasoning: Techniques for enterprise systems*. Morgan Kauffman.
- (Weiss y Kulikowski, 1984) Weiss, S. M.; Kulikowski, C.A., (1984). *A Practical Guide to Designing Expert Systems*. Rowman and Allanheld.
- (Weyns *et al.*, 2005) Weyns, D.; Parunak, H.V.D.; Michel, F.; Holvoet, T.; Ferber, J., (2005) . Environments for Multiagent Systems: State-of-the-art and Research Challenges. *Environments for Multiagent Systems, Lecture Note in Artificial Intelligence (LNAI)*, Vol. 3477, pp. 1-47.
- (Wijk *et al.*, 1998) Wijk, J.; Geurts, D.; Bultje, R., (1998). 7 Steps to Virtuality: Understanding the Virtual Organisation Process before Designing the ICT Support. *Objects, Components and the Virtual Enterprise, an interdisciplinary workshop at OOPSLA 98*.
- (Wognum y Faber, 1999) Wognum, P.M.; Faber, E.C.C., (1999). A Framework for Improving the Quality of Operation in a Virtual Enterprise. *Infrastructures for Virtual Enterprises. Networking Industrial Enterprises*. PRODNET Working Conference for Virtual Enterprises (PROVE'99), Porto, Portugal, p. 365-376.
- (Wooldridge *et al.*, 2000) Wooldridge, M.; Jennings, N.; Kinny, D., (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 3, nº 3, pp. 285-312.
- (Wooldridge y Ciancarini, 2001) Wooldridge, M.; Ciancarini, P., (2001). Agent-Oriented Software Engineering: The State of the Art. *Agent-Oriented Software Engineering. Lecture Notes in AI*, vol 1957, pp. 1-24.

- (Wooldridge y Jennings, 1995) Wooldridge, M.; Jennings, N., (1995). Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, vol. 10, nº 2, pp. 115-152.
- (Wooldridge, 1997) Wooldridge, M., (1997). Agent-based Software Engineering. *IEEE Transactions on Software Engineering*, vol. 144, nº1, pp. 26-37.
- (Wooldridge, 2002) Wooldridge, M., (2002). *An Introduction to Multiagent Systems*. John Wiley and Sons.
- (Wu y Su, 2005) Wu, N.; Su, P., (2005). Selection of Partners in Virtual Enterprise Paradigm. *Robotics and Computer-Integrated Manufacturing*, vol. 21, nº 2, pp. 119-131.
- (Zadeh, 1965) Zadeh, L.A., (1965). Fuzzy Sets. *Inf. Control*, vol. 8, pp. 338-353.
- (Zambonelly *et al.*, 2000) Zambonelly, F.; Wooldridge, M.; Jennings N.R., (2000). Organisational Rules as an Abstraction for the Analysis and Design of Multi-Agent Systems. *International Journal of Software Engineering and knowledge Engineering*, vol. 11, nº 3, pp. 303-328.
- (Zarli y Poyet, 1997) Zarli, A.; Poyet, P., (1997). Integrating Emerging IT paradigms for the Virtual Enterprise: the VEGA Project. *Proceedings of ICE 97, 4th International Conf. On Concurrent Engineering*, Nottingham, UK.
- (Zarli y Poyet, 1999) Zarli, A.; Poyet, P., (1999). A Framework for Distributed Information Management in the Virtual Enterprise: The VEGA Project. *Infrastructures for Virtual Enterprises. Networking Industrial Enterprises*. PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, pp. 293-306.

RECURSOS WEB¹²⁹

- (Accenture, 2004) “*Las reglas de Accenture para triunfar en la economía digital*”. <http://www.marketingycomercio.com/numero11/11reglas.html>.
- (Agentbuilder web, 2006) “*AgentBuilder site*”. <http://www.agentbuilder.com/>
- (Aglets, 2006) “*Aglets Software Development Kit from IBM*”. <http://www.trl.ibm.co.jp/aglets/>
- (ALIVE web, 2006) “*The ALIVE Project*”. <http://www.vive-ig.net/projects/alive/>
- (ASIF web, 2008) “*Asociación de la Industria Fotovoltaica*” <http://www.asif.org/>
- (AUML web, 2007) “*The FIPA Agent UML Web Site*”. <http://www.auml.org>
- (BeanGenerator web, 2007) “*OntologyBeanGenerator*”. <http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator>
- (BIDSAVER web, 2006) “*BIDSAVER, Business Integrator Dynamic Support Agents for Virtual Enterprise*”, <http://www.ceconsulting.it/ve/bidsaver.html>
- (Bioclimatic web, 2007) “*Bioclimatic Ingeniería*” <http://www.bicingeneria.com>
- (BMO web, 2007) “*BMO, Business Management Ontology*”. <http://ontologyonline.org/visualisation/c/Directory/Bmo+-+business+management+ontology/>
- (CE-NET II web, 2006) “*CE-NET II, Concurrent Enterprising Network of Excellence*”. <http://www.ce-net.org/>
- (Citeseer web, 2008) “*CiteSeer: Scientific Literature Digital Library*”. <http://citeseer.ist.psu.edu/citeseer.html>
- (CLIPS web, 2007) “*CLIPS, A Tool for Building Expert Systems*”. <http://clipsrules.sourceforge.net/>

129 La fecha que aparece en las referencias web hace referencia a la última visita.

- (COVE web, 2008) “*COVE - CO-operation infrastructure for Virtual Enterprises and Electronic Business*”.
<http://www.uninova.pt/~cove/coveproject.htm>
- (Crossflow web, 2006) Crossflow Project, (1998). “ESPRIT Project 28635 Cross-Organisational Workflow Management”. The Crossflow Consortium, <http://www.crossflow.org>
- (Cuesta, 2005) Cuesta, P., (2005). Página de la asignatura Sistemas Multiagente. Universidad de Vigo.
<http://trevinca.ei.uvigo.es/~pcuesta/sm/sm.html>.
- (DanteConsulting, 2003) “*Dante Consulting*”. <http://www.dante-consulting.com/>
- (Dell web, 2006) “Página oficial de Dell”. <http://www.dell.com/>
- (e-COGNOS web, 2006) “*e-COGNOS, Methodology, tools and architectures for electronic COnsistent knowledGe maNagement across prOjects and between enterpriSes in the construction domain*”. <http://e-cognos.cstb.fr/>
- (e-COLLEG web, 2006) “*e-COLLEG, Advanced Infrastructure for Pan-European Collaborative Engineering*”. <http://www.ecolleg.org/>
- (e-MMEDIATE web, 2006) “*e-MMEDIATE, Electronic Managing of Product Manufacturing Engineering*”. <http://www.eu-emmediate.com/>
- (Eclipse web, 2008) “*Plataforma de desarrollo abierta Eclipse*”.
<http://www.eclipse.org/>
- (ECOLEAD web, 2006) “*ECOLEAD, European Collaborative Networked Organisations Leadership Initiative*”. <http://ecolead.vtt.fi/>
- (elEconomista web, 2008) “*elEconomista.es*”
<http://www.eleconomista.es/index.html>
- (Energy web, 2007) “*Energy Spain*”
<http://www.energy-spain.com/energia-solar>
- (ENTERPRISE web, 2007) “*ENTERPRISE ontology*”.
<http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html>

- (EXTERNAL web, 2006) “EXTERNAL, Extended Enterprise Resources, Network Architectures and Learning”. <http://research.dnv.com/external/>
- (FIPA OSS, 2005) “FIPA OSS – Ontology Service Specification”. <http://www.fipa.org/specs/fipa00086/XC00086D.html>
- (FIPA web, 2007) “IEEE Foundation for Intelligent Physical Agents”. <http://www.fipa.org/>
- (GIMP web, 2008) “GNU Image Manipulation Program”. <http://www.gimp.org/>
- (GLOBMEN web, 2006) “GLOBMEN, Global Engineering and Manufacturing in Enterprise Networks”. <http://cic.vtt.fi/projects/globmen/>
- (GNOSIS web, 2006) “GNOSIS Virtual Factory”. http://www.vtt.fi/aut/tau/network/ims/web/ims_gnos.htm
- (GRASIA web, 2007) “GRASIA, Grupo de Agentes Software: Ingeniería y Aplicaciones”. <http://grasia.fdi.ucm.es/>
- (IBM web, 2007) “International Business Machines (IBM)”. <http://www.ibm.com/es/>
- (ICCI web, 2006) “ICCI - Innovation co-ordination, transfer and deployment through networked Co-operation in the Construction Industry”. <http://cic.vtt.fi/projects/icci/>
- (III web, 2007) “Informática Integral Inteligente web”. <http://www.informaticaintegral.net/sisexp.html>
- (IMS web, 2006) “Intelligent Manufacturing Systems”. <http://www.ims.org/>
- (INGENIAS web, 2006) INGENIAS (2006): <http://grasia.fdi.ucm.es/ingenias/>
- (ISO web, 2006) “International Organization for Standardization”. <http://www.iso.org/iso/home.htm>
- (IST web, 2006) “IST, Information Society Technologies”. <http://cordis.europa.eu/ist/about/era.htm>
- (ISTforce web, 2006) “ISTforce - Intelligent Services and Tools for Concurrent Engineering”. <http://www.ISTforCE.com>

- (JADE web, 2007) “*JADE - Java Agent Development Framework*”. <http://jade.tilab.com/index.html>
- (Jadex web, 2008) “*Jadex BDI Agent System*”. <http://vsis-www.informatik.uni-hamburg.de/projects/jadex/>
- (JESS web, 2007) “*Jess, the rule engine for the java platform*”. <http://herzberg.ca.sandia.gov/jess/>
- (JessTab web, 2007) “*JessTab: Integrating Protégé and Jess*”. <http://www.ida.liu.se/~her/JessTab/>
- (KM FORUM web, 2006) “*KM FORUM - The European Knowledge Management Forum*”. <http://www.knowledgeboard.com/>
- (KSE web, 2006) “*KSE, ARPA Knowledge Sharing Effort*”. <http://www.cs.umbc.edu/kse/>
- (labelec web, 2007) “*Laboratorio de electrónica de la Escuela Politécnica Superior de Cáceres*”. http://www.info-ab.uclm.es/labelec/Solar/Solar_Fotovoltaica/index.htm
- (LEAP web, 2006) “*The Lightweight Extensible Authentication Protocol (LEAP) IST Project*”. <http://leap.crm-paris.com>
- (Mappy web, 2007) “*Portal cartográfico Mappy*”. <http://www.mappy.com>
- (MASSIVE web, 2006) “*MASSIVE - Multiagent Manufacturing Agile Scheduling Systems for Virtual Enterprises*”. <http://centaurus.dee.fct.unl.pt/massive>
- (Matisse web, 2006) “*Matisse Project Home Page*”. <http://form.netbeans.org/>
- (Mozilla web, 2008) “*Página web de Mozilla en España-Europa*”. <http://www.mozilla-europe.org/es>
- (Multiagent Systems web, 2007) “*Multiagent Systems. Tools for building MAS*”. http://www.multiagent.com/Software/Tools_for_building_MAS/s/index.html
- (Netbeans web, 2008) “*El Entorno de Desarrollo Integrado Netbeans*”. <http://www.netbeans.org/>

- (NIIP web, 2003) “*NIIP, National Industrial Information Infrastructure Protocols*”. <http://www.niip.org>.
- (OMG web, 2006) “*OMG, Object Management Group*”. <http://www.omg.org/>.
- (ONTOBLOGIA web, 2007) “*ONTOBLOGIA, Ontologías, Lógica Computacional e Inteligencia Artificial*”. <http://www.cs.us.es/~joaquin/blog/index.php/category/economia-y-semantic/>
- (Open Office web, 2008) “*La suite Open Office*”. <http://es.openoffice.org/>
- (PRODCHAIN web, 2006) “*PRODCHAIN, Development of a decision support methodology to improve logistics performance in production networks*”. <http://www.prodchain.net>
- (PRODNET II, 2006) “*PRODNET II, Production Planning and Management in an Extended Enterprise*”. <http://www.uninova.pt/~prodnet/>
- (Protégé web, 2007) “*Protégé web site*”. <http://protege.stanford.edu/>.
- (REA web, 2007) “*REA Enterprise Ontology Source Page*”. <http://www.msu.edu/user/mccarth4/rea-ontology/>
- (Reticular web, 2006) “*Reticular Systems, Inc.*”. <http://www.reticular.com>
- (SCFAB web, 2006) “*Semiconductor Marketing Associates: The Semiconductor Industry's Fab Information*”. <http://www.scfab.com>.
- (StandishGroup web, 2008) “*The Standish Group*”. <http://www.standishgroup.com/>
- (SUN web, 2007) “*Sun Microsystems*”. <http://java.sun.com>
- (SYMPHONY web, 2006) “*SYMPHONY, A Dynamic Management Methodology with Modular and Integrated Methods and Tools for Knowledge Based, Adaptive SMEs*”. http://www.ims.org/projects/project_info/symphony.html
- (THINKCreative web, 2006) “*THINKcreative - Thinking Network of experts on emerging smart organizations*”. <http://www.Uninova.pt/~thinkcreative/>

- (TiLab web, 2007) “*TiLab, Telecom Italia Lab*”. <http://www.telecomitalialab.com>
- (TOVE web, 2007) “*TOVE Ontology Project*”.
<http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html>
- (Ubuntu web, 2008) “*Distribución Linux Ubuntu*”.
<http://www.ubuntu-es.org/>
- (UML web, 2006) “*UML, Unified Modeling Language*”. <http://www.uml.org/>
- (UPV MAS, 2007) “*Curso de doctorado de la Universidad Politécnica de Valencia sobre Sistemas Multiagente*”. <http://www.upv.es/sma/index.htm>
- (VE-Forum web, 2008) “*VE-Forum.org, página web oficial*”. <http://www.ve-forum.org/>
- (Via Michelin web, 2007) “*Portal cartográfico Via Michelin*”. <http://www.viamichelin.es>
- (VOSTER web, 2008) “*VOSTER, Virtual Organisations Cluster*”.
<http://cic.vtt.fi/projects/voster/projects.html>
- (W3C web, 2006) “*W3C - The World Wide Web Consortium*”. <http://www.w3.org/>.
- (WfMC web, 2006) “*The Workflow Management Coallition Homepage*”.
<http://www.wfmc.org/>
- (Wikipedia web, 2008) “*Wikipedia*”. <http://es.wikipedia.org>.
- (Zeus web, 2006) “*Zeus, open source agent development tool kit*”.
<http://labs.bt.com/projects/agents/zeus/>