

E.T.S. INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE VALLADOLID  
Mención en tecnologías de la información



# Workflow Designer

Manual de usuario

**Autor:**

Fernando Merino Cejudo

**Tutores:**

Benjamín Sahelices Fernández (UVA)

Carlos Caño y Diego Próspero (HPE)

# Índice

<b>1. Traducir flujos</b>	1
1.1 Mediante el uso de comandos . . . . .	1
1.2 Mediante el uso del plug-in de Eclipse . . . . .	2
<b>2. Validar los flujos</b>	4

# 1. Traducir flujos

## 1.1. Mediante el uso de comandos

Como intérprete de nuestro XSLT hemos utilizado "saxon" ya que aparte de ser de código libre y uno de los más eficientes y flexibles que hemos encontrado entre muchos otros, está implementado tanto en java como .NET y JS.

Al menos que se disponga de la versión licenciada de *saxon* (dispone de ambas) tendremos que descargarnos la versión *Saxon-B* mediante el siguiente enlace:

<https://sourceforge.net/projects/saxon/files/Saxon-B/9.1.0.8/saxonb9-1-0-8j.zip/download>.

Una vez descargada la aplicación, para realizar las transformaciones bastaría con ejecutar el siguiente comando:<sup>1</sup>

```
java -jar saxon.jar -s:Source.xml -xsl:Converter.xsl -o:output.[xml|bpmn]
```

Por ejemplo, si quisiéramos traducir un flujo HPE (BuyerProcess.xml) a BPMN ejecutaríamos lo siguiente:

```
java -jar saxon.jar -s:BuyerProcess.xml -xsl:Activator_to_BPMN.xsl \  
-o:BuyerProcess.bpmn
```

O si por el contrario si quisiéramos realizar la traducción inversa:

```
java -jar saxon.jar -s:BuyerProcess.bpmn -xsl:BPMN_to_Activator.xsl \  
-o:BuyerProcess.xml
```

---

<sup>1</sup>Indicar que para la correcta ejecución del comando necesitaremos tener instalada una versión de java compatible, por ejemplo la 1.6 o 1.7 ( si es que optamos por la implementación en java)

## 1.2. Mediante el uso del plug-in de Eclipse

Para convertir un flujo mediante el plug-in de Eclipse lo primero que necesitamos es instalar dicho plugin, para ello nos situaremos en el directorio donde se encuentra la instalación de Eclipse y entraremos en la carpeta *plugins*, por último, en esta última carpeta pegaremos el fichero *com.hpe.e4.converter\_1.0.0.201607202003.jar* que se encuentra en el zip del disco adjunto dentro de la carpeta *Developed*.

Una vez copiado a la carpeta indicada iniciaremos Eclipse<sup>2</sup> y ya podremos convertir cualquier flujo HPE a BPMN y viceversa.

Para realizar dicha transformación basta con situarse en el fichero que se desea convertir y automáticamente el sistema mostrará la opción disponible en base a su extensión; si el fichero tiene una extensión *.xml* el plug-in mostrará la opción para intentar convertir dicho fichero a bpmn, si por el contrario la extensión del fichero seleccionado es *.bpmn*, la opción que mostrará será convertirlo a HPE como se muestra en la siguiente imagen:

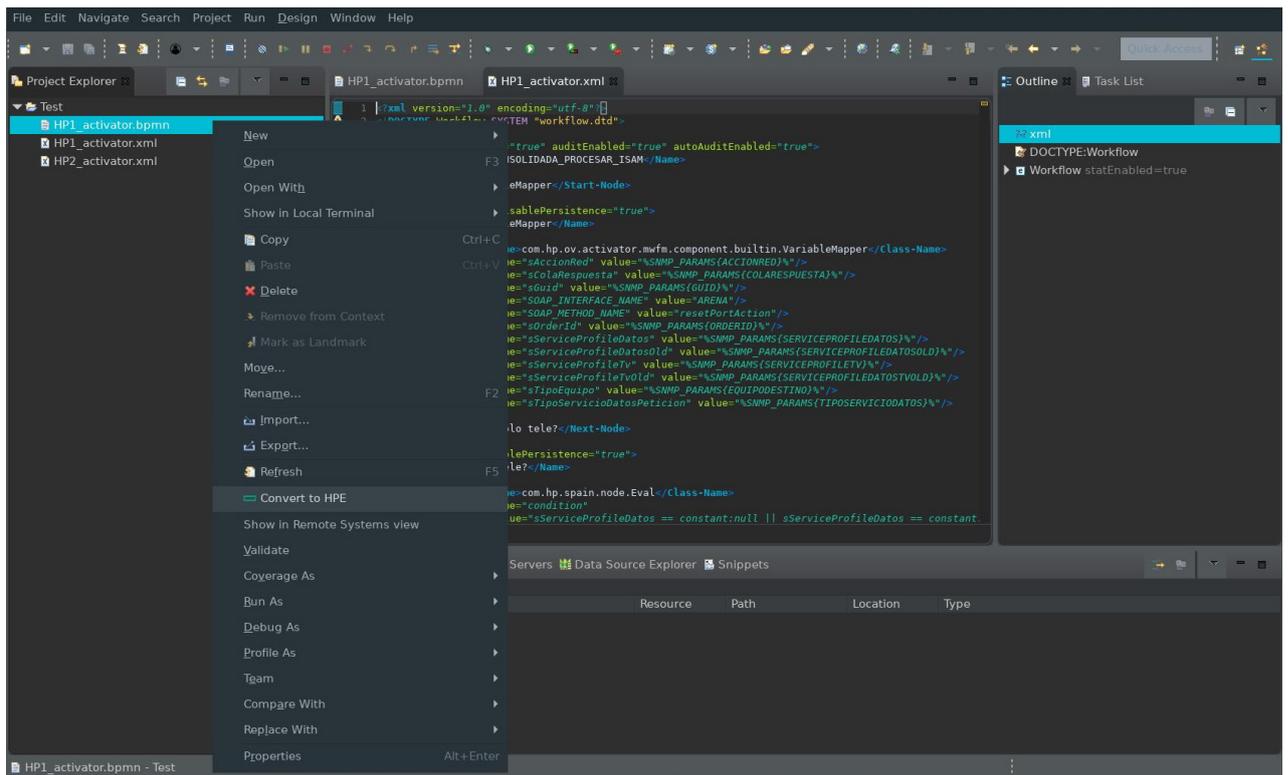


Figura 1: Ejemplo del plug-in para Eclipse

<sup>2</sup> El plugin se ha probado con todas las versiones de Eclipse Neon y con la recién estrenada Oxygen

Se ha desarrollado para la versión Neon de Eclipse en Java 1.8 , aunque es compatible con versiones de java anteriores ( se ha probado con java 1.6 y 1.7).

Una vez indicada la conversión a realizar el traductor creará un fichero con el mismo nombre que el fichero a traducir pero con la extensión modificada.

Como comentábamos en la memoria, si el flujo de entrada no estuviera bien formado y el plugin encontrara algún error, el fichero de salida (que también tendría el mismo nombre que el de entrada), estaría vacío. Si se diera este caso bastaría con validar el flujo de entrada para descubrir el error.

Por si se quisiera ampliar la funcionalidad del plug-in, se aporta el fichero *com.hpe.e4.converter.source\_1.0.0.201607202003.jar* que se podrá importar como proyecto especial (plug-in) siempre y cuando descarguemos previamente una versión especial de Eclipse (se puede descargar gratuitamente desde la página oficial) que permite el desarrollo que plugins para esta plataforma.

## 2. Validar los flujos

Si queremos validar los flujos generados por los traductores, o simplemente validar los flujos antes de traducirlos para comprobar que en efecto están bien formados, nos podemos ayudar de la herramienta *xmllint*.

Se puede descargar e instalar desde cualquier distribución/versión de linux/mac respectivamente sin problemas, no obstante si trabajásemos en un sistema Windows habría que seguir las indicaciones del siguiente enlace:

<https://stackoverflow.com/questions/19546854/installing-xmllint#21227833>.

Tras su descarga si quisiéramos validar un flujo en la notación HPE ejecutaríamos:

```
xmllint --noout --dtdvalid workflow.dtd flow.xml
```

Si su salida no muestra ningún error, el flujo es correcto, si no fuera el caso, nos mostraría el fallo:

```
flow.xml:11: parser error : Opening and ending tag mismatch:
Nname line 11 and Name
<Nname>Start new transaction</Name>
```

Del mismo modo si quisiéramos validar un flujo basado en la notación BPMN ejecutaríamos:

```
xmllint --noout --schema BPMN_HPE.xsd flow.bpm
```

Aunque en este caso si el flujo valida correctamente obtendremos el siguiente mensaje:

```
flow.bpm validates
```

Un mensaje muy típico cuando ocurre un error en la validación es el siguiente:

```
flow.bpm:20: element messasdsage: Schemas validity error : Element
' {http://www.omg.org/spec/BPMN/20100524/MODEL}messasdsage' :
This element is not expected. Expected is one of (
{http://www.omg.org/spec/BPMN/20100524/DI}BPMNDiagram,
{http://www.omg.org/spec/BPMN/20100524/MODEL}relationship ).
flow.bpm fails to validate
```

Para validar el xml podemos abrir dicho fichero en un editor como vim y ejecutar el siguiente comando (si tenemos instalado previamente *xmllint*)

```
!xmllint --noout --schema BPMN_HPE.xsd %
```