Otto-von-Guericke-Universität Magdeburg

Topic:

# Concept and Implementation of a Profile-Management System for Adaptive SAP Landscapes

## Diploma Thesis

Fakulty of Informatics
Workgroup Business Informatics

| | |
|---|---|
| Supervisors: | Prof. Dr.-Ing. habil. Reiner R. Dumke, |
| | Dr.-Ing Gamal Kassem |
| Proposed by: | Ronny Zimmermann(FIN/UCC) |
| Prepared by: | Diana Abu Gosh, Matr.-Nr.: 177213 |
| | Walther-Rathenau Strasse 55 |
| | 39104 Magdeburg |
| | dabughosh@hcc.uni-magdeburg.de |

Deadline: 01.07 2009

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

*"There is a time in the life of every problem when it is big enough to see, yet small enough to solve." Mike Levitt*

The University Competence Centre (UCC)[1] of Magdeburg provides more than 210 customers from Europe, Africa and the Middle East access to certain Systems Applications and Products in Data Processing (SAP) systems. Customers are primarily universities or other educational institutions that require SAP software for research and teaching. For the sufficient support of the multiple numbers of customers a lot of SAP systems are needed. UCC possesses approximately 120 systems, organized in a heterogeneous system landscape[2]. There are mainly WEB AS ABAP-systems [3] but there are JAVA-systems[4] as well. This diploma thesis will further on take into consideration the Advanced Business Application Programming (ABAP) systems, due to the fact that they are dominate over the JAVA systems in the viewd land-

[1] In June 2001, the SAP University Competence Center (UCC) was established by the project partners SAP AG, Hewlett Packard (HP), T-Systems CDS GmbH and Otto-von-Guericke Universität Magdeburg.

[2] System Landscapes are computing environments that consist of a number of hardware and software components working together with regard to installation, software updates, and demands on interfaces.

[3] Web AS ABAP technical systems are based on the same Basis Component (BC), which includes the ABAP interpreter, ABAP Dictionary, ABAP Workbench, and so on. A dedicated database, which stores most of the data of the BC and the application components, is associated with the system.

[4] Web AS Java systems consist of one or more instances and each instance is installed on a separate host. An instance itself consists of cluster nodes that can communicate with each other.

scape. Moreover, their parameterization is more explicitly structured in comparison to that of JAVA systems.

## 1.1 Motivation

The growing number of customers and their altering demands imply that the SAP landscape should possess a range of adaptability and flexibility which shouldn't just meet today's requirements. What is more, unexpected changes should be automated and supported by possibly less human interference. That is the reason why in the past 2 years the landscape has migrated to an adaptive infrastructure. The advantage of an adaptive infrastructure is that it makes it possible to flexibly react to unanticipated changes, for instance if a customer requires a service involving more computing resources than foreseen. In this case free computing resources should be found on another server so that the service is run on that server. Controlling the system landscape is done by a central SAP system which uses the Adaptive Computing Controller tool[5]. It provides a central point of control for assigning computing resources to run any service on any server at any time. Furthermore, it is an effective tool for optimizing the use of the available computing resources.

## 1.2 Problem Statement

This section provides the context for the diploma thesis by making a detailed explanation of the problem and discussing the importance of solving it. To be able to meet the increasingly growing customers' demands and to support the often changes in the business environment, SAP system landscapes need

[5] The Adaptive Computing Controller tool provides a central point of control for flexibly assigning computing resources. The tool provides also the capability for enterprise services computing.

to possess a certain range of adaptability. For this reason some SAP system landscapes are organized by a central system, the Adaptive Computing Controller. It provides a central point for distributing computing resources to run any service on any server at any time. Undoubtfully that brings numerous advantages in terms of flexibility and adaptability. For instance, if an SAP System requests more hardware resources, they can be quickly assigned to it at any time. However, one essential question remains without an answer; Namely, what changes take place in the SAP system's configuration so that complete advantage of the new resouces is provided? After the hardware sources are assigned they are not optimally assimilated by the SAP systems. The problem is that no change in the SAP system's configuration takes place so that the system can work effectively and effciently with the new resources. To be more exact, the system's profile parameters remain unchanged and therefore although hardware resources are set to the system, they are not fully used. Ideally, receiving new hardware resources should be followed by change of those profile parameters that are relevant to those resources. For example the parameters of the memory management system, such as Limit for Extended Memory (zttaroll extension), EM Quota for Dialog Processes (zttaroll extension dia), EM Quota for Non-Dialog Work Processes (zttaroll extension nondia) and so on should be given new values when more RAM is assigned to the SAP system to provide complete advantage of the new resources. The performance of the system measured in SAPS[6] should increase when more resources are available and vice versa. Consequently, major challenge of the diploma thesis is to determine a way for automated profile parameterization on resource change. A vital point to be considered is that systems' profile parameters are managed separately in

[6] The SAP Application Performance Standard (SAPS) is a hardware-independent unit that describes the performance of a system configuration in the SAP environment. It is derived from the Sales and Distribution (SD) Benchmark, where 100 SAPS is defined as 2,000 fully business processed order line items per hour. In technical terms, this throughput is achieved by processing 6,000 dialog steps (screen changes), 2,000 postings per hour in the SD benchmark, or 2,400 SAP transactions. In the SD benchmark, fully business processed means the full business process of an order line item: creating the order, creating a delivery note for the order, displaying the order, changing the delivery, posting a goods issue, listing orders, and creating an invoice.(1)

every system. In the context of an SAP system landscape with hundreds of systems, managing the single profile-parameters causes a big time complexity. The traditional approach has relied on human intervention to manage this complexity. However, this is now getting untenable. Therefore a central unit granting options for monitoring, controlling and adjusting system parameterization of all the systems would bring a considerable advantage. The role of monitoring is to collect the information necessary to determine whether the system is behaving in an acceptable way and to allow reliable decisions to be made. Control actions are then taken manually, in a centralised location, based on experience and human judgement. Automation and multiple changes therefore concentrate on the control aspect, combining with automated monitoring to close the loop, at least for the more common (routine) situations. Automated and centralized profile-management is needed to meet the necessary business requirements nowadays. Providing a central unit for profile management would bring numerous advantages in the whole landscape parameterization. As the global business environment is getting increasingly more dynamic, companies need excellent software performance to be able to respond to market opportunities or competitive threats. Agility is becoming a crucial requirement of enterprise systems. Providing agility requires a structured approach to omputing infrastructure and its management, which separates the logical functions offered by the infrastructure from the physical equipment which delivers them. Flexible and effcient use of computing resources in SAP system landscapes is vitally important so that the growing business needs can be satisfied. In this sense, assuring increased system performance through correct distribution and assimilation of computing resources in the system landscapes is of decisive importance. In view of the above discussion the stated problem can be related to the topic 'Virtualization'. Chapter 2 of the diploma thesis discusses the basics virtualization and some virtualization technologies and thereby sets the fundamentals of the diploma thesis.

## 1.3 Goals of the Proposed Work

Having the hardware resources optimally assigned, the SAP systems should operate effectively and efficiently on them. With the help of the system profile parameters it is ensured that the system configuration is set for the runtime environment. Therefore, their dynamical behavior is a key factor for achieving better adaptability and flexibility. In view of the above discussion, following is the brief outline of the goals set forth to this diploma thesis work.

- Investigating to what extent the profile parameters of the runtime environment can work with the adaptive landscape, in other words defining how adaptive they are.
- Classifying profile parameters into groups with respect to the physical resources of the landscape.
- Examining the present state of the art by completing a market research and literature analysis for existing possible solutions.
- The work proceeds by proposing a model for a profile-management system.

## 1.4 Thesis Organization

The remaining part of the thesis is organized as follows:

**Chapter 2**

This chapter starts setting the basis of the thesis. Since the stated problem is related to the topic virtualization, this chapter introduces the concepts of virtualization and virtualization technologies. Furthermore it discusses the advantages and disadvanatges of applying virtualization. Finally, it discusses the consequences of virtualization on applications' behaviour.

**Chapter 3**

This chapter introduces further fundamental definitions of the thesis. It introduces the concepts of SAP systems and SAP system landscapes. What is more, it discusses SAP system parameterization and its in achieving optimal performance and resource assimilation. The chapter makes classifications of profile parameters on different criteria, such as operation system dependency, relevance to the hardware resources of the system, etc.What is more, it tries to answer the question how dynamic an SAP can be and thus to fulfill the technical requirement for achieving the scientific goal. Initially, dynamics in terms of an SAP System is defined. Afterwards current concepts of dynamics in SAP systems are discussed. Furthermore, dynamics of the system's state is described with respect to the system profile parameters. Then, this chapter tries to answer how the current concepts of dynamics can be improved so that the system-parameters behave adaptively with respect to the landscape. To conlude the two fundamental chapters, a brief discussion over virtualized SAP system landscapes is presented.

**Chapter 4**

Based on the previous two fundamental chapters, this chapter gives an outline of several possible approaches to the stated problem and chooses the most optimal one. Consequently, a concept for a profile management unit is proposed in terms of its functional and non-functional requirements. Additionally, the integration with the Adaptive Computing Controller is concerned.

**Chapter 5**

This chapter tries to accomplish the scientific goal, by implementing the concept of a centralized profile-management unit with logic which is developed and proved, and its implementation is described in terms of its design and certain key algorithms.

**Chapter 6**

This chapter will make a market research and literature analysis of already existing solutions for the stated problem.

**Chapter 7**

This chapter brings the thesis to an end and evaluates potential applications of the profile management unit as well as the potential fields of future work.

# Chapter 2

# Virtualization

*"As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality." Albert Einstein*

Virtualization is a proven software technology that is rapidly transforming the IT landscape and fundamentally changing the way in which people compute. Today's powerful computer hardware has been designed to run a single operating system and a single application. This leaves most machines vastly underutilized. Virtualization allows running multiple virtual machines on a single physical machine, sharing the resources of that single computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer. This chapter of the diploma thesis touches upon the basics of virtualization and discusses its advantages and disadvantages. To be able to define whether or not virtualization techniques are the correct approach to the stated problem, a good working of some fundamental virtualization technologies is essential.

**Section 2.1**

This section makes an introduction to virtualization, gives a definition of the concept and discusses some common arts of virtualization and virtualization technologies.

**Section 2.2**

This section gives an overview of the advantages and disadvantages of applying virtualization.

**Section 2.3**

This section discusses discusses the consequences of virtualization on applications' performance and behaviour.

## 2.1 Virtualization

### 2.1.1 Definition

Virtualization is a vast concept that refers to the abstraction of computing resources. It is defined by Chris Wolf and Erik M. Halter as *"the act of abstracting the physical boundaries of a technology"(2)* . A loose but perhaps more comprehensive definition determines virtualization as

*"a framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others"(3).*

Roughly speaking, it can be distinguished between several kinds of virtualization technologies. These include but are not limited to platform virtualization, resource virtualization and application virtualization.

### 2.1.2 Virtualization Technologies

Platform virtualization is performed by a control program (host software) which creates a virtual machine for potential 'guest software'. It makes the physical characteristics of computing platform invisible for the users and shows them an abstract platform.The guest can be used as if it was installed as a stand-alone hardware platform. Typically, many such virtual machines can be simulated on a single physical machine, their number is limited by the hosts hardware resources. In the context of platform virtualization, there are several commonly used techniques: full virtualization, hardware-assisted virtualization, partial virtualization, paravirtualization, operating system level virtualization.

- **Full Virtualization**

  This virtualization technique is used to implement a Virtual Machine [1] environment that provides a full simulation of the underlying hardware. The result is a system in which all software capable of execution on the raw hardware can be run in the virtual machine. Full virtualization is only possible when the correct combination of hardware and software elements is available. A major challenge for full virtualization is the interception and simulation of certain operations, such as I/O instructions. The results of every operation performed within a given virtual machine must be kept within that virtual machine. Virtual operations cannot be allowed to change the state of any other virtual machine, the control program, or the hardware. Some machine instructions can be executed directly by the hardware, since their effects are entirely contained within the elements managed by the control program, such as memory locations and arithmetic registers. But other instructions that would "pierce" the virtual machine cannot be allowed to execute

[1] A virtual machine is of an operating system with its applications running in an isolated partition within the computer. It makes it possible for different operating systems to run on the same computer at the same without interfering each other.

directly; they must instead be trapped and simulated. Such instructions either access or affect state information that is outside the virtual machine. Full virtualization has proven highly successful for sharing a computer system among multiple users, isolating users from each other (and from the control program) and emulating new hardware to achieve improved reliability, security and productivity.(4)

- **Hardware-Assisted Virtualization**

  This approach makes efficient full virtualization possible with the help of the hardware capabilities of the host processor. Hardware-assisted virtualization reduces the maintenance overhead of paravirtualization as it restricts (ideally, eliminates) the amount of changes needed in the guest operating system. It is also considerably easier to obtain better performance. That approach, however, involves many VM traps, and thus high CPU overheads; this limits scalability and the efficiency of server consolidation. (4)

- **Paravirtualization**

  Paravirtualization refers to modifying the OS kernel to replace non-virtualizable instructions with hyper calls that communicate directly with the virtualization layer hyper visor. Paravirtualization is different from full virtualization, where the unmodified OS cannot recognize being virtualized and sensitive OS calls are trapped using binary translation. The advantage of paravirtualization is in lower virtualization overhead, but the performance advantage of paravirtualization over full virtualization can vary strongly depending on the workload. As paravirtualization cannot support unmodified operating systems (e.g. Windows 2000/XP), its compatibility and portability is poor. (4)

- **Operating System-Level Virtualization (Containers Virtualization)**

  This is a server virtualization method where the kernel of an operating system allows for more than one isolated user-space instances, instead of just one. These instances may seem as a real server, from the point of view of its owner. In addition to isolation mechanisms, the kernel often provides resource management features to limit the influence of one container's activities on the other containers. The advantage of that technique lies undoubtfully in the little or no overhead, however the lack of flexibility in comparison to other techniques is significant. The three techniques differ in complexity of implementation, breadth of OS support, performance in comparison with standalone server, and level of access to common resources. For example, VMs have wider scope of usage, but poor performance. Para-VMs have better performance, but can support fewer OSs because of need to modify the original OSs. Virtualization on the OS level provides the best performance and scalability compared to other approaches. Performance difference of such systems can be as low as comparing with that of a standalone server. Virtual Environments are usually also much simpler to administer as all of them can be accessed and administered from the host system. Generally, such systems are the best choice for server consolidation of same OS workloads.(4)

Resource Virtualization is the virtualization of definite system resources, such as storage volumes, name spaces, and network resources. This kind of virtualization is often used to address computer problems in the sphere of security, performance, optimization, system administration and reliability. Resource virtualization shows potential as well in the sphere of resource management in distributed network environments (7) . In the context of resource virtualization, worth to be mentioned are memory virtualization, storage virtualization and network virtualization.

- **Memory Virtualization**

  Memory Virtualization introduces a way for decoupling memory from the processor to provide a shared, distributed or networked function. This is not more addressable memory but virtualized memory shared between multiple machines. Memory Virtualization is focused on application performance and has a direct touch point with end users. It finds broad application across IT infrastructures, for example in extending memory beyond a physical servers capacity, implementing shared memory for clustered or grid computing environments, enabling Cloud Computing and Real-Time Infrastructure in the enterprise data centers. The challenge facing Memory Virtualization is to overcome the many and sometimes unrecognized workarounds to not having access to sufficient memory. To begin with, extending memory is considered. An applications working data set is often larger than the available physical memory in the server. Using Memory Virtualization, the entire working data set can be loaded into memory for the processor to access directly, without going to disk. The benefits to the end user and IT organization of extending memory in this scenario are tremendous. Memory Virtualization implementations consistently show significant performance improvement to the end user. Direct cost savings through server consolidation, reduced over-provisioning, deferring server upgrades, and increased utilization are immediately achieved through Memory Virtualization. Next to discuss is implementing shared memory. In the shared resource model, applications or services use only as much memory as required. Memory is distributed and shared as an available network resource. Large enough amounts of memory can be readily available to any application. In this case, companies will now be able to harness large amounts of available memory in their expensive high-performance servers. Last but not least to discuss is enabling Cloud Computing and other next generation data center initiatives. The choice to use a next generation data center or a Cloud Provider is an important strategic decision for every enterprise. Memory Virtualization plays an enabling role in both environments. Memory Virtualization is critical to implementing agile services for these infrastructures. By offering memory as a pooled resource in the data center, Memory Virtualization delivers dramatic cost savings from reduced power and cooling. Real-time Infrastructure is available at 30 percent less cost in a pooled resource architecture versus over a tiered architecture. Memory Virtualization is an enabler of Cloud Computing in this use case. Memory Virtualization enables different infrastructure Cloud Models to be available as different services with fungible resources. Virtual memory provides the OS and application memory for this distributed operating system. To conclude, Memory Virtualization can dramatically boost performance at significantly lower cost while delivering service levels that will transform business. Making memory a truly shared, network resource has broad and deep implications across the spectrum of enterprise applications, clustered computing and data center operations. (6)

- **Storage Virtualization**

  Storage virtualization is the technique of abstracting logical storage from physical storage. This virtualization technique helps storage administrators to perform the tasks of backup, archiving, and recovery more easily, and in less time, by disguising the actual complexity of the storage area network. Virtualization of storage helps to achieve location independence by abstracting the physical location of the data. The virtualization system presents to the user a logical space for data storage and itself handles the process of mapping it to the actual physical location. Virtualization of storage helps achieve location independence by abstracting the physical location of the data. The virtualization system presents to the user a logical space for data storage and manages the process of mapping it to the actual physical location.

- **Network Virtualization**

  Network virtualization is a virtualization technique that combines hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network. Network virtualization involves platform virtualization, frequently in combination with resource virtualization. Network virtualization is categorized as either external, combining many networks, or parts of networks, into a virtual unit, or internal, providing network-like functionality to the software containers on a single system. Whether virtualization is internal or external depends on the implementation provided by vendors that support the technology. The goal of network virtualization is to optimize network speed, reliability, flexibility, scalability, and security. Network virtualization is said to be especially effective in networks that experience sudden, large, and unforeseen surges in usage.

Application Virtualization is the generalized term for software technologies that improve portability and compatability by encapsulating them through the OS System. The difference between Operation system virtualization and application virtualization consists in the fact that in Operation System virtualization the whole operation system is virtualized, while in application virtualization only certain applications are virtualized. Application virtualization can be either limited or full. Limited virtualization is used lately in some operating systems to support the virtualization of legacy applications. Full virtualization requires a virtualization layer [2] Full application virtualization consists of the illusion of accessing the physical resource by the application performing the file operations.

- **Application Streaming**

  Application streaming offers a server centric architecture which enables the deployment and management of applications. Only definite components of a computer program have to be available at any instance for the end user to perform a particular function. This means that a program does not have to be fully installed on a client computer. Rather parts of it can be delivered over a low bandwidth network when they are required. This technique provides multiple benefits such as increased performance and rapid development.

[2] An abstract layer that enables the creation of multiple virtual machines. The virtual layer is between the physical layer and the guest operating system. Thus it protects software running on a VM from hardware-changes.(8)

- **Desktop Virtualization**

  Desktop virtualization is the decoupling of a users physical machine from the desktop and working software. Usually desktop virtualization products emulate the PC hardware environment of the client and run a virtual machine alongside the existing operating system located on the local machine or delivered to a thin client from a data center server. This virtualization technique provides numerous advantages such as low downtimes, reduction of the cost of new application deployment and multiple desktops on demand.(9)

Important techniques to be mentioned in the sphere of application virtualization are application streaming and desktop virtualization.

## 2.2 Advantages and Disadvantages of Virtualization

To begin with, virtual machines can be used to consolidate the workloads of several under-utilized servers to fewer machines, perhaps a single machine (server consolidation). Related benefits (perceived or real, but often cited by vendors) are savings on hardware, environmental costs, management, and administration of the server infrastructure. Furthermore, the need to run legacy applications is served well by virtual machines. A legacy application might simply not run on newer hardware and/or operating systems. Even if it does, if may under-utilize the server, so as above, it makes sense to consolidate several applications. This may be difficult without virtualization as such applications are usually not written to co-exist within a single execution environment. What is more, virtual machines can be used to provide secure, isolated sandboxes for running untrusted applications. Virtualization is an important concept in building secure computing platforms. Important to consider is that virtual machines can be used to create operating systems, or execution environments with resource limits, and given the right schedulers, resource guarantees. Virtual machines can provide the illusion of hardware, or hardware configuration that is not available. Virtualization can also be used to simulate networks of independent computers. A considerable advantage of virtual machines is that they can be used to run multiple operating systems simultaneously: different versions, or even entirely different systems, which can be on hot standby. Some systems of that kind may be hard or impossible to run on newer real hardware. Moreover, virtual machines make software easier to migrate, thus aiding application and system mobility. Virtualization provides tools for research and academic experiments. Since they provide isolation, they are safer to work with. They encapsulate the entire state of a running system: you can save the state, examine it, modify it, reload it, and so on. The state also provides an abstraction of the workload being run. Additionally, virtualization can enable existing operating systems to run on shared memory multiprocessors. Virtual machines can be used to create arbitrary test scenarios, and can lead to some very imaginative, effective quality assurance. Virtualization can be used to introduce new features in existing operating systems without too much work. Virtualization on commodity hardware has been popular in co-located hosting.

Many of the above benefits make such hosting secure, cost-effective, and appealing in general. (5) Nevertheless, virtualization has negative sides as well. Companies need to be aware of many significant issues from both a broad business perspective, and specifically regarding IT management. These issues affect their ability to implement virtualization technologies, as well as the advantages they can gain. The remaining part of this section discusses some of the disadvantages of virtualization. Firstly, it considers the sphere of cost accounting. Cost accounting and licensing become much harder in a virtual environment. Enterprises need to measure highly dynamic resource usage, not just acquisition costs. Some tools exist that can help, but they are rare and incomplete. One solution for licensing costs is to look at open source opportunities to obviate licensing issues, but this is unlikely to resolve all the pertinent issues. Another significant disadvantege is in the area of the human issues. Enterprises should not underestimate the potential for human issues to affect their virtualization plans adversely. Virtualization requires a new set of skills and methodologies within both IT and the end-user community - including new and creative thinking, not just new training and skills. Departments and business owners may baulk at sharing resources. Enterprises should be prepared to deal with these human issues early and often in their virtualization projects. Important to be considered are also the support issues. Compatibility and support requirements may also preclude running specific virtual workloads together on a single system. Enterprises need to be aware of the hardware and software requirements from both their virtualization vendor and their other software providers, and be prepared to meet them before deploying virtualization technologies. A critical issue about virtualization is the increasing degree of complexity. irtualization tends to make the environment more complex by adding a new layer of software that must be maintained, including performance and availability monitoring, upgrades, patches, etc. Add to this the increased complexity of diagnosing problems, additional service desk calls due to user complexity, and the complexity of managing virtual images, and virtualization complexity can turn out to be a difficulty for an unprepared enterprise. The issue of security should also not be estimated. While virtualization can have many worthwhile security benefits, security also becomes more of a management issue.

There are more systems to secure, more points of entry, more holes to patch, and more interconnection points. Virtualization also introduces the risk of attack from entirely new forms of malware - root kit infections, low-level hypervisor attacks, deployment of malicious virtual systems, and malware that actually detects a virtual environment and modifies itself accordingly. Security becomes not necessarily more fragile, but certainly more critical. Other potential problems include diffculty in using existing monitoring and management tools; new and significant hardware issues such as increased uptime requirements and more difficult capacity measurement and planning; potential bandwidth issues, especially for end-user virtualization; support, integration, and compatibility of different operating systems; dealing with misleading and connecting vendor claims; and the potentially unexpected costs of additional hardware and software.

However, each virtualization deployment is different and a single use case can scarcely achieve all of these benefits. In fact, in some cases, an enterprise may deploy virtualization technologies and achieve none of these benefits. The key is in obtaining a full understanding of the environment, the need, and the goals, and matching the right virtualization solution to the business.

## 2.3 Applications' Behaviour under Virtualization

Virtualization is extremely suitable when it is used for applications that are designed for either small to moderate sized usages. However, it should never be used in scenarios where multiple servers must be clustered together so that performance requirements can be achieved, since the added overhead and the complexity would lower the level of performance. Many experts in the virtualization industry spend too much time thinking about a higher than average number of CPU utilization, and they do not spend enough time considering the responsiveness of the application.

To be able to figure out how applications' behaviour under virtualization, a comparison should be done between the behaviour in the native and the virtualized environment. Applications running in native mode have the whole machine to utilize. The native OS handles the resources and scheduling for the application software stack. Virtualization is designed to 'transparently' present generalized hardware to a Guest OS thru a Virtual Machine Monitor. Thus the general guidelines for the validation of application software can also be applied to virtual mode. First, in a virtualized mode, there is a layer of abstraction of the underlying platform hardware by VMM or Hypervisor. The VMM presents an abstraction of the underlying hardware to the Guest

OS. The VMM manages resource-allocations and scheduling for each of the Virtual Machines hosted on top of the native machine. This additional layer may remove special hardware features or introduce extra resource contention among multiple virtual machines being hosted. These two may introduce unexpected test failures into the validation.

Furthermore, in virtual environments, there are unique configurations that are either not needed or not possible in native environments. For example, the number of VMs and their resource allocations must be predefined. Some VMMs allow direct assignments of devices to particular VMs such as using Intel Virtualization Technology (Intel VT) for Directed I/O (Intel VT-d) for added performance. These need to be understood in the test planning. There are several challenges in validating applications in virtual environments. Some considerations are:

- Are there any 'special features' of the hardware used that may not react the same way when virtualized? For instance, some CPUID leaves may not pass unmodified through the VMM. Some I/O features may demonstrate reduced performance, etc.
- Are there any limits imposed by current generation VMM which might impact your application? For instance, current MSFT Viridian has a 1 vCPU and 3.6Gb memory limit per GuestOS.
- What is the application architecture? Multi-tier? Will the entire application stack be virtualized and consolidated onto the same platform?
- How does the application software behave in a multi VM heterogeneous configuration? Does having totally different VM/application-software running on same machine effect our application of interest?
- What are the minimal resource requirements for application software when run in a GuestOS (and also along with other Guest OS on same machine)? How much network bandwidth/latency are required for application to perform satisfactorily

- Does the application software require guarantee of minimal/maximum resources? How many shares of CPU, Memory, Network and disk/IO resources does it require?

- How does the application software behave when resources are overcommitted (eg: have only 2 physical CPUs, but 4 VMs are hosted; same for other resources like memory, network and disk). Generally, overcommitting memory is the most likely of these to lead to catastrophic performance issues. (11)

To conclude, at least for the near future, the biggest difference of a virtualized hardware platform from the native hardware platform will remain their difference in performance (at least on certain types of workloads). For the near future, customers will always compare virtualized environments to native and will always expect leading performance on their current hardware.

# Chapter 3

# SAP System Landscapes

*"Get your facts first, and then you can distort them as much as you please."*
*Mark Twain*

To begin with, this chapter of the diploma thesis will focus on the fundamentals of SAP Landscapes and SAP Systems. Further on it will discuss system profiles, their different types and roles in an SAP system.

**Section 3.1**

This section gives a brief outline of the term SAP System and describes the different kinds of SAP systems.

**Section 3.2**

This section concentrates on the fundamentals of SAP system landscapes, SAP System Landscape Directory (SLD)[1] and describes what types of technical systems there are in the SLD.

**Section 3.3**

This section focuses on the steps that already taken from SAP AG in the sphere of virtualization.

[1] The System Landscape Directory of SAP NetWeaver (SLD) serves as a central information repository for the system landscape.

**Section 3.4**

This brief section tires to make a generalization of the stated problem, based on the fundamentals of the thesis.

## 3.1 SAP Systems

In SAP systems the term 'instance' is often used as a synonym for 'application server' or 'server'. SAP instances define a group of resources like memory and work-processes, maintaining an application server in a client/server environment. An SAP system consists of one or more instances. Each instance has its own directories on the server where it runs, own communication entries in the host and its own SAP profiles.

One can distinguish between the following types of SAP systems:

1. Web AS ABAP

   Web AS ABAP technical systems, ranging from any SAP system to an APO[2] server or CRM server, are based on the same Basis Component (BC), which includes the ABAP interpreter, ABAP Dictionary, ABAP Workbench, and so on. A dedicated database, which stores most of the data of the BC and the application components, is associated with the system.

2. Web AS Java

   Web AS Java systems consist of one or more instances and each instance is installed on a separate host. An instance itself consists of cluster nodes that can communicate with each other.

3. Standalone Java

[2] Advanced Planner and Optimizer

Standalone Java technical systems are standalone Java applications that are installed in a computer system from an administration perspective. They can also be installed as an operating system service or as a daemon.

4. Third-Party

   Third-party technical systems contain third-party software components and products.

5. PI

   PI technical systems are the infrastructure systems (such as integration servers and adapter engines) of PI (14).

## 3.2 SAP System Landscapes

System Landscapes are computing environments consisting of a number of hardware and software components that depend on each other with regard to installation, software updates, and demands on interfaces. Roughly speaking, the system landscape can be defined as the consent of the following the servers:

1. Development,
2. Quality Assurance,
3. Production.

Requests are always made in the sequence Development, Quality Assurance and Production.

On the development server the consultants do customization according to the company's requirements. On the quality server customization is tested by the core team members.On the production server the live data of the company is recorded.

These three are landscape of any Company. They organized their office in these three way. Developer develop their program in Development server and then transport it to test server. In testing server tester check/test the program and then transport it to Production Server. Later it is deployed to client from production server. (12)

The administration of the system landscapes is supported by the SAP SLD. (13)

Roughly speaking, in the SLD there are five basic kinds of technical systems

1. Web AS ABAP,
2. Web AS Java,
3. Standalone Java,
4. Third-Party,
5. Process Integration (PI)[3].

## 3.3 Virtualized SAP System Landscapes

For the reason that SAP Applications belong to the most 'critical' in an enterprise, it took SAP more time than expected to certify some virtualization companies. Finally, however VMWare5 was SAP certified and customers could run their SAP Applications on VMWare ESX Server. To stimulate that, SAP AG and VMWare established a global partnership. Compa-

[3] SAP Process Integration is a software package that realizes data exchange between SAP and other systems. As a part of the SAP Netweaver it is a middleware component that serves as Enterprise Application Integration.

nies that were using SAP applications for their business processes could use VMWare ESX Server get full SAP support for the entire lifecycle of the SAP Software. In this way the VMWare Infrastructure supports SAP solutions under Windows and Linux in production environments. What is more, SAP has founded an 'Enterprise Virtualization Community' together with popular producers from the virtualization sphere.The goal of this union is the development of new, business process oriented ways of virtualization, that helps customers to use their resources in a better way. The producers from the virtualization sphere were presented by VMWare and the Virtualization Competency Center. In this way SAP made a step forward to the virtual world and could offer customers same flexibility in services and optimal use of resources as they had in other organization applications.

The following example shows how virtualization can help SAP systems to respond to increased requirements for breakdown security and flexibility 3.1. Challenge: Modern business processes require as big as possible flexibility and need for that highly available and breakdown secure infrastructures. Solution: The virtualization solution VMWare Infrastructure 3 decouples systems from the from the hardware and integrates on the lowest level functions for breakdown security and efficient resource management.

- VMWare Infrastructure 3 offers services for automatic failover between hosts and supported redundant LAN- and memory network conditions for highest availability.
- Expensive Server hardware is optimally used by the SAP applications. Capacities of CPU, Main memory, network or disc space can be flexibly divided.
- VMWare Virtual Center manages the whole infrastructure centrally through an uniform surface. This provides a central overview with the advantage of quickly recognizing the point of failure.
- Products such as VMWare consolidated Backup lead a central data securing of the whole landscape directly through a Storage Area Network

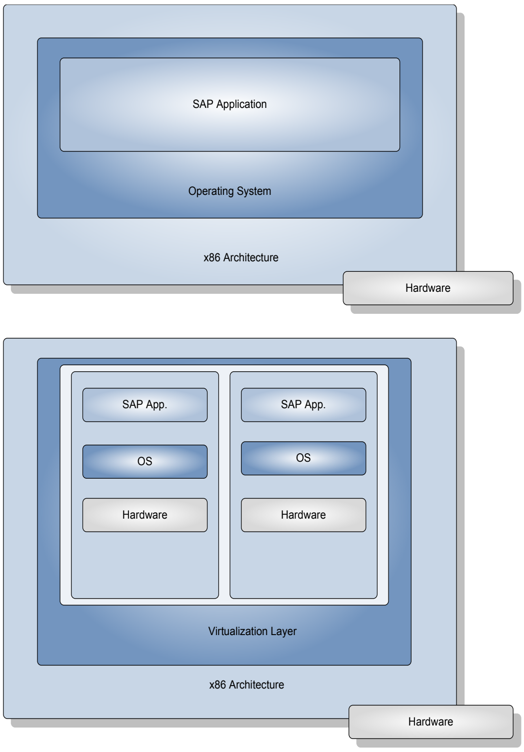


Figure 3.1: SAP Architecture without and with Virtualization

and enable a fast Disaster Recovery. To conclude, without virtualization software and hardware are strongly coupled to each other. The Virtualization layer helps to oercome the disadvantages of that coupling through virtual machines that run independently of the hardware behind.

To conclude, without virtualization software and hardware are strongly coupled to each other. The Virtualization layer helps to oercome the disadvantages of that coupling through virtual machines that run independently of the hardware behind.

## 3.4 Problem Generalization

Before computing resources in a data center environment are virtualized, important are all the critical junctures of the network topology, the current environment and if there are applications that could be virtualized. Essential is also where the users are coming from and, of course, what business goals, objectives, and Service Level Agreements [4] there are. When all these issues are clarified, virtualization can take place.

In this sense, the scenario of the stated problem 1.2 can be brought to the essentials of maximizing applications' performance in virtualized data center environments. Very important to consider at this stage is the architecture of the application. Can this application be deployed in a manner that it can be virtualized? Does it support clustering or are there tools that help it support clustering so that each application instance recognizes state? If that is the case, that application is an appropriate candidate for virtualization within the broader context of the application delivery network framework. Ultimately,the underlying application infrastructure should be investigated to determine what can be virtualized.

Similarly to the SAP applications' profile files, other applications, for instance, also contain some configuration information that can reflect on their performance and optimal resource assimilation. When applying any virtualization technique this configuration should be properly adjusted, to make sure it does not any restrictions that can cause performance problems.

For example, some Oracle [5] applications like the Oracle E-Business Suite[6] possess configuration files. In these configiration files there are certain paramaters indicationg timeouts, et. that are responsible for the applications' performance. In case of applying any virtualization technologies these con-

[4] http://www.service-level-agreement.net/
[5] http://www.oracle.com/index.html
[6] http://www.oracle.com/lang/de/applications/index.html

figuration files and their paramaters should be taken in consideration in order to prevent performance loss.

To infer, it is essential to consider applications' configuration when applying virtualization technologies, because sometimes unforeseen existing limitations can hamper the desired performance increase.

# Chapter 4

# Profile Parameters and their Classification

This chapter of the thesis will touch upon the profile parameters. It presents a classification of the profile-parameters on some vital characteristics, such as relevance to hardware resources, work-processes or operation-system dependencies.

**Section 3.1**

This section explains the basic types of profile files along with their purposes and function.

**Section 3.2**

This section concentrates on the profile parameters of the memory management system.What is more, it gives a concise explanation of the memory types and memory management.

**Section 3.3**

This section presents characteristics of some other groups of profile parameters, for instance dynamic and non-dynamic, operation system specific, etc.

Profile-parameters define the runtime configuration of the instances, the available services and services that can be found. For this reason profile-parameters are at the heart of the concept central management and dynamics in SAP Landscapes.

## 4.1 Profile Files

SAP profiles are operating-system files that contain instance configuration information. Individual configuration parameters can be customized to the requirements of each instance. Profile parameters configure the following:

- The runtime environment of the instance (resources such as main memory size, shared memory, roll size); The memory types and memory management system are explained in detail in section 3.1.2.
- Which services the instance itself provides (work processes)
- Where other services can be found (database host).

After installing an SAP instance, a start profile and an instance profile are automatically generated. If it is the first instance of an SAP system, the system also creates a default profile. Otherwise, the existing default profile is just updated. These three profiles contain the parameters that determine the operating system resource usage of the instance, for example, the number of work processes, important data directories, and the name of the database host and the buffer sizes (15). See figure 4.1.

### 4.1.1 Start Profiles

When an SAP instance starts on a host, the start profile defines which SAP services are started (message server, dialog, gateway or enqueue process, for

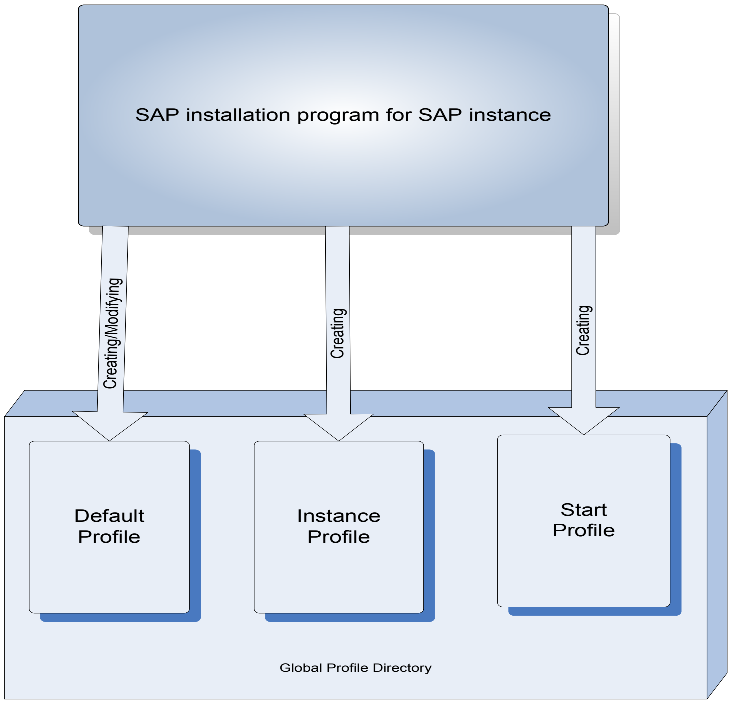


Figure 4.1: Profile Files

example). The startsap [1] program assures starting these service processes. The processes that can be started include:

- Application server
- Message server
- Systems Network Architecture(SNA) Gateway
- System log send demon
- System log receive demon

[1] The startsap program is responsible for starting service processes, it makes a call to a start profile to begin the startup process.

| Parameter Name | Parameter Value |
|---|---|
| SAPLOCALHOST | a18z |
| SAPLOCALHOSTFULL | a18z.hcc.uni-magdeburg.de |
| SAPSYSTEMNAME | A18 |
| INSTANCE NAME | DVEBMGS18 |
| ... | ... |
| Execute 00 | immediate (DIR CT RUN)sapcpe(FT EXE) pf=( PF) |
| ... | ... |

Table 4.1: Part of a Start Profile

Start profile files contain some common profile parameters, for example the name of the system, the number of the instance and its name. In addition to them, start profiles permit some other parameters, for instance 'Execute_n', 'Start. Beside the common profile parameters, such as the name of the SAP System, instance number and name of the SAP instance, the only parameter names that are permitted in a start profile are:

- Execute_n

  This parameter is used to start operating system commands which prepare the SAP System start. For example, this parameter can be used to start the SAP-related database or to set up links to executables files.

- Start_Program_n To start an SAP instance, for example, on an application server.

- Stop_Program_n To start an operating system command or SAP program after the SAP instance was stopped. For example, the keeping or removing of shared memory areas that were used by the SAP System (16).

Table 4.1 shows an example of a start profile.

### 4.1.2 Default Profiles

To assign the same parameter value for all application servers (such as the name of the database host, or the host on which the message server is running), it should be entered it in the default profile. Roughly speaking, there are no limitations about parameters that can be put in the default profile; any parameter can be listed here (17).

### 4.1.3 Instance Profiles

Instance profiles provide an application server with additional configuration parameters and therefore supplement the settings in the default profile. Usually, these are parameter settings that are adapted to the instance according to the required resources and applications. They also define the available instance resources (main memory, shared memory, roll memory, and so on), and how to allocate memory to the SAP application buffers. Section 3.1.2 will make a detailed discussion of these issues (18).

To start application servers on several computers using identical parameter settings, the same instance profile can be used. It is generally not necessary for each application server to have its own instance profile. Instance profiles are also called system profiles.

At system start the start profile is read by the start program and the services that should be started on the corresponding instance are planned. When the system starts, it checks the parameter settings in the instance profile. If a certain parameter is not found there, the system searches the default profile. If no corresponding entry can be found, the parameter will not be considered at all. The default profile contains the parameter settings that are supposed to be valid for all application servers.

When an SAP startup process wants to read a profile parameter, it checks:

- Whether the parameter is contained in the instance profile
- The default profile, if the parameter cannot be found in the instance profile.

If neither profile contains the parameter, the default value is taken out of the startup program coding.

Individual setup parameters can be customized to the requirements of each instance.

The individual profile parameters allow you to configure:

- The runtime environment of the instance (resources such as main memory size, shared memory, roll size); See 4.2.1,
- Which services are available for the instance, i.e. the work processes
- Where other services can be found, i.e.database host

## 4.2 Profile Parameters of the Memory Management System

To be able to classify profile parameters with respect to the physical resources of the system a good working knowledge of the SAP Memory types and Memory Management System is essential.

### 4.2.1 SAP Memory types and Memory Management System

- **Virtual Memory**

  All operating systems (supported by SAP) support virtual memory technology. A process allocates virtual memory using logical (virtual) addresses. Each process has its own virtual address space. Virtual memory is fully independent of the physical main memory.

- **Address space**

  On 32 bit platforms a virtual address can have values up to 4GB. As areas of the virtual address space are reserved, this leaves about 2GB available on most platforms. This is not a problem for large SAP systems.

- **Memory allocation**

  Allocating memory for a process consists of the following steps:

  1. Reservation of a segment in the physical memory.
  2. Linking the physical memory segment to the virtual address space virtual address space of the process, which means reserving a segment of the same size in the virtual address space and mapping the virtual addresses to physical addresses.

- **Local process memory**

  The operating system differentiates between local process memory and shared memory. For local process memory the operating system keeps both allocation steps transparent. Using an API virtual memory only is requested; the operating system does the other tasks, such as reserving physical memory, loading and unloading virtual memory into and out of the main memory.

- **Shared Memory**

  If several processes are to access the same memory area, both allocation steps are not transparent.

  One object is created that represents the physical memory and can be used by various processes. The processes can map the object fully or partially into the address space. The way this is done varies from platform to platform. Memory mapped files, unnamed mapped files, and shared memory are used.

- **Roll Area**

  The roll area is an area of memory located in the heap of virtual address space of the work process. This kind of memory area has 2 segments. The roll area is always the first memory that receives a work process. Only afterwards can extended memory be requested. The work process gets initially the first segment, set by the profile parameter zttaroll_first. When the first memory segment is depleted, additional memory is assigned to the work-process. SAP extended memory (EM) is at the heart of the SAP management system. When roll area is depleted, the work process turns to its virtual address space for reserved part of EM. The size of extended memory is determined by the profile parameter eminitial_size. The EM is implemented according to the operating system. For example, under Windows additional EM is either assigned dynamically as needed or a definite is set. Private memory is assigned to a dialog-work process when both the roll area and the extended memory are used up. Non-dialog (background, update, enqueue and spool) work-processes obtain heap memory after depleting the roll-area. See figure 4.2.

The memory management system allocates memory to the SAP work process, so it can execute desired requests, such as ABAP programs. Memory can be assigned from the following areas: roll area, extended memory, and heap (private) memory. Work processes are assigned memory types in a different order. This depends on the operating system and whether the work process is dialog or non-dialog. In this way optimal use of the characteristics of the individual memory types is enabled.

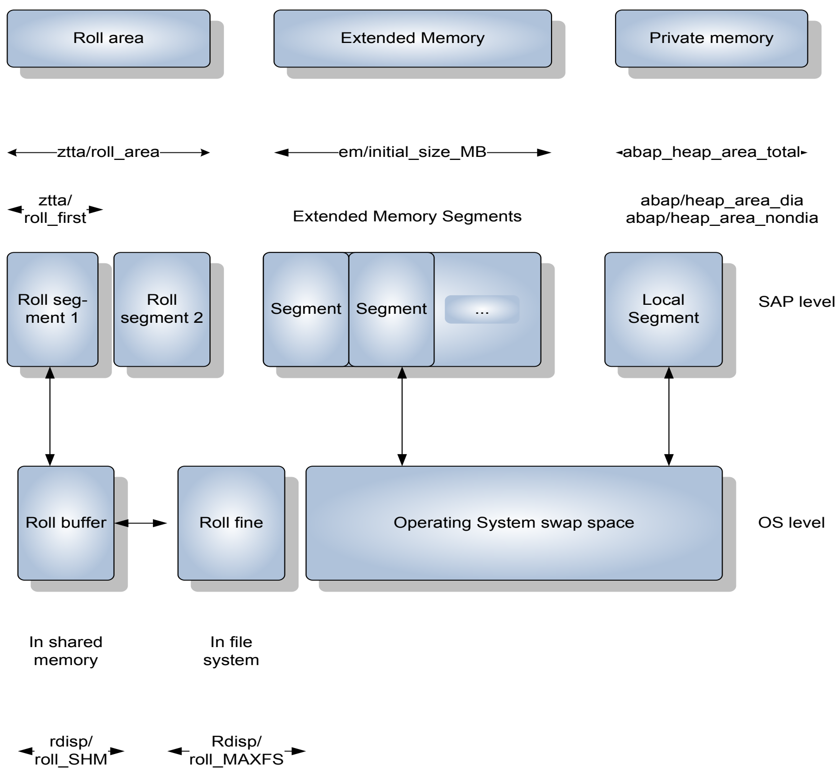


Figure 4.2: Memory Managament

The memory management system is adjusted to the resources and requirements by profile parameters. Through profile parameters the management system can be set in an optimal way, so that maximal profit is achieved. Furthermore, profile parameters control and restrict the behavior of the memory management system according to the operating system and the available hardware resources (20).

| Parameter Name | Brief Description |
|---|---|
| abapheaplimit | Work Process Restart |

Table 4.2: Parameters for Controlling Memory Management

### 4.2.2 Classification of the Profile Parameters of the Memory Management

These parameters govern how the memory management system behaves and restrict it so that it does not exceed operating system or hardware limitations. Certain parameters may restrict individual work processes, and quotas may limit resources used by all work processes, for example, the extended memory. There are also parameters that are used for configuring the memory management statistics.

Generally speaking, the parameters of the memory management system can be separated in several groups:

- Controlling Memory Management

  This group consists of one parameter. Its value specifies the memory amount in bytes. When this amount is exceeded, a work process is restarted after executing a dialog step. See table 4.2.

- Memory Management Resources for one Work-Process

  The profile parameters in this group refer to one user context. They define threshold values for EM, roll area and heap memory for dialog and non-dialog work processes. See table 4.3.

- Memory Management Limitations

  The profile parameters in this group specify maximal values and limitations of EM, paging buffer and are used to restrict space usage of an SAP server to a specific amount. See 4.4.

| Parameter Name | Brief Description |
|---|---|
| ztta/roll_extension | Limit for Extended Memory |
| ztta/roll_extension | Extended Memory Limit |
| ztta/roll_extension_dia | EM Quota for Dialog Processes |
| ztta/roll_extension_nondia | EM Quota for Non-Dialog Work Processes |
| abap/heap_area_dia | Heap Memory Limit for Dialog Work Processes |
| em/initial_size_MB | Extended Memory Pool Size |
| em/global_area_MB | Size of the Extended Global Memory |
| rdisp/ROLL_SHM | Roll Buffer Size |
| rdisp/ROLL_MAXFS | Maximum Roll File Size |
| rdisp/PG_SHM | Paging Buffer Size |
| rdisp/PG_MAXFS | Maximum Size of the SAP Paging File |
| ztta/roll_first | Initial Allocation Size from the Roll Area |

Table 4.3: Memory Management Resources for one Work Process

- Memory Management Statistics

  The profile parameters of this group are used to detect if the size of user context exceeds some threshold and the intervals at which this happens. This information is collected for statistical purposes. See table 4.5

- Release of SAP Memory for the Operating System

  The values of the parameters in this group reflect thresholds of memory amount, period of the time showing how long the threshold has been exceeded, time intervals between threshold exceeding and releasing new memory, etc (21).

| Parameter Name | Brief Description |
|---|---|
| abapheap_area_dia | Heap Memory Limit for Dialog Work Processes |
| eminitial_size_MB | Extended Memory Pool Size |
| emglobal_area_MB | Size of the Extended Global Memory |
| rdispROLL_SHM | Roll Buffer Size |
| rdispROLL_MAXFS | Maximum Roll File Size |
| rdispPG_SHM | Paging Buffer Size |
| rdispPG_MAXFS | Maximum Size of the SAP Paging File |
| zttaroll_first | Initial Allocation Size from the Roll Area |
| zttaroll_area | Roll Area |
| emblocksize_KB | Segment Size of the EM |

Table 4.4: Profile Parameters for Memory Management Limitations

| Parameter Name | Brief Description |
|---|---|
| emstat_log_size_MB | Statistics - User Context Size |
| emstat_log_timeout: Statistics - User Context Size | |

Table 4.5: Profile Parameters for Memory Management Statistics

## 4.3 Other Classifications of Profile Parameters

### 4.3.1 Dynamic and Non-Dynamic Profile Parameters

Roughly speaking, dynamic parameters are parameters whose values can be changed dynamically while the system is running. Dynamic changes are lost the next time the SAP is restarted. Conversely, changes of non-dynamic parameters take place after restarting the system.

### 4.3.2 Operation System Specific Profile Parameters

Profile parameters that specify file names paths are operation system specific. These parameters have to be set according to the operating system. It is not

| Parameter Name | Brief Description |
|---|---|
| esdisclaim_threshold_MB | Controls the Release of EM |
| esdisclaim_coasting_time_alloc | Controls the Release of EM |
| esdisclaim_coasting_time_alloc | Controls the Release of EM |
| esblockdisclaimsize_KB | Controls the Release of EM |
| esfreelist_compactor | Improves the Use of EM |

Table 4.6: Profile Parameters for Release of SAP Memory for the Operating System

recommended to change the values of these profile parameters. Thereby the use of a parameter value can be restricted to a certain operating system. An acronym for the name of the operating system must be places in front of the parameter.

Permitted acronyms for various operating systems are: aix, hpux, osf1, sinix, sunos, wnt (Windows), and as4 (iSeries).

### 4.3.3 SAP System specific Profile Parameters

If a parameter is preceded by an SAP System name, this assignment is only valid for this SAP system. The way in which the SAP system name is separated from the parameters indicates that the name used in the prefix is the name of an SAP system. SAP System names and parameters must be separated by /system-name/.

# Chapter 5

# Concept of a Profile Management Unit

*"It is a paradoxical but profoundly true and important principle of life that the most likely way to reach a goal is to be aiming not at that goal itself but at some more ambitious goal beyond it." Arnold Toynbee*

This chapter of the diploma thesis presents some possible solution concepts for the stated problem. Further on, based on the chosen solution concept it proposes a concept of a profile management unit in terms of its definition and the requirements it should fulfill. Subsequently, it discusses the benefits of the integration of that unit with the Adaptive Computing Controller. This chapter is divided in three sections as follows:

**Section 5.1**

This section describes possible ways of solving the problem with their advantages, disadvantages and feasibility. Finally, the most suitable way for solving the problem is pointed out.

**Section 5.2**

This section defines a centralized profile management unit and analyses its functional and non-functional requirements, from a black-box perspective, without considering its implementation yet.

**Section 5.3**

This section makes a step forward from the goal of centralization and basic functionalities and touches upon a very essential issue : the integration with the Adaptive Computing Controller

## 5.1 Discussing Possible Solution Concepts

Certainly, several solution concepts to the stated problem can be outlined. Based on the fundamentals of the diploma thesis, a careful investigation of the possible solution concepts for solving the problem is essential. This is necessary so that finally the most optimal solution concept to the problem is chosen. This section describes several possible solutions with respect to their advantages, disadvantages and the degree of complexity of their realization. Finally, it gives arguments to support the chosen solution way.

### 5.1.1 Solution Concept 1

Memory is nowadays probably the most important computing resource of an enterprise when it comes to application performance. That is why the first solution concept to be discussed tries to take advantage of applying memory virtualization. There are many possible solution concepts to realize memory virtualization, one could be through Virtual Machine Monitors [1] Concretely, this solution concept views a shared resource model, in which applications or services use only as much memory as required. It consists of pooling server RAM and making it available to virtual machines (VMs) on an as-needed basis. In this way cluster and server utilization can be improved via the sharing of resources. What is more, efficiency can be increased and runtime for data intensive I/O bound applications can be diminished. What is more, isolation is achieved, which improves system security and prevents software failures in one VM to effect the other VMs. It allows a system manager to configure the environment in which guest OSs will run. This also allows applications on multiple servers to share memory, decreasing total memory needs. Certainly, memory virtualization increases the memory availability through making RAM availabble to virtual machines. The final goal, however is increased applications' performance, as well. To make the advantages of this solution concept available for the SAP application, the application should be able to flexibly adjust itself to changing resources. What is very important to consider in the context of SAP applications'performance is the systems' parameterization. In the context of SAP systems organized in a landscape, the first idea is to search for a way in Solution Manager 2. Nevertheless, among its functionalities there are no options for centrally managing systems' parameterization. Therefore, to get optimal use of the resources, applications can be initially set to "see" more resources than there actually are. The advantage of this solution concept is that the applications would work with as much as needed resources and when more resources are needed and assigned, the applications can would optimally assimilate them.

The fact that the applications are set to "see" more resources than available imposes that for the realization of this solution both real and virtual resources should be provided. In case virtual resources are needed, there is a so called

[1] A virtual machine monitor is a host program that allows a single computer to support multiple, identical execution environments. All the users see their systems as self-contained computers isolated from other users, even though every user is served by the same machine. In this context, a virtual machine is an operating system that is managed by an underlying control program.

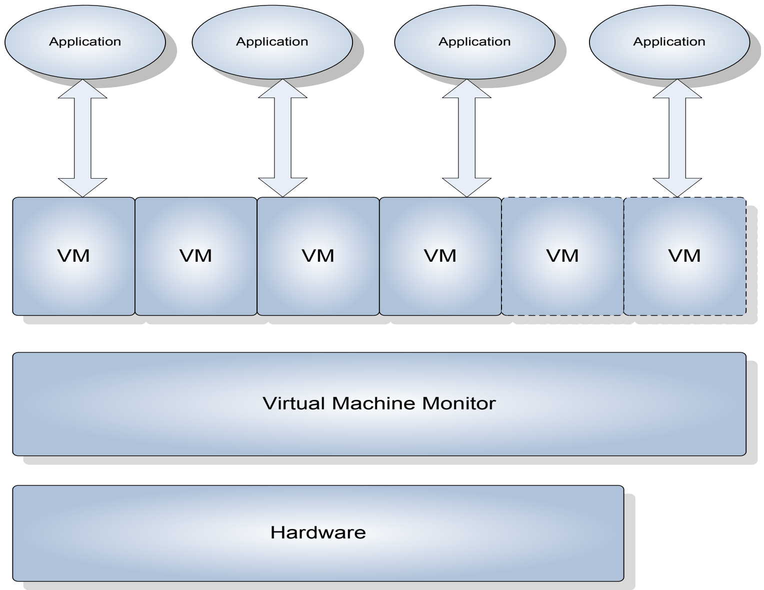


Figure 5.1: Solution Concept 1

'shadow page' that turns the virtual resource into physical. In the context of virtualization this is feasible since the probability that all the real resourcces are needed in one time-slice is ignorable. In the context of SAP, however, this solution is not SAP certified. Therefore this solution concept turns out to involve a high risk and proves itself as rather unsuitable.

### 5.1.2 Solution Concept 2

The goal of the second solution concept is to overcome the disadvantage of the first one. Therefore here the need for virtual resources should be avoided. One way to achieve that can be by making the application check for itself what resources there are available. This check takes place periodically (for instance every 10 seconds) and if there are more available resources it takes benefit of them. The benefits of this solution concept, however, with respect to the complexity of realizing it are not satisfactory. More to be considered is that solving the problem in this way does not take into account reacting to extreme situations. For example, if one system requires more resources and another application should be stopped to fulfill that. Therefore a precise mechanism for optimally assigning the computing resources is needed. Such a mechanism is already provided by SAP AG, and that is namely the Adaptive Computing Controller. Consequently, the next way of solving the problem concentrates on taking advantage of the ACC.

### 5.1.3 Solution Concept 3

The third solution concept to be described discusses the consequences of application's behavior when resources have been assigned by the ACC. The ACC makes numerous advantages available. It makes it possible to easily add and remove computing resources with minimal administrative effort. The tool supports a variety of operating systems and blade servers from leading vendors. Using application services, the Adaptive Computing Controller tool manages and assigns instances involving SAP applications to a dedicated computing resource. The tool requires no local disk space to run SAP applications; rather, it stores application data on a centralized storage system within the network. The Adaptive Computing Controller tool builds connections between computing and storage resources, provides a transport layer for virtualization, and supports different network topologies (for example, TCP/IP, iSCSI, and Fibre Channel).With the Adaptive Computing Controller tool, adaptive business solutions can be operated, observed, and managed from a central point. The tool uses standard technologies, and interfaces with SAP Solution Manager's controller command interface for communication with third-party software. So far that appears to suit well. However, one very important question rises. Namely, what happens in the system when new resources are made available? Do the profile parameters change their values? Solution Manager also provides options for central system management, system landscape analysis, overview of system groups, etc. What remains unsolved however is the centralized profile management and the focus on the system parameterization. The goal of the third solution concept is to work out these missing issues. Certainly to be adjusted is the application parameterization. The system's parameters need to be managed centrally with the possibility to work hand in hand with the ACC. The technical requirement for realizing this solution concept is the careful investigation of the profile parameters, responsible for the assimilation of hardware resources, namely CPU, RAM, I/O, and defining the degree of dynamics of an SAP system with respect to those parameters. After fulfilling the technical requirement a solution can be formulated which determines how multiple profile parameters change their values synchronously with the assignment of new computing resources. This is explained in more detail in section 5.3. Further on this diploma thesis concentrates on this solution concept to the problem and works out a concept for it and a corresponding implementation.

## 5.2 Definition and Requirements

A landscape of multiple systems along with their parameterization, defined by numerous profile parameters leads to the demand for central management. In the context of a landscape with hundreds of systems, centralizing the functions for monitoring, controlling and adjusting system parameterization of the SAP systems promises numerous benefits. Managing the system profile parameters requires to be supported by a tool which externally accesses the single systems. Options for proving certain parameter values from the profile files and comparing them to the corresponding values on other instances should be available. Furthermore, there should be the possibility for changing or deleting parameter values, as well as multiple change. Thereby the complexity of logging on to every system and making changes at single steps can be speared. A clear overview of the functions is given by way of illustration through a use-case diagram (26) what functions should be available for the end user and mentions along the functional capabilities. Further on this use-case diagram can be used for the purpose of the software development in the first phase of the software life-cycle (the requirement analysis).

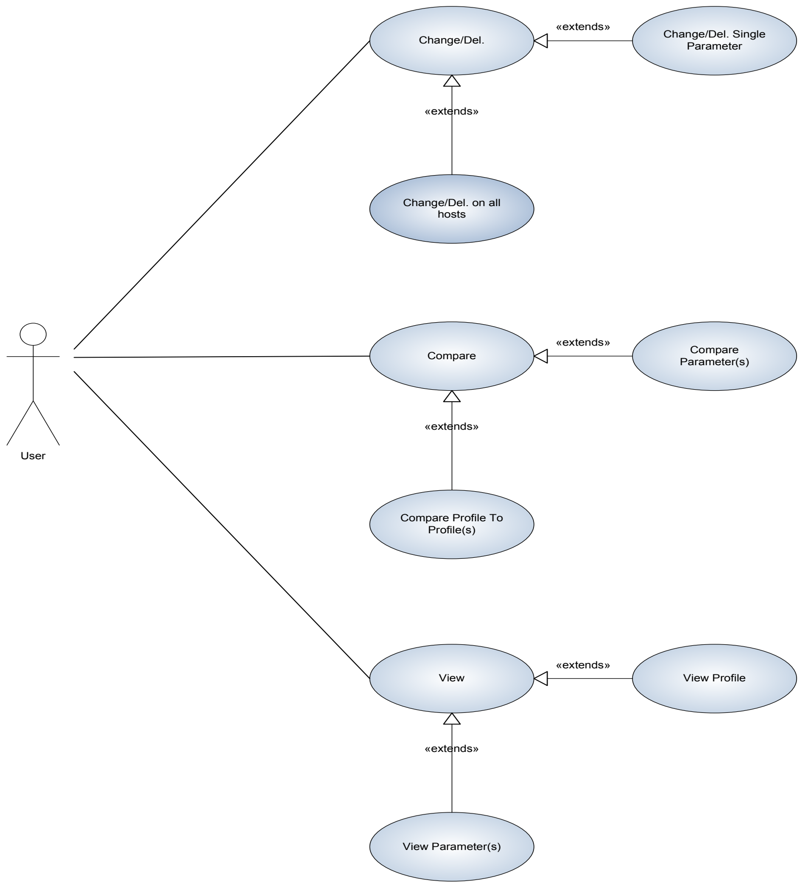


Figure 5.2: Use Case Diagram

- View Parameter

  A function for searching and viewing a single profile parameter, the profile parameters of a system or check the value of a certain parameter on different systems.

- Compare Parameters

  A function for filtering desired parameters and systems, i.e. searching for values in specific ranges, comparing differences between one and the same parameter on multiple systems, or comparing system's profile to system's profile.

- Change Parameters

  Changing the values of a those parameters, that are unfixed, which categorizes a functional capability.

- Check

  Checking if certain values are out of range, or if some drastic changes have occurred in a short time.

In order for the key functions to be implemented, the unit should be able to execute the following:

- Connect to an SAP System Landscape,
- Connect to different systems from the system landscape,
- Show the active instances of the different systems,
- read single profile parameters and their values,
- read whole profiles,
- set changes on one or more parameter values,
- activate new versions of profiles.

Apart from the functional, some non-functional requirements are important to be considered:

- Reliability

- Usability

  The application should be easy to understand and work with.

- Performance

  The waiting times for search should not be too long.

- Re-usability

- Security

  Only authorized users should be granted access to the system.

- Changeability

The application should be implemented in an understandable way and properly documented, so that changes can be easily made. This concept can find broad application in administration of system profiles within the framework of an SAP system landscape, since it solves the task of capturing and managing the system profiles of ABAP instances over the landscape. It can be used in data center environments to provide system landscape administrators with a quick and clear overview of the parameterization of the technical systems and their instances. Landscape administrators can get information about SAP systems in the landscape, their active instances and the corresponding profile files along with the individual parameters.

## 5.3 Integration with the Adaptive Computing Controller

The integration between the profile management unit, defined in the previous section, and the ACC has a significant meaning for improving systems' performance is SAP systems' landscapes. The integration with the Adaptive Computing Controller means that when hardware resources in terms of CPU or RAM are assigned to a system, the SAP systems should operate effectively and efficiently on them. There should be the possibility to adjust the system's parameterization to work with these resources optimally. This adjustment should be automated and should require possibly less human interaction. Responsible for setting the system configuration for the runtime environment and assimilating the hardware resources are the profile-parameters. Therefore, to achieve that integration, a concept should be worked out to define certain change of the profile parameters as a response of certain change of the resources on which the SAP system is running. The parameters responsible for the hardware assimilation can roughly be divided into three groups:

- Memory related,
- CPU related,
- Input and Output related profile parameters

The Memory related profile parameters are quite many and the memory management system already has introduced some dynamics. The CPU related profile parameters are those, that show the number of processes. There are no parameters for the systems' input and output.

In this sense an algorithm should be determined for an executable module to be performed on resource assignment. Every time resources are granted to a system, the executable module should check if the system's parameterization is adjusted to assimilate these resources. This means to perform a check for e.g. if the number of work processes is not restricted to a lower amount of CPU resource. If that is so, then the parameter's value which reflects this number should be increased. Similarly, a check should be performed for the profile parameters responsible for the RAM.

In this way the problem of over-granting computing resources can be solved and system's performance can be increased.

# Chapter 6

# Software Development

*"Design is not just what it looks like and feels like. Design is how it works."*
*Steve Jobs*

Having formulated the concept of a profile management tool, this chapter describes its development through software life-cycle and more concretely the way of implementation in terms of development environment and implementing the most important methods.

A software life cycle model represents the most important steps of a software project from conception to the ready product. It specifies the relationships between project phases, including transition criteria, feedback mechanisms, milestones, baselines, reviews, and deliverables. Typically, a life cycle model addresses the phases of a software project: requirements phase, design phase, implementation, integration, testing, operations and maintenance (28). During the requirement analysis, the customers' requirements are investigated with respect to their feasibility. The design phase describes how the software should be implemented with regard to a definite computer platform. The implementation phase generalizes the coding of the design in a certain programming language. The acceptance testing phase has the goal to determine the product's suitability and its validity.

| Input Product | Process | Output Product |
|---|---|---|
| Communicated Requirements | Requirements Engineering | Requirements Specification Document |
| Requirements Specification Document | Design | Design Specification Document |
| Design Specification Document | Programming | Executable Software Modules |
| Executable Software Modules | Integration | Integrated Software Product |
| Integrated Software Product | Delivery | Delivered Software Product |
| Delivered Software Product | Maintenance | Changed Requirements |

Table 6.1: Processes and Products of the Waterfall Model

The most commonly used software models are the spiral model [1], the waterfall model[2], the evolutionary model [3], and the iterative development[4]. Undoubtfully, one of the most widely accepted and used software model is the classical model (The Waterfall Model). Although it has often been criticized for its rather low flexibility, it remains very suitable for projects with steadily defined requirements, that are not likely to change. Therefore, the waterfall model proved itself to be suitable enough for the profile management tool. This model depicts the software life cycle as a series of products and processes. Each process transforms a product to produce a new product as output. Then the new product becomes the input of the next process. Products undergo multiple refinements. Table 6.1 lists the processes and products of the Waterfall Model.

The stages of the Waterfall Model are:

- Requirement Analysis Definition

[1] http://www.buzzle.com/editorials/1-13-2005-64082.asp
[2] http://www.buzzle.com/editorials/3-13-2005-67039.asp
[3] http://hbswk.hbs.edu/item/2201.html
[4] http://c2.com/cgi/wiki?IterativeDevelopment

All possible requirements of the system to be developed have to be carefully understood in this phase. Requirements are sets of functionalities and constraints that the end-user expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed with respect to their validity and the feasibility of incorporation in the system to be developed is studied. As a result, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

- System Software Design

  Before a starting the coding, it is highly important to understand what is there to be created. The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

- Implementation Unit Testing:

  Having the design documents ready, the work is divided in modules/units and the coding can be started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality; this is referred to as Unit Testing. Generally speaking, unit testing verifies if the modules/units meet their specifications.

- Integration System Testing:

  As specified above, the system is first divided in units which are developed and tested for their functionalities. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as desired. After successfully testing the software, it can be delivered to the customer.

- Operations Maintenance:

This phase of the Waterfall Model is a recursive and therefore also a long-lasting phase. In the majority of cases, problems with the system developed come up after its practical use starts, so the issues related to the system are solved after deployment of the system. Not all the problems come in picture directly; they rather occur with the time and need to be solved; hence this process is referred as maintenance.

When choosing the Waterfall Model, it is very essential to be aware also of its disadvantages. The Waterfall Model has mainly been criticized for its poor flexibility on adding system requirements. To begin with, there is a high probability that not all of the requirements are clarified at once. In most cases customers keep on adding requirements. This turns out to be a strong disadvantage as it negatively reflects the system development. What is more to be considered is that the problems with one phase are never solved completely during that phase and in fact many problems regarding a particular phase rise after the phase is signed off, this results in a poorly structured system and limited flexibility.As the requirements of the customer goes on getting added to the list, not all the requirements are fulfilled, this results in development of almost unusable system. These requirements are then met in newer version of the system; this increases the cost of system development.

DESIGN NOT FINISHED YET xxxxxxxxxxx

The requirement analysis is already done in section ??. The design phase concerns the following: Connecting to SLD To be able to get an overview of all the systems, it is important to read the sld data, i.e. to obtain a list of the names of the technical systems, belonging to the SAP system landscape. The module for connecting to SLD gets as an input the URL address of the SLD, a corresponding user name and password. As an output a list of the available technical instances is delivered. The connection to the SLD is realized through the Common Information Model (CIM). CIM is a standard of the Distributed Management Task Force(DMTF)[5]. The system-specific data such as application server name, system number and so on is then used for establishing the connection to ABAP systems. Connecting to ABAP systems The connection to the different systems of the landscape is then established through the SAP Java Connector [6](SAP JCO). It is a middleware component that enables the development of SAP-compatible components and applications in Java. SAP JCo supports communication with the SAP Server in both directions: inbound calls (Java calls ABAP) and outbound calls (ABAP calls Java), in this case inbound calls. Provided the application server name, system number, client, user name and password, the connection to the different ABAP systems can be realized. This is the prerequisite for executing remote function calls [7] and calling Business Application Programming Interfaces [8] (BAPI). **E**xecuting Functional Modules Before being executed the corresponding functional modules for showing the active instances of the different systems, reading single profile parameters and their values, reading whole profiles, setting changes on one or more parameter values and activating new versions of profiles, the functional modules should be found and tested. Then they can be called and executed and their output can be read. Provided the connection to the ABAP system is successful, the execution of BAPIs is possible.

[5] www.dmtf.org
[6] http://help.sap.com/saphelp nw04/helpdata/en/6f/1bd5c6a85b11d6b28500508b5d5211/content.htm
[7] http://saprfc.sourceforge.net/
[8] http://www.sapbapi.com/

# Chapter 7

# Conclusions and Future Work

*"It's more fun to arrive a conclusion than to justify it."*
*Malcolm Forbes*

This diploma thesis discusses the granting and assimilating of computing resources in SAP landscapes and raises the topic of assuring complete benefit from the advantages of the ACC. This chapter provides a summary of this work and gives the directions for future work.

## 7.1 Thesis summary

Centralized profile management in SAP landscapes is very helpful for the optimal assimilation of computing resources in SAP system landscapes. This diploma thesis discusses the assignment and assimilation of computing resources in SAP system landscapes. It investigates the present state of the art to figure out that SAP systems have a relatively limited dynamics when consuming hardware resources is concerned. The diploma thesis also investigates the profile parameters responsible for the proper incorporation with the ACC, namely the parameters of the memory management and those of the CPU. This thesis proposes a concept for a profile management system. Through central profile management it becomes possible to monitor and adjust systems' parameterization and therefore to achieve a better assimilation of computing resources.

## 7.2 Future Work

This diploma thesis can be extended in a few directions as outlined below.

### 7.2.1 Automating the Change of a group of Profile Parameters

The automated change of those profile parameters, that are responsible for consuming the granted hardware computing resources is going to be very helpful in SAP system landscapes. As a result of this diploma thesis, there is a concept presented for monitoring and changing parameter values. Based on that, an algorithm can be carried out to define exactly the change of the profile parameters corresponding to the granting of determined resources. This can be done by a precise investigation of the memory management system and its profile parameters. These parameters are the largest group of parameters that should be taken into consideration. What is left are just the parameters responsible for the CPU assimilation, which are presented by the numbers of processes.

### 7.2.2 Measurement and Evaluation of Systems' Performance

In order to obtain an accurate picture of the achieved incorporation of the computing resources in the SAP system landscape, an evaluation of the systems' performance with and without adjusting the parameterization is essential. SAP AG already has a standard for expressing systems' performance: SAPS. With its help the performance improvement on parameterization change can be measured and presented. Efforts in this direction can be very helpful for deciding weather or not applying the concept of profile management is worthy.

### 7.2.3 Benefits of Centrally Managing Groups of Profile Parameters

There might be dispute whether or not central profile management is has advantages only in hardware resource incorporation. This diploma thesis can be extended by showing other benefits of centrally managing other groups of profile-parameters than those of the CPU and the memory management system. For example, centrally managing the parameters for the configuration of login security can contribute for a better monitoring and if needed modification of some security issues in the system landscape. Analogously, other groups of profile parameters[1] can be investigated in order to extend the applications of the concept for profile management.

[1] http://www.mariewagener.de/node/129

# Bibliography

[1] Kehinde Eseyin (2005), SAP Benchmark Glossary, SAPS , http://www.sap.com/solutions/benchmark/glossary.epx#SAPS, 07.03.2009

[2] Chris Wolf and Erick M. Halter(2005) Virtualization : from the desktop to the enterprise

[3] kernelthread.com (2004), An Introduction To Virtualization, Amit Singh, http://www.kernelthread.com/publications/virtualization/, 07.03.2009

[4] VMWare,Understanding Full Virtualization, Paravirtualiza- tion, and Hardware Assist, White Paper, (01/01/2007) http://www.vmware.com/ les/pdf/VMware paravirtualization.pdf, 07/03/2009

[5] Business Trends Quarterly ( n.d.), Virtualization:Big Picture http://btquarterly.com/?mc=virtualization-big-picture&page=vir viewresearch, 07.03.2009

[6] RNA Networks (n.d.), Memory Virtualization, the third Wave of Virtu- alization, http://www.rnanetworks.com/news/press-releases /1-latestnews/ 191-vmblog-memory-virtualization, 07.03.2009

[7] J. Moon, S. Park, and J. Lee (2008), Resource Management through Resource Virtualization in Distributed Network Environments

[8] S. Shumate (2004), Implications of Virtualization, http://www.dell.com/downloads/global/power/ps4q04-20040152-Shumate.pdf

[9] Ph. Winslow, R. Semple, J. Maynard (2007), Desktop Virtualization Comes Of Age, http://virtualization.sys-con.com/node/466375

[10] Intel Software Network (02.10.2008) Verifying ISV Applications Running Under VirtualizationVerifying ISV Applications Running Un- der Virtualization, http://software.intel.com/en-us/articles/verifyingisv-applications-running-under-virtualization/

[11] Intel Software Network (October 2008), Verifying ISV Applications Running Under Virtualization, 20.04.2009, http://software.intel.com/en-us/articles/verifying-isv-applications-running-under-virtualization/

[12] ToolBox for IT, Understanding the SAP System Landscape, Kehinde Eseyin, 1/20/2007, 14.01.2009, http://it.toolbox.com/blogs/sap-library /understanding-the-sap-system-landscape-14030

[13] SAP Knowledge Warehouse, SAP System Landscape Directory, n.d., 14.01.2009, http://help.sap.com/saphelp_ nw04/helpdata/en/31/f0ff69551e4f259fdad799a229363e/frameset.htm

[14] SAP NetWeaver Library, Technical Systems, n.d., 18.01.2009, http://help.sap.com/saphelp_nw70/helpdata/en/24 /8fa93e08503614e10000000a114084/frameset.htm

[15] SAP Knowledge Warehouse, Profile Parameters, n.d. 28.12.2008, http://help.sap.com/saphelp_nw04/helpdata/en/c4 /3a610f505211d189550000e829fbbd/frameset.htm

[16] SAP Knowledge Warehouse, Start Profiles, n.d. 28.12.2008, http://help.sap.com/saphelp_nw04/helpdata/en/c4 /3a610f505211d189550000e829fbbd/frameset.htm

[17] SAP Knowledge Warehouse, Default Profiles, n.d. 28.12.2008, http://help.sap.com/saphelp_nw04/helpdata/en/c4 /3a610f505211d189550000e829fbbd/frameset.htm

[18] SAP Knowledge Warehouse, Instance Profiles, n.d. 28.12.2008, http://help.sap.com/saphelp_nw04/helpdata/en/c4 /3a610f505211d189550000e829fbbd/frameset.htm

[19] SAP Knowledge Warehouse, Transport Profiles, n.d. 28.12.2008, http://help.sap.com/saphelp_nw04/helpdata/en/ac /841c3717419328e10000009b38f936/frameset.htm

[20] SAP Netweaver Library, Memory Management: Basic Concepts, n.d., 14.01.2009, http://help.sap.com/saphelp_nw70/helpdata/EN/34 /d9c8b3c23c11d188b40000e83539c3/frameset.htm

[21] SAP Netweaver Library, Profile Parameters of Memory Management, n.d., 16.01.2009, http://help.sap.com/saphelp_nw70/helpdata/EN/34 /d9c8b3c23c11d188b40000e83539c3/frameset.htm

[22] SAP Netweaver Library, Standalone Enqueue Server, n.d., 12.02.2009, http://help.sap.com/saphelp_nw70/helpdata/EN/cb /68edd90237c04aa9591469a4584cd1/frameset.htm

[23] SAP Netweaver Library, Profile Parameters for the Enqueue Clients, n.d., 12.02.2009, http://help.sap.com/saphelp_nw70/helpdata/EN/cb /68edd90237c04aa9591469a4584cd1/frameset.htm

[24] SAP Netweaver Library, Profile Parameters for the Standalone Enqueue Server, n.d., 12.02.2009, http://help.sap.com/saphelp_nw70/helpdata/EN/cb /68edd90237c04aa9591469a4584cd1/frameset.htm

[25] SAP Netweaver Library, Profile Parameters for the Enqueue Replication Server, n.d., 12.02.2009, http://help.sap.com/saphelp_nw70/helpdata/EN/cb /68edd90237c04aa9591469a4584cd1/frameset.htm

[26] G. Booch, J. Rambough, I. Jacobson (1996), The Uni
ed Modelling Laguage for Object Oriented Development, Rational Software Corporation

[27] Rautenstrauch, Schulze (2003) Informatik fuer Wirtschaftswis- senschaftler und Wirtschaftsinformatiker, p.319

[28] R. Dumke (2003) Software Engineering, p.18