Date of acceptance Grade

Instructor

Service-Oriented Process Models in Telecommunication Business

Sauli Talja

Helsinki 06.01.2013

The University of Helsinki

The Department of the Computer Science

M. Sc. Thesis

HELSINGIN YLIOPISTO・HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI・

Tiedekunta/Osasto・Fakultet/Sektion – Faculty/Section Faculty of Science	Laitos・Institution・Department Department of Computer Science	
Tekijä・・・Författare・Author Sauli Talja		
Työn nimi・・Arbetets titel・Title Service-Oriented Process Models in Telecommunication Business		
Oppiaine・Läroämne・Subject Computer Science		
Työn laji・・Arbetets art・Level M. Sc. Thesis	Aika・・Datum・Month and year 06.01.2013	Sivumäärä・・Sidoantal・Number of pages 82

Tiivistelmä・・Referat・Abstract

The thesis concentrates on to evaluate challenges in the business process management and the need for Service-oriented process models in telecommunication business to alleviate the integration work efforts and to reduce total costs of ownership. The business aspect concentrates on operations and business support systems which are tailored for communication service providers. Business processes should be designed in conformance with TeleManagement Forum's integrated business architecture framework. The thesis rationalizes the need to transform organizations and their way of working from vertical silos to horizontal layers and to understand transformational efforts which are needed to adopt a new strategy.

Furthermore, the thesis introduces service characterizations and goes deeper into technical requirements that a service compliant middleware system needs to support. At the end of the thesis Nokia Siemens Networks proprietary approach – Process Automation Enabling Suite is introduced, and finally the thesis performs two case studies. The first one is Nokia Siemens Networks proprietary survey which highlights the importance of customer experience management and the second one is an overall research study whose results have been derived from other public surveys covering application integration efforts.

ACM Computing Classification System (CCS):
C.2.4 [Distributed Systems]
D2.12 [Interoperability]
H5.3 [Group and Organization Interfaces]

Avainsanat – Nyckelord・Keywords
Business Process Management, Communication Service Provider, Operations and Business Support Systems, Process Automation Enabling Suite, Service-Oriented Architecture, TeleManagement Forum, Web Services

Säilytyspaikka・Förvaringställe・Where deposited
Kumpula Science Library, serial number C-

Contents

1 Introduction

Business Process Management (BPM) is a diversity discipline strategy to take care of enterprise's business processes. Organizations can attempt to avoid IT systems' integration barriers by using service-oriented process models and middleware solution suites. Requirements originate from the evolution in market dynamics, prevailing business models, globalization and collaboration. Organizational silos are a traditional way of classifying and differentiating responsibilities. Normally it means that each silo is taking care of its own applications, services, processes and system infrastructure interfaces during the system's life cycle. The core idea of the new approach is to transform organization structures from the vertical silos to horizontal layers so that the same operation is provided always by the same service, irrespective of the silo. This approach creates economies of scale and provides lowered cost because of reusable services. The new architectural paradigm is thus based on horizontally divided operations. Furthermore, it is important to understand the organizational change management process which is required to adopt the new business approach. Strategic alignment frameworks help to alleviate integration barriers and they offer different strategic paths where enterprises should aim their course to align business, IT and organizations closer together.

There are multiple methods for the applications to communicate over the internet or over any other public or private communication network. The traditional way is based on tight application coupling and awareness of the business process co-operators and other peers. As the opposite, the new idea is to minimize tight coupling and just reserve and bind as few resources as possible and still to provide the qualified but loosely coupled service. Besides multiple architectural communication alternatives, there are also multiple techniques to integrate heterogeneous applications together. These two techniques need always to be managed together to implement flexible and inter-operable services.

Service-oriented architecture (SOA) is the concept to organize and utilize resources which exist under the control of separate ownership silos [OAS06]. SOA provides a unique way to offer, discover, interact with and use resources to produce desired effects consistent with measurable preconditions and expectations. Service-oriented architecture is all about constructing standard interfaces to access different business functions that are exposed by various core systems. This grants a great opportunity for companies to integrate their business systems as services. The Service-orientation is the key success factor to break down vertical silos and to transform organizations towards horizontal layers.

Process Automation Enabling Suite (PES) is a *Nokia Siemens Networks (NSN)* proprietary set of service infrastructure components for building, deploying, managing and aligning Service-oriented process models. PES offers the SOA based middleware platform and the Service-oriented business process models tailored especially for *Communication Service Providers's (CSP)* needs. These process models have been defined based on *TeleManagement Forum's (TMF) initiative (NGOSS)* which standardizes the scoping work in any customer project and support collaboration efforts during the integration architecture designing process. PES supports up-to-date technical demands which are required by the SOA middleware systems.

Communication service providers use a great variety of software applications to manage their vital business processes. A very important role in supporting daily operations, such as end-user provisioning, service provisioning, service management, charging, rating, billing and customer care, are executed by the *operations and business support systems (OSS/BSS)* which automate customer-, service- and network infrastructure related processes. The core idea of NGOSS is to standardize these processes.

The research in the thesis concentrates on the telecommunication software, especially on operations and business support systems software. Nevertheless, it does not exclude the fact that similar analysis can be applied to any other vertical or horizontal software market. In parallel, the thesis introduces the nature of complex OSS/BSS systems and analyzes reasons why it is important to deploy Service-oriented business process models on the basis of the system complexity and modularity. Furthermore, the thesis highlights the importance of *Customer Experience Management (CEM)* for continuous interaction with the customer to be able to react quickly on new customer demands. Finally, the thesis summarizes evaluation results received from SOA adoption projects and estimates how suitable PES is for the overall adoption process.

In overall, the thesis consist of the following chapters. Chapter 2 concentrates on common challenges in the area of Business Process Management. Chapter 3 concentrates on the horizontal transformational process. Chapter 4 introduces characteristics which the services need to comply with. Chapter 5 introduces the technical aspects which the services and the service platform need to fulfill. Chapter 6 introduces PES and evaluates the results received from the analyzed application integration projects. Finally, Chapter 7 summarizes the thesis, gives conclusions and estimates reasonable future steps.

2 Challenges in Business Process Management

A business process is an activity or set of activities that accomplish a specific organizational goal [Gom08]. It can also be visualized with a workflow as a sequence of activities. Business Process Management (BPM) defines behavioral roles of business processes that are seen as composition of activities with or without human intervention. A business process is a collection of related, structured activities and tasks that produce a specific service for a particular customer. Therefore, business process management is a systematic approach to improve business processes throughout bringing business and IT persons together with a shared set of languages, tools and strategies.

This chapter goes through those challenges which Communication Service Providers (CSP) have faced during the past few years when managing their business processes. Furthermore, the chapter presents the nature of OSS/BSS systems and introduces a new enterprise framework for business and IT alignment and recommends how the change management process should be adapted while moving from vertical product silos towards horizontal service layers. Besides, there are Total Cost of Ownership calculations included which are based on traditional and the SOA middleware integration.

2.1 Evolution in Business of Communication Service Providers

In recent years, enterprises have heavily invested in *Enterprise Resource Planning (ERP), Customer Relationships Management (CRM), Supply Chain Management (SCM)*, as well as, in other enterprise-level systems. Data, information and knowledge sharing across these systems flows via electronic data interchange (EDI) systems which is a widely used electronic data document messaging standard [JeD09].

System integration is typically done with proprietary solutions and every organization has its own home-made approach of interacting with its partners. While EDI is a widely adopted exchange standard, enterprises still process EDI messages using non-standard, proprietary solutions. This makes it hard to incorporate message exchanges in more comprehensive enterprise-level processes. Point-to-point integrations have typically bundled integration and business logic together existing in either or both the end points.

ERP mainly focuses on supply and demand for key resources and materials. ERP applications consist of operations, such as human resource management, financial operations, catering, legal management and insurance management. Besides, CRM applications usually contain operations related to order management, customer management and trouble ticket handling. CRM is mainly tailored to perform customer operations. Thus, CRM system should enable an organization to gain greater customer insights and experiences in order to provide more appropriate products and services for them, meanwhile ERP has a stronger focus on making routine internal operations and processes to work more effectively [KiB07], whereas SCM mainly concentrates on managing networks. Some differences also arise in terms of the back-end focus on ERP versus the front-end focus on CRM. Whilst ERP is used by back-end staff, such as financial operations and human resources, CRM is used by front-end staff, such as technical call centre, marketing and sales. CRM is used wider for inter-organizational communication whereas ERP is used for intra-organizational communication.

These systems have gradually grown into information silos with very few data sharing across the systems. In parallel, these systems do not provide appropriate flexibility which is needed by constantly changing business processes. Despite of variety of utility programs and workarounds, enterprises have started to find these systems expensive and hard-to-maintain and information systems have started rapidly to become outdated. To fulfil this gap, a new generation of enterprise- and web techniques have emerged which enable flexible information sharing across organizations, departments and systems by providing a core platform for integration [KDK07].

Software vendors disagree over the boundaries of BPM and the discrepancy between them is deep. Each vendor has its unique definition of the BPM market. The same matters in ERP, CRM and SCM business as well. Nobody can explicitly list core processes and operations that should be covered by OSS/BSS. There is no consensus around SOA either. However, all the vendors share the same ultimate goal - to manage business processes more effectively.

SOA in OSS/BSS business is still a definitely new concept but it has been applied in ERP and CRM projects already over a one decade. There are plenty of research results available regarding those projects. The common understanding is that ERP, CRM, SCM and OSS/BSS systems all share multiple common operations in the areas like sales, marketing, customer service, technical support, and product-, service-, network-, order- and revenue management. From this perspective the thesis performs a rough generalization and consolidates these five terms and states that BSS is the tailored CRM for customer operations, whereas OSS is both the tailored ERP for service operations and SCM for network operations.

One globally significant group of companies using these systems are Communication Service Providers which create value for their customers by offering network connectivity and exchanging informative messages their users want to deliver in time. The core business processes in the CSPs' core business area varies within a single organization. On one hand, the core business can be based on physical networks and operating the network maintenance and evolution, on the other hand managing the customers and the usage of the connectivity services over these networks. In the past, CSPs' operations have focused mainly on developing and maintaining the network infrastructure. Eventually, the need for standardization of the systems and operations has become apparent [LFP09]. Finally, when network infrastructure standardization efforts have started to work properly, it is time to take a leap to a next level. Recently, telecommunication business people have realized the importance of adopting a new service strategy and to focus on standardization efforts to streamline business processes. Even if the traditional voice business is still today the dominant source of revenue, the actual need for new services is rapidly increasing. Today, CSPs see their future role as becoming an intelligent trading company providing third party services and applications for the end users.

In overall, CSPs are today still seen as reliable but inflexible network providers with good end-user relationship but who do not understand to exploit their business potentials. Dynamic services as well as satisfying end user experiences are the suggested cornerstones for new business models and revenue flows.

The main challenge to satisfy the end customer is the lack of real-time data related to customer insights and experiences which makes it difficult to react on customer demands. Most of data still exists in silos which makes real-time data mining challenging. CSPs need to enhance their competencies regarding organizations’ internal and external agility and flexibility and to optimize their internal processes and response times. The top-three operations for CSPs are to improve end-user’s satisfaction, increase data network capacity and reduce operational expenditures.

The IT organization’s ability to meet business needs makes co-operation easier and more effective, boosting the value of IT and aligning it closely with business and enterprise strategy. For a company to achieve the maximum value from IT, the organization needs to maximize its credibility and make IT an attractive area for investments. However, poor performance in delivering infrastructure services and business changes, and the lack of payoff from prior projects, can cause organizations to become risk-averse to IT-related investments. This may lead to under-utilization and under-investment in IT which prevents other organizations to realize the full potential of the IT organization and potentially placing the company at a disadvantage against more-sophisticated software vendors [Gom08].

In recent years, software vendors have faced the globalization challenges, increased pace of innovations and highly specialized customer requirements. To be able to serve these challenges, software vendors have focused more on their core competencies, which in turn, required more network-intensive business behaviour. The result of narrow specialization have led to the situation that software vendors needed to acquire knowledge outside of their own core expertise area in order to develop and provide competitive value for their customers [RaW03]. This evolution has led to increased efforts to develop essential assets and capabilities through service ecosystems and virtual breeding environments. The composition and management of virtual breeding environments and strategic navigation within this environment has formed an interesting challenge to software vendors. Therefore, an important challenge is to recognise how enterprises interact with external parties, and that they may have to interact with process workflows defined by external parties. The whole system lifecycle management from service planning, deployment, assurance, fulfilment until the service charging and billing needs to be quick, easy, systematic and reliable. This requires efficient and inter-operable notations and tools in all the service areas [Kil04].

Mainly due to internet revolution new competitors have appeared to the market and started to threaten the traditional business models of telecommunications operators by marketing their services directly to the end-users. On the other hand, some business models of the internet have granted end-users a possibility directly to define new contents using open services and to manage the lifecycle of these services independently. In order to support this approach, it has become mandatory for CSPs to change their old fashioned business models and replace them with more agile processes. This can be done by recognizing the CSPs' key properties that can be provided only in the core network and then offering them through well defined interfaces. End-users may use these capabilities to invent new services, thus generating a fruitful and self-organizing ecosystem around the CSPs' physical telecommunication networks [Ram09].

Most enterprises are nowadays reluctant to invest money for standardized, "bulk" type software products. The evolution of standardized software will inevitable go to the same direction than what happened to the hardware industry approximately one decade ago - the price of the bulk software will inevitably and continuously go lower. The large amount of free software that can be downloaded from the internet for free has given the biggest impact on this evolution. Therefore, it will force the software companies to invent new revenue sources and pricing models by means of new business models and services.

Commercial-off-the-shelf (COTS) products typically fulfil approximately half of the enterprise's business process requirements [JeD09]. The other half of business operations are either in-house developed or outsourced and they need to be integrated into the existing IT architecture. However, multiple players from different industrial areas have to co-operate in order to be able to provide these services. Therefore, the underlying service ecosystem, virtual breeding environment and business processes need to be more dynamic and complex compared with current interoperability requirements. In case this complexity can systematically be integrated together, it will grant a great business advantage for those enterprises who are able to foresee how their dynamicity can be governed in the future. In virtual breeding environments, collaborators constitute a set of activities that should create benefits for all the participants belonging in the same environment. Each participant has different requirements and interests and those need to be taken into account in order to manage and orchestrate collective actions that are required for service offerings.

There are two main strategies to take care of service provisioning. The first one utilizes a common integration platform for integration work. The second one reduces the amount of applications and systems and strives for purchasing them all from a small set of product vendors who can provide large suites of services and business process models. Even if the second strategy is pursued, consolidating to just one single system and vendor is practically impossible. Therefore, some external integration is usually needed. Most CSPs host some sort of OSS/BSS systems. These systems provide the highest level of standardized integration tools and methods but still they do not cover all the enterprise level requirements. The rest are usually customized applications that must be integrated with the IT architecture. The target architecture has to be robust enough to support both COTS and product customizations and to serve seamless interaction between them [JeD09].

CSPs play a very important role in their future’s business models. Their expectations and needs have a great impact on how other roles of business models should be defined. Properly defined business processes can support to elaborate both strategic and operational work of the enterprise, enhance business performance and achieve better cost-efficiency. Provisioning of processes is not as flexible and adaptable operation than performing the same operation for resources. It is important to begin to define processes already at the initial solution requirement phase to be able to utilise processes properly.

The nature and importance of well-defined processes has grown when products, solutions and services are not any more developed and operated by a single enterprise working in a static business environment. Virtual breeding environments will become more and more complex and at the same time the product and service lifecycle shorten. Therefore, business processes without properly defined descriptions and provisioning cannot be further operated and deployed in the future. The operations and business processes should be accurately defined in process models so that enterprises are able to construct and provision new services.

New competitors from many areas, previously unrelated to telecommunication, are entering to the market with new disrupting techniques and business models and trying the get their own share from the total revenue. The technology and market evolution forces continuous investments and need for new services with the latest capabilities. CSPs are therefore looking for alternative options and ways to deploy and manage services faster and with lower cost [Mae09].

Despite the benefits that customer-centric service creation brings to the CSPs, there are many challenges which they must solve first. To begin with, the business models that can be applied are not fully clear. The fact that consumers are allowed to construct their own services and drive service lifecycles, poses great challenges to current business models [Ram09]. A great majority of CSPs who have historically developed and maintained all their applications by themselves to run their operations and business processes, have recently modified their business strategy to concentrate mainly on their core competence area – communication networks and services. For that reason, CSPs have also outsourced software development and maintenance to newly emerging software vendors [FTL09].

The evolution of telecommunication market moves from voice and simple messaging services to heterogeneous value-added data and multimedia services. The evolution does not only concern to move from pure telecommunication networks to IP-based environments, but to exploit markets with innovative offerings to keep the customer loyalty level high and to attract new consumers [NSN10b]. CSPs must focus on confronting individual needs and high expectations from their subscribers. Customers expect to purchase service offerings to meet their personal demands which can be tailored according to their preferences and lifestyle. It will require a flexible, customer-centric approach instead of the old-fashioned, silo-centric system architecture. CSPs are confronting demands to set up a complete service ecosystem and operational environment within a very short time period.

2.2 Nature of OSS/BSS

Operations support systems are software systems which support telecommunication network management operations, such as maintaining network inventory, provisioning services, configuring network, collecting and mediating usage information, monitoring Key Performance Indicators (KPI) and managing faults. Besides, business support systems are software systems that supports customer management processes, such as taking orders, providing customer service, problem handling, collecting, mediating and rating charging data, processing bills and collecting payments [LHF11]. Within the context of this study, the assumption is that OSS/BSS share the same common properties at the operational level than CRM, ERP and SCM systems.

At the early phases of operation supporting systems, software offerings targeted at the management of the telecommunication networks, especially to the resource management layer. More recently, the direction has moved towards supporting the processes near customer interfaces, thereby creating an entry initiative for new vendors, which offers applications and systems from other software industries.

Nowadays, the operations and business support systems offer solutions for managing processes of the telecommunication service providers. These software applications are usually supported by some sort of middleware system which manages data coming both from end-customers and telecommunication network interfaces. The operations and business support systems are large, heterogeneous, composed of numerous applications. Together, the set of OSS/BSS software systems can be seen as single large-scale system, in which each individual software system is seen as module or subsystem, operating independently or inter-operating with others [MTV08]. Normally, these modules have a significant number of interdependencies with each other. Therefore, enterprise level applications integration is usually a highly complex and costly process.

The telecommunication service providers have traditionally preferred to purchase the whole operations and business support systems from a single provider. The main reason for this approach is to minimize the application integration barriers and deployment efforts, which play a demanding role in the system's lifecycle management. The multi-layer OSS/BSS implementation has to be integrated with multiple networks and services hosted by CSP. These networks and services are heterogeneous by their nature, and the multi-layer OSS/BSS solution should be adoptable and flexible enough to be integrated with other business processes for different business purposes. The scalability of the service portfolio of the OSS/BSS vendor is an important factor in enabling service offerings, since the vendor needs to have resources to provide the solution covering several application areas in multiple layers. The complete set of OSS/BSS applications can be offered to the CSP by a vendor as pre-integrated OSS/BSS. Otherwise, the cost benefits would be diminished as a result of high costs when integrating software applications [LMP10].

2.3 Cost Analysis of Service-Oriented Business

When comparing a new service model to a traditional, product-centric business model, initial costs to CSP are lower, fixed costs become variable costs and sources of Total Costs of Ownership (TCO) changes. In the traditional model, especially maintenance costs become higher than in the Service-oriented business model, due to exponential growth of point-to-point connections. Furthermore, integration costs are quite often ignored in the decision making process and the purchase decision is made based on the actual COTS and basic operations that the product provides. A recent study estimates that 92% of an application’s cost occurs after the initial delivery and more than 70% of IT spending goes for maintaining and managing existing applications [Kyt10, HiS10]. The fact weakens resources that could otherwise be allocated to innovate and grow the real business challenges. Figure 1 presents two charts. The left bar presents TCO calculations based on traditional system integration and the right one presents the same based on Service-oriented integration. Traditional point-to-point integration approach has typically bundled integration and business logic together existing in either or both the end points. The SOA based integration architecture separates the interface integration logic from the business logic.

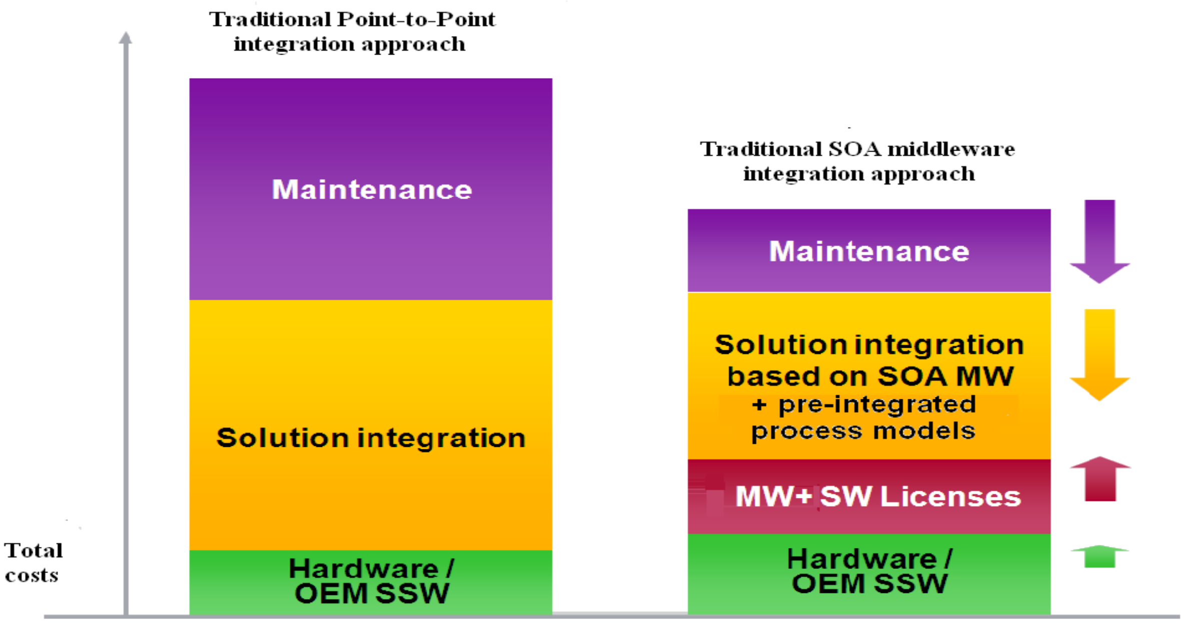


Figure 1. Total costs of ownership for traditional and SOA middleware integration.

The tangible and quantitative benefits is achieved from automating manual processes, leveraging integration efforts by means of the SOA middleware and reducing implementation costs through reusable and pre-integrated process models. Besides, there are additional intangible and quantitative benefits, such as ability to co-operate with external business partners and increasing customer satisfaction. Qualitative benefits can also include the alignment of IT and business processes to achieve overall efficiency improvements.

The inability to understand the value from IT investments is due to the lack of alignment between the business and IT strategies of organizations [HeV99]. In overall, IT investments have special characteristics which make it complicated to evaluate all the risks and benefits. Firstly, the benefits are intangible by their nature and monetary measurements cannot be applied in all circumstances. Secondly, the benefits of IT investments can only be realised during a long period of time which makes short-term results irrelevant. Thus, traditional investment evaluation techniques are not sufficient enough. Most of the traditional investments criteria assume that initial investments, cost of capital, incremental cash flows and the economic time horizon of investment alternatives are pre-known. It is implied as well that all the effects of investments can be traced, measured, and transformed into monetary assets. Intangible costs and revenues are considered to compensate each other and subjective criteria are just ignored. Thirdly, in IT investments, benefits are seen differently by different people. IT personnel may see only technological values, while business personnel see only aspects relevant to business. There has to be somebody who can align all the IT and business aspects together and get the overall strategic picture, and especially, communicate it efficiently back to all interest groups. Probably the possible candidate to carry this action is the chief information officer or the chief technical officer from the top-management. Fourthly, these investments evolve over time and get fused with each other [Hal02].

Figure 1 presented that during the transition process, IT operational costs will not be reduced. Vice versa, they will increase for a short period [Kyt10]. Jefic and Devost in their research study have estimated that the Return on Investment (ROI) period intersects after 21 months, but that it is impossible to apply it straightforward to any other change management process than which they actually performed [JeD09]. In the meantime, they have also noticed that IT employees are still mostly responsible for making decisions related to SOA budgeting and spending, not business employees.

Service-orientation is an evolutionary step towards a multi-layer architecture. Every service applies to the entire customer base with the horizontally layered architecture and especially throughout the enterprise's service portfolio - overcoming vertical silos. CSPs can rapidly deliver differentiated and targeted solutions to meet individual customer demands and market requirements, and thus increasing their profits. The solution is designed around a complete set of CSP’s business process models using an end-to-end customer viewpoint. Services need to be simplified to guarantee seamless integration with other enterprise layer applications and external systems – ERP, CRM, inventory management, SCM, product lifecycle management or any relevant system [Don09].

Re-usable process models should be ranked in a sensible manner so that each organization can market those to the others. Ben-Menachem in his study [Ben05] suggests to assess the business value of reusable process models by means of a classification scheme, which involves categorizing the models, calculating each model’s value and assigning a value coefficient for a model. Calculating the model’s value is based on cost analysis of the implementation costs, the maintenance costs, the estimated replacement costs and the annual business value contributed by that model. In assigning a value coefficient, the business value over a timely period minus the maintenance costs is divided by the costs of replacement, that is, reimplementation of that model. The reusable process models are then ranked based on their value coefficient. The ranking shows which business operations have the highest potential as Web Services.

These reusable, SOA modelled assets can be categorized in many frameworks, such as in process models, workflow patterns, service interfaces and data models. These assets can be implemented to simplify reuse of legacy systems and to encourage incremental migrations to current technologies. This approach helps to minimize the impact on services which use legacy applications by shielding them from migration activities [Wal07]. After realizing instant SOA benefits, CSPs should start thinking about a wider SOA adoption. To achieve this point, they should apply SOA best practices which are based on these pre-integrated business process models [Sne05].

The adaption of design model requires knowledge of how to design building blocks and how to visualize the overview of the system rather than just quickly applying a dedicated design model to solve the actual problem. A developer might rush into using a dedicated model to make the software more flexible by modifying the future requirements which will cause over designing or over engineering [Rah10]. Besides, many persons do not realize that testing is equivalent in time and cost to develop it [Sne06]. These two facts can mainly lead to a model adaption failure. Three estimations can be derived from a common TCO (Total Cost of Ownership) cost analysis. The first one is the ratio between external and internal purchases. The second one is the ratio between product specific software development and customized software development. The third one is the cost ratio between external and internal purchases.

Within the majority of CSPs, the total amount of externally purchased software costs exceeds considerably the costs of internally developed software. Typically, the amount of money spent to external work is approximately triple times higher than the total amount of money spent to in-house development [FTL09]. In other words, the ratio concerns all the operations and business support systems related purchase costs, not only costs related to one specific product or service. Furthermore, the ratio between product specific and bespoken software development were approximated. The product specific costs include product licenses, product specific services, and integration work, while customized software development includes only software services and integration work.

The final estimation is that implementation related in-house costs are approximately double or even triple times higher than implementation related professional services out-sourcing, and that in-house tailored work is about four times as high as bespoken professional consultancy services. According to the same estimation, the proportion of product licenses, software leasing over the internet by means of cloud computing [Yrj11] and customized services are constantly increasing while the use of self-made in-house work is decreasing [FTL09].

Large vendors which are oriented towards operations and business support services have a stronger position in the market segments. These companies provide professional services and are more capable to support their customers due to having more competent resources. Same companies are also compromised to maintain a broad variety of legacy network technologies efficiently which is a very important aspect when systems are eventually going to be upgraded and migrated towards new technologies.

The business always requires a COTS item as the basis. The COTS item is normally a product. The product runs an application. The application performs operations and the application can offer these operations as services. When services are composed together they form a process which is the basic element for daily business. There are actually five rules of thumb to manage and deploy products and services which are now introduced here.

The first one is the rule of the core product. It states that without products there are not services either. The core product is the source of services, such as maintenance, migration and upgrade services. Many CSPs purchase services from the same vendors which they have already bought their products. Therefore, base products and value added services are often coupled together. Enterprises should maintain strong and wide product portfolio to keep CSPs paying for solution implementation and strategic services as well as long-term support agreements and maintenance contracts.

The second rule is the rule of the customer-specific integration minimization. It states that customers try to minimize operational costs by avoiding all unnecessary integration work. Besides, software vendors also try to minimize integration costs to gain higher profits. Therefore, software vendors are more willing to provide ready-made assets which they have already pre-integrated and evaluated as operational units.

The third one is the rule of maximizing commonalities re-usage. It suggests to share similar knowledge base in parallel with design and implementation related assets which are needed for implementing new services. Whenever commonalities exist across ready-made products and new products, by re-using earlier knowledge during the implementation phase of new products, the vendors also improve the quality of the implementation as compared with a “do-it-all-by-yourself” approach [MTV08].

The fourth one is the rule of productizing services in a manner that software vendors can deliver services more efficiently. Productizing of services can raise from modules or design reuse, customer trainings, evaluating maximum peak performance, estimating security threats, system’s scalability or capacity estimations. In other words, it means to sell knowledge re-use as customization work. The knowledge re-use system, such as any sort of centralized data warehouse system plays a very important role here.

The fifth rule is the rule to transform services into products. It should be considered that how the organization should offer new services that generate extra value to their standard products. Services coupled to products can make the products less commodity-like as well as generate new revenue flows and profits, even if the actual product business suffers [Cus08]. The key business strategy in this rule is to deploy pervasive solution on top of the core product.

The fourth and fifth rules are closely connected together and the decision has to be made which service components will be composed together as tightly coupled core product and which service components will be sold separately as value-added services. The strategic approach is to sell the core product with minimum amount of highly-qualified software components and customize loose coupled modules and interfaces on top of that. Besides the service customization, many kind of extra value services can be offered for customers to estimate how customized components and core-product work together. When these customized components eventually become to “de-facto” standardized and extra value of them diminish, then they can be added to the core-product and coupled tightly together.

Software product vendors should treat services as strategic area and as opportunity to increase cost margins and revenues. This business approach toward services should not be limited to the pure software business but it can be extended to other levels, such as telecommunication equipments and computer hardware. Companies have to realize that services are real money which they may have ignored and just hoped that service partners can support them to sell more products. If product revenues disappear, then former partners must compete for same money. The dedicated IT service companies will not politely embrace new business strategy of the product software vendors because those companies have traditionally built their business strategy for installing, integrating, and customizing enterprise systems. However, before the product software vendors get deeper into the service business, such as product customization, integration and tailored measurement and quality services, the change can cause a negative impact on profits until the product software companies gain enough scale and competence knowhow to perform these services in an efficient way. After the migration period, they begin again to make money from services, much like dedicated IT services companies do [Cus08].

2.4 Enterprise Framework for Business and IT Alignment - BOIT

There is still lack of effective frameworks, on which the potential of IT can be based, even though information systems have evolved from their traditional approach of administrative support toward a more strategic role within organizations. The effectiveness of the strategic management of IT depends upon alignment among four categories - the business strategy, the IT strategy, the organizational infrastructure and processes to carry out the business strategy, and the IT infrastructure and processes to carry out the IT strategy [Coo00]. To align business and IT strategies together Henderson and Venkatraman have introduced their strategic alignment model [HeV99]. Via vertical dimension Figure 2 shows the interplay between the strategy and operational aspects by focusing on the four relations. These relations are indicated by arrows. The model identifies the need to categorize two types of integration between IT and business. The first one - strategic integration - is the link between business strategy and IT strategy and it deals with the capability of IT to support business strategy. The other one - operational integration - deals with the organizational infrastructure and processes and I/S infrastructure and processes and concentrates on the criticality to ensure coherence between the organizational demands and the capability to deliver IT functionality [HeV99].

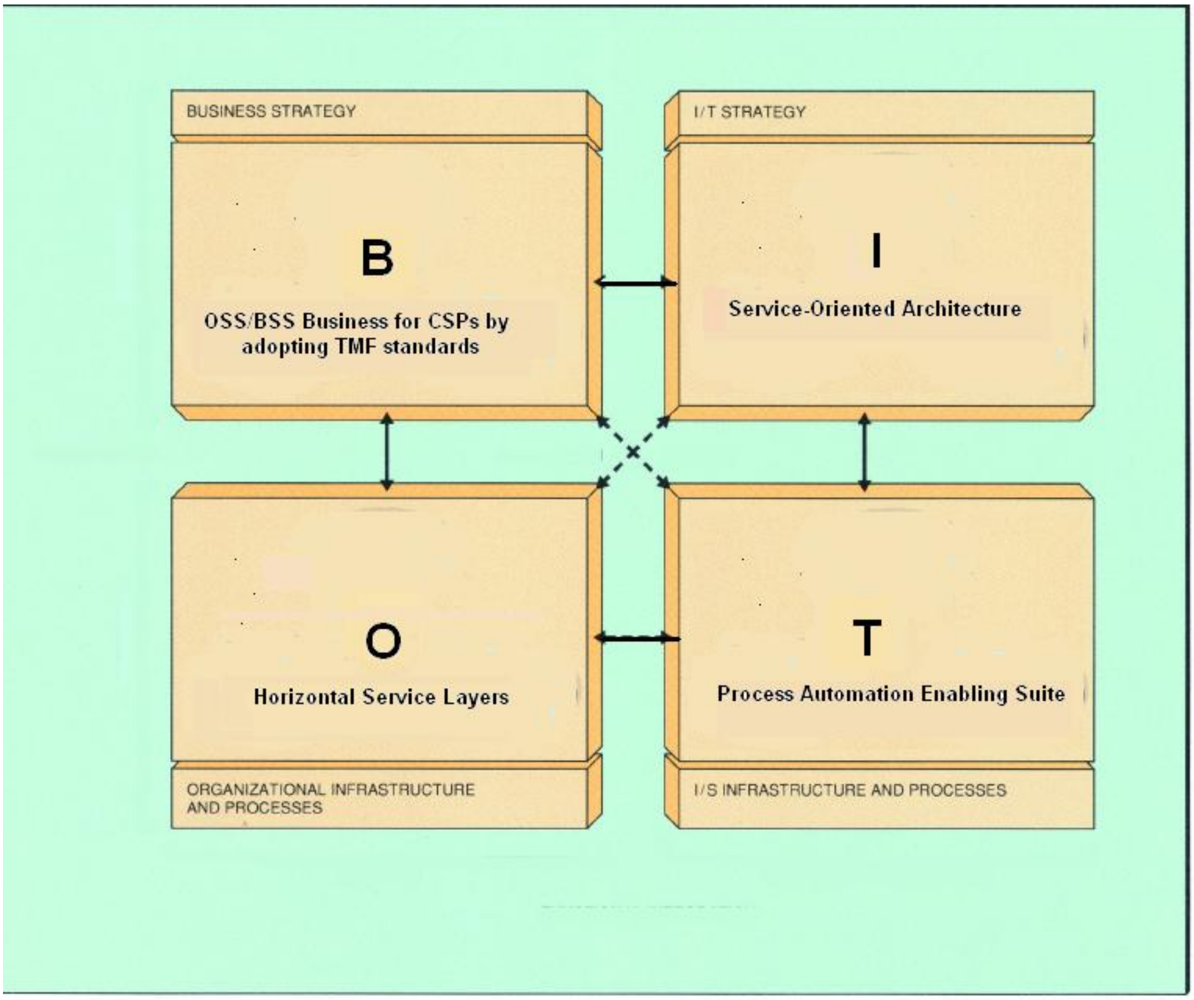

BUSINESS STRATEGY
I/T STRATEGY
B
OSS/BSS Business for CSPs by adopting TMF standards
I
Service-Oriented Architecture
O
Horizontal Service Layers
T
Process Automation Enabling Suite
ORGANIZATIONAL INFRASTRUCTURE AND PROCESSES
I/S INFRASTRUCTURE AND PROCESSES

Figure 2. Strategic alignment model [HeV99].

The framework is all about to assure the coherence of IT assets and to assure the alignment between business and IT. The coherence can be achieved from increased business process maturity and full adoption of Service-orientation as new IT strategy among the organizations. The aspects of dimension covers the spectrum from very business-oriented to very technology-oriented. Table 1 covers the four dimensions of BOIT framework which can be aligned with the strategic alignment model.

Business Strategy → Business (B)	The business aspect describes the business goals of electronic business. As such, it answers the question why a specific business scenario exists or should exist or *what* should be reached. Topics can be leveraged of efficiency levels, access to new markets, and reorientation of interaction with customers. How things are done is not of interest in this aspect. The business strategy in the context of this thesis is evaluated constant but it does not mean that it will stay static forever but should be prepared to continuously make necessary adaptations.
Organizational Infrastructure and Processes → Organization (O)	the organization aspect describes how organizations are structured to achieve the goals defined in the business aspect. Organization structures and business processes are main ingredients here – automated systems are not yet in scope in this aspect.
I/T Strategy → Infrastructure (I)	The infrastructure aspect covers the infrastructure (architecture) of automated information systems required to make the organizations defined in the organization aspect work. As such, it describes how automated systems support the involved organizations.

I/S Infrastructure and Processes **→Technology (T)**	The technology aspect describes the technological realization of the systems of which the architecture is specified in the infrastructure aspect. This aspect covers the concrete ingredients from information and communication technology, including software, languages and protocols.

Table 1. Four aspects to form the BOIT framework [HeV99, Gre06].

A critical lever for achieving dynamic capability is not any specific set of enhanced technological functionality but organizational capabilities to adapt technology to differentiate its services from other vendors. No single IT application could offer a sustained competitive advantage. Rather, advantage is achieved through the capability of an organization continuously to exploit IT functionality. This requires a fundamental change in management level thinking about the role of IT in organizational transformation and reasonable understanding of the IT strategy as its supporting role for business strategy. This is especially important because IT has emerged as an important source of strategic advantage to enterprises [HeV99].

It is equally important to identify the specific role of the management level because the adoption of the Service-oriented strategy requires deep management level involvement and a robust communication plan. The communication plan is used to spread the transformational messages of new strategy to target groups via correct channels. The main goal is not only to achieve awareness of new IT strategy among the employees but get them really to understand and committed to new IT strategy. The aspects that should be taken into consideration when leading the enterprise level transformational process, have been analyzed explicitly in the article written by Kotter [Kot07].

Service-oriented IT strategy should be adopted by synchronizing it with the business strategy, organization infrastructure, and IT technology. The business strategy is evaluated constant in the context of the thesis. Based on this assumption, the business strategy can be ignored when evaluating the best alignment perspective. After the deep alignment analysis, the service level alignment perspective has been chosen as thesis' key alignment strategy. As shown in Figure 3, this alignment model concentrates on to build a professional IT service organization. Therefore, it demands understanding of IT strategy with corresponding understanding of information system infrastructure and its main processes. The IT organization must deploy resources and be active for growing and agile demands received from customer experience management systems.

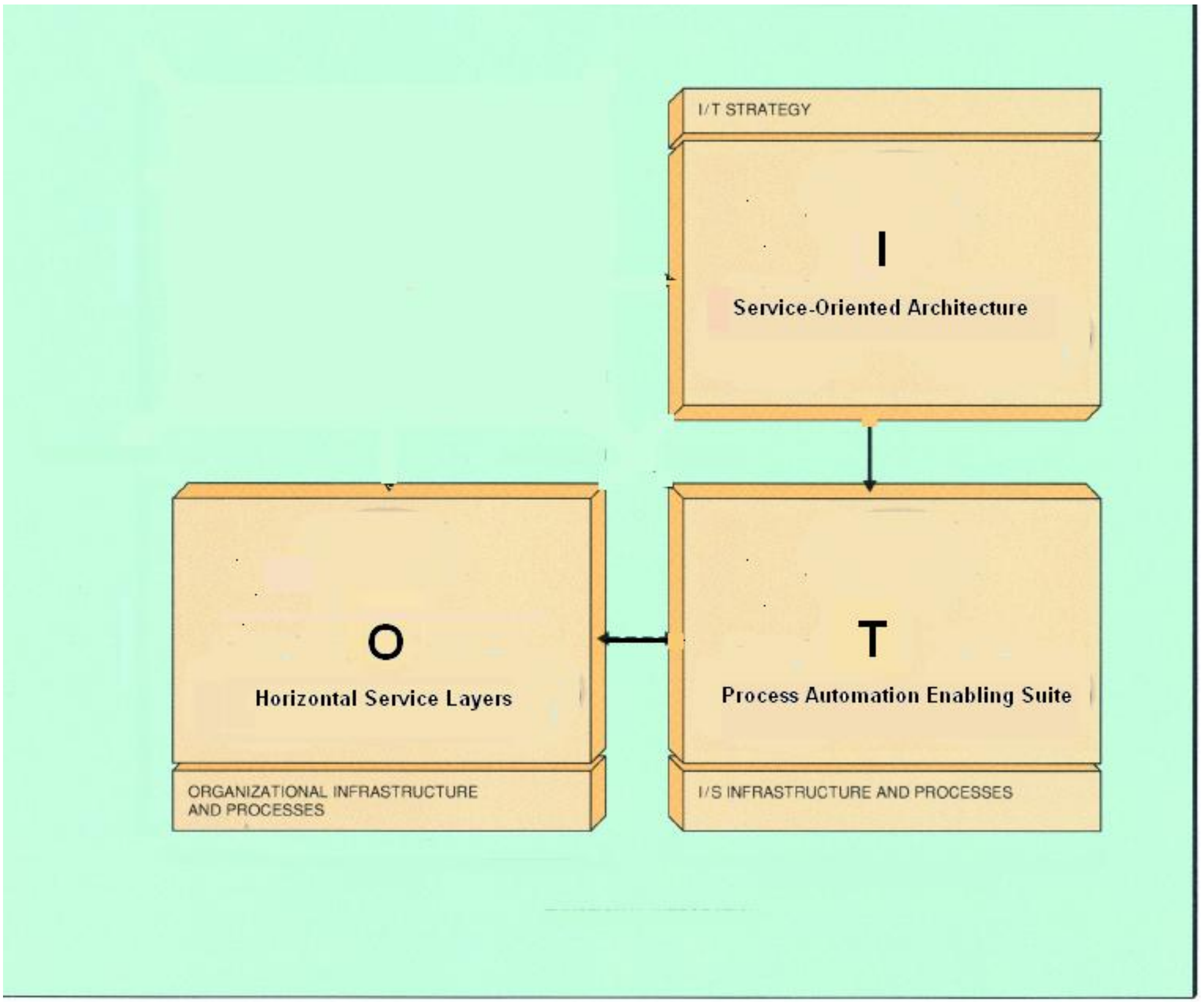

I/T STRATEGY
I
Service-Oriented Architecture
O
Horizontal Service Layers
ORGANIZATIONAL INFRASTRUCTURE AND PROCESSES
T
Process Automation Enabling Suite
I/S INFRASTRUCTURE AND PROCESSES

Figure 3. Service level alignment perspective [HeV99].

The actual IT strategy implementation has to be performed by all business units and product lines across organizations. A top-down strategy works its way from business process modeling workflows down to implementation level, whereas a bottom-up strategy usually involves the use of technical frameworks that guide how to integrate legacy systems [Fet06].

The Service-oriented IT strategy should not only look at the current "as-is" status, neither it should purely align with top-down or bottom-up approach. Instead, both approaches should be taken into the consideration when aiming at the final "to-be" state. The goal can be seen in Figure 4 which composites these two approaches transforming finally to a meet-in-the-middle path. The meet-in-the-middle path ensures that services are constructed and modularized based on real business needs and guarantees that an iterative and agile development lifecycle keeps on rolling. At the same time it gives answer to the question: "Which one is first – the business process or services?"

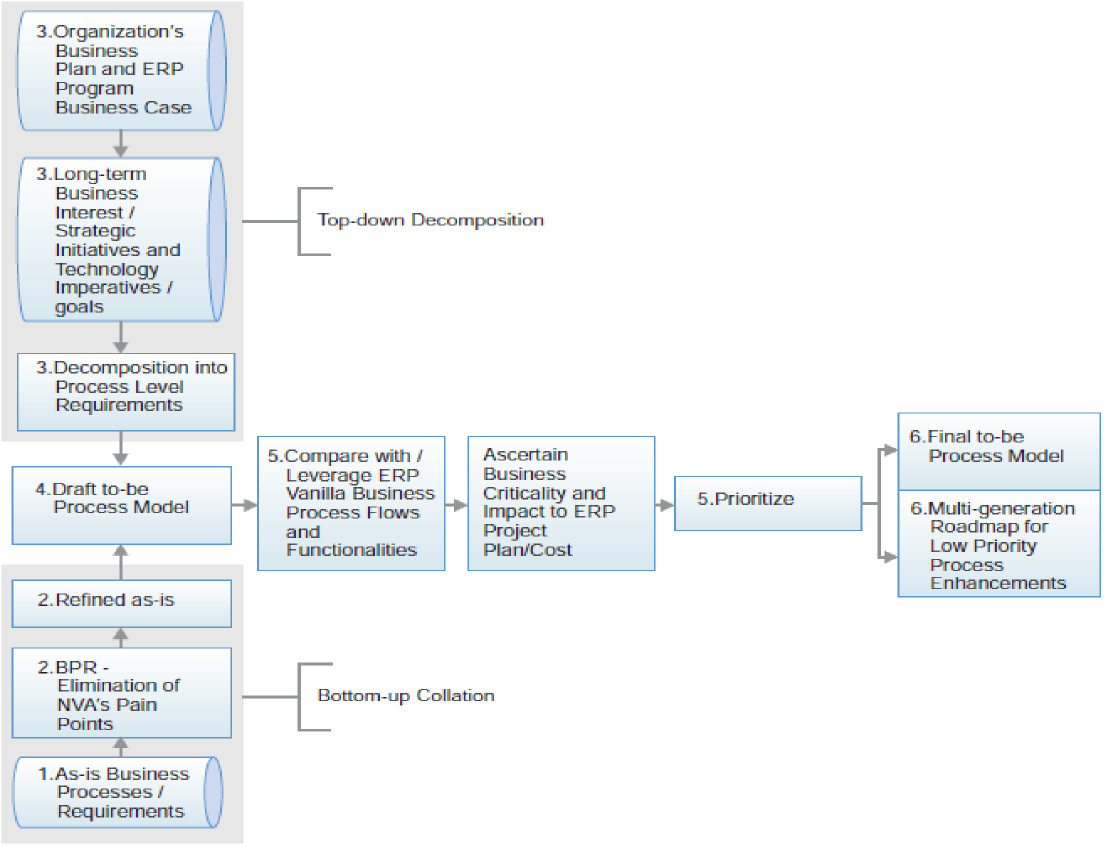

3.Organization's Business Plan and ERP Program Business Case
3.Long-term Business Interest / Strategic Initiatives and Technology Imperatives / goals
Top-down Decomposition
3.Decomposition into Process Level Requirements
4.Draft to-be Process Model
5.Compare with / Leverage ERP Vanilla Business Process Flows and Functionalities
Ascertain Business Criticality and Impact to ERP Project Plan/Cost
5.Prioritize
6.Final to-be Process Model
6.Multi-generation Roadmap for Low Priority Process Enhancements
2.Refined as-is
2.BPR - Elimination of NVA's Pain Points
Bottom-up Collation
1.As-is Business Processes / Requirements

Figure 4. The alignment of Top-down and Bottom-up approaches [Mal10].

One major challenge in Service-orientation is to define the actual concept of a service. A consensus has to be achieved between establishing direct connections between business concepts and existing legacy systems. This agreement can only be achieved through a detailed case study of a particular business area, examining its core business processes and the architecture of its existing IT legacy systems [Fet06]. Far from deriving business strategy from IT strategy, or even developing them in parallel, the best option is to develop them as a unified entity [WDM01]. Before the internet, it was not possible to distribute business processes across the world so easily, just like it was not possible to break up businesses into individual business units without having them connected via master database server hierarchy [Sne06]. From this perspective, technology and business challenges complement each others.

The modernization of legacy systems towards Service-oriented architecture has clear potential benefits. However, it is extremely important to choose the appropriate modernization strategy before adapting it. The recommended approaches are replacement and wrapping strategies [ACD10]. The replacement strategy can be chosen for those legacy systems and applications which can be mostly covered with COTS, excluding small amount of customization efforts. Besides replacement, wrapping can be performed for those operations which are not covered by COTS. Wrapping technology provides a new interface to legacy systems, making it easily accessible by others. Wrapping constructs a service definition interface which can easily be invoked by other services.

On the other hand, variability of interfaces suggests that at least one of the interacting subsystems is subject to change, and therefore integrating these subsystems inside a module would transfer the management of the variation from the customer to the software vendor. Therefore, in order to succeed in innovation process, the organization can out-source relatively independent modules, but integrate and manage closely interdependent ones by itself [MTV08].

It makes occasionally sense to retire the application and replace it with COTS or then perform a complete recompilation of the legacy system from scratch. If the business rules of the maintenance agreements are well understood, then it should be realized that legacy systems contain obsolete technologies which are difficult and costly to maintain, especially if the system has reached the end-of-maintenance milestone. The replacement strategy can be chosen if wrapping, redevelopment, and migration strategies generate costs which cannot be justified. Rewriting the application from scratch is not cheap but rather time consuming. However, it offers a great advantage to deliver a customer-specific solution that can be built exactly to response all the organizational needs [ACD10].

It might be easier to start the transformational activities from the business drivers and top-down approach because models of the existing and future processes are crucial inputs for business and IT alignment. The top-down development strategy is evaluated to be more interoperable than the bottom-up approach because it is avoiding product-specific models and mainly concentrating on interfaces and message definitions. This path should lead to a higher likelihood of interoperability. The possible negative impact of bottom-up approach can lead to poor business service abstractions since the designing decisions are dictated by the existing IT environment restrictions, rather than based on pure business needs. In some circumstances it may also lead to insufficient, non-functional requirement characteristics [PTR05]. When these restrictions is recognized, SOA combined with the meet-in-the-middle strategy should generate a success.

3 Process Integration through Service-Oriented Models

Constantly changing business needs and rapidly changing IT platforms require independence and generalization of process models [WDM01]. SOA serves these aspects providing platform independent services to construct business processes. There are constant needs to integrate process models together but integration without sufficient knowledge can lead to project delays and underestimated extra costs. Service-oriented business process models offer the solution for this challenge.

This chapter recommends how the organizational transformation can be conducted. Furthermore, the chapter introduces TMF's NGOSS which can be exploited to standardize and consolidate CSPs' common business processes. Finally, this chapter presents benefits CSPs will achieve by adopting Service-oriented business process models as part of their daily business operations which can be offered as comprehensive OSS/BSS Service as a Solution.

3.1 From Vertical Product Silos to Horizontal Service Layers

Vertical industry means technology area which has a clear specialization and limited transferability of skills and knowledge outside its own silo [LHF11]. The terms vertical silo, product specific and industry specific software are used to describe software, which cannot be easily re-deployed in any other vertical industry than within its original silo, as opposed to horizontally layered and generally purposed software. Horizontal approach refers to a development technique in software systems where it becomes possible to offer the same software for multiple customers within the same industry, across technological and industrial boundaries. However, it can also define a development technique where a software vendor expands its offering to new market segments within a vertical software market or enters totally new vertical software market [FTL09].

Historically, vertical industry has played a dominant role in the telecommunication area and industry specific software is not developed as OSS or delivered as a service either. OSS adoption has been prevented by specificity of processes, applications and their interfaces. OSS software has traditionally been vertically integrated and relatively consolidated compared to other IT business industries. The key reason for vertical disintegration seems to be the desire to gain efficiency, innovativeness and flexibility in software manufacturing. The outsourcing of non-core business operations also seems to have driven towards the vertical disintegration.

The key part of the strategic change is to overcome the traditional industry-specific vertical organization silos and to move towards service centric horizontal layers. Historically, CSPs have been forced to update their legacy systems to cope with every new operation. This have led to multiplicity of incompatible systems, enlarging the complexity of the IT systems. The reward for reducing the number of processes and systems and the simplification of the service introduction process represents the significant reduction in operational costs. Offering attractive services and stimulating service usage is currently coupled with mastering a multitude of workflows and legacy interdependencies. It is not economically justifiable to have two completely separate systems only with small differences in the functional level. Shifting towards the service centric approach brings significant competitive advantages to CSPs in terms of improved business flexibility and bringing new services faster to market as seen in Figure 5. Furthermore, it optimizes operational efficiency and reduces costs due to reduced number of processes, systems and interfaces [Don09].

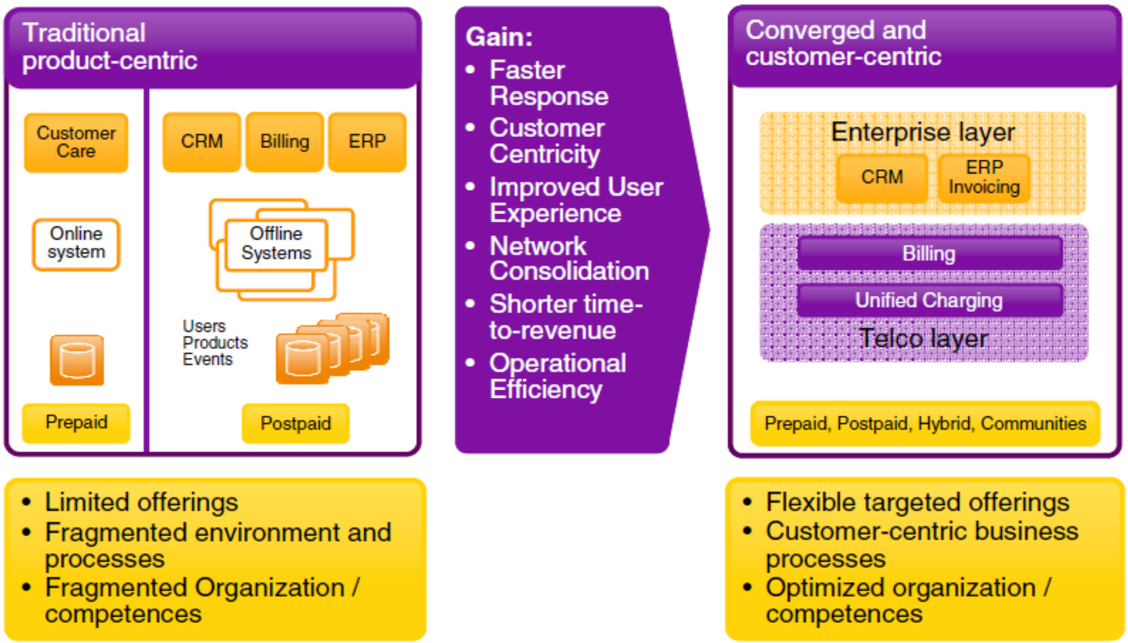

Traditional product-centric
Customer Care
CRM
Billing
ERP
Online system
Offline Systems
Users
Products
Events
Prepaid
Postpaid
Gain:
• Faster Response
• Customer Centricity
• Improved User Experience
• Network Consolidation
• Shorter time-to-revenue
• Operational Efficiency
Converged and customer-centric
Enterprise layer
CRM
ERP Invoicing
Billing
Unified Charging
Telco layer
Prepaid, Postpaid, Hybrid, Communities
• Limited offerings
• Fragmented environment and processes
• Fragmented Organization / competences
• Flexible targeted offerings
• Customer-centric business processes
• Optimized organization / competences

Figure 5. From product-centric silos to customer-centric service layers [Don09].

Systems in the telecommunication industry are required to interact with telecommunication networks. These systems are also required to support processes specific to the industry. Some of the OSS/BSS operations have recently been unified across the CSP silos. For example, CRM systems could be the same for all the provided services [TyF08]. Thus, new CRM service layer can be horizontally integrated to support OSS/BSS interoperability.

The importance to understand the need of new information technology to support new business strategy were highlighted by Cooper et al. in their research [Coo00]. However, it is not the technology alone that leads to significant benefits. An old-fashioned organization plus new technology usually equals a costly old organization. Therefore, the organizational change management process has to be performed in parallel with the technology change management process.

In organizations with strong process focus, top-management prioritizes and supports process initiatives and also realizes the importance to use real-time process data for fast decision-making. Organizations that embrace process-centric strategy should focus on end-to-end processes improvement, rather than settling for departmental change. Actually, the organizational change management process does not necessarily mean to reconstruct all the business units and departments, but rather let them to adopt new strategy. Process-centric organizations can quickly realize that processes may change over time, thus those need to be constructed in a way that allows easy, constant and dynamic change [BEA08].

A vital part of the new service strategy is also to create a flexible and agile organization. By understanding customers and having accurate service profitability knowledge, the company can phase out or modify unprofitable services and design new alternatives for customers. In other words, the organization must establish a sense of urgency. Every organization needs go through the SWOT-analysis to figure out the Strengths, Weaknesses, Opportunities and Threads that change management will bring in front of them [Coo00]. The reason why it is recommended to perform the SWOT-analysis is that individuals and organizations have a great tendency for ignoring disturbing evidences and simply continuing to act along established patterns when they lack a plausible way to get out of a bad situation. This kind of behaviour can also be called as self-deception.

Enterprises have to break down their vertical organization silos and to expose applications to inter-operate with each other and to compose business processes spanning several organizations within the enterprise through horizontally layered system integration by means of SOA. Traditionally, applications were built without integration as a goal which led to monolithic and silo styled applications. Besides, IT is not yet respected as a core business and services are still not part of the horizontal business strategy [JeD09]. This lack has to realized and corrected in parallel.

Figure 6 presents the overall idea of the transformational process. In the figure vertical silos are transformed to horizontal layers. On the bottom layer there are products which offer applications. Applications, in turn, provide operations in the form of services. Finally, services constitute processes, and the processes are the cornerstone of the daily business.

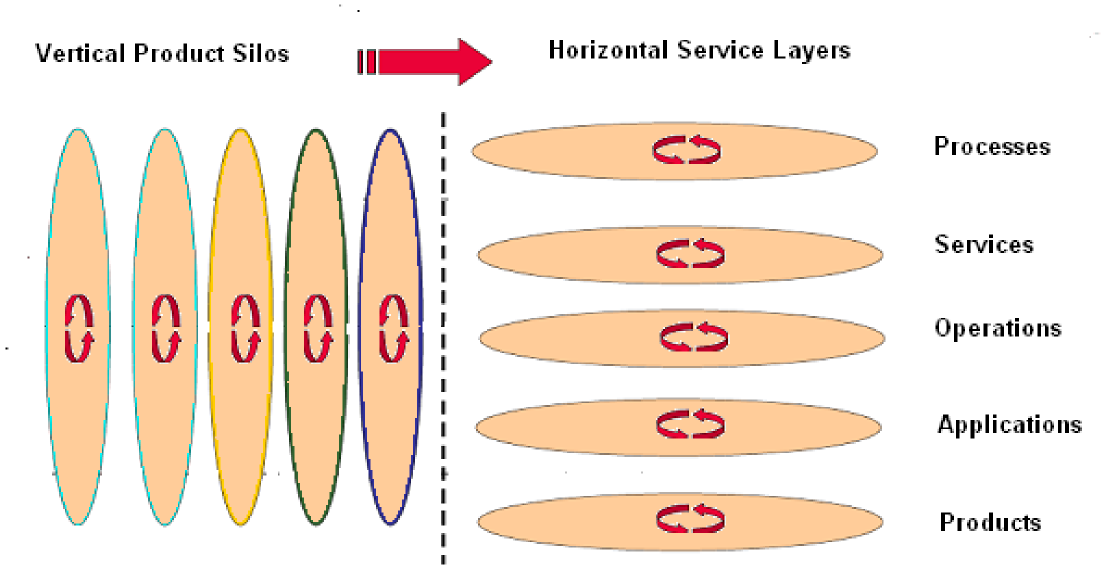


Figure 6. From vertical products silos towards horizontal service layers.

The market evolution of the vertical software industry has been investigated by Tyrväinen and Frank and they have estimated that at the beginning of a system’s lifecycle, software is mainly designed by in-house resources to automate business processes as part of their R&D departments activities. Along the software evolution, the amount of specialized vendors increases as the target and CSP’ money spending shifts from in-house labour to use of IT service providers and further to purchase of COTS. During the mature software phases, use of open source software cuts some market volume from software products, while use of external IT consultancy is still visible in the volume of a vertical software market [TyF08].

The strategic change can be considered as vertical disintegration of the telecommunication operational industry [FTL09]. The new business strategy has to be adopted which means that a core-product has to be transformed to horizontal service layer mode and on top of that vertical customization efforts and extra value services can be tailored just for customer specific needs. Tyrväinen and Frank have also estimated that the software vendors operating on market segments with higher volumes of in-house software development, are on an average larger than others [TyF08]. Same vendors also support a larger scale of network technologies than smaller competitors operating on product specific market segments. The number of COTS the software vendor offers seems to correlate with the total market segment volume.

3.2 TeleManagement Forum’s Initiative - NGOSS

CSPs have recently started to classify their operations. TeleManagement Forum is actively promoting their NGOSS for defining standardized operations of telecommunication business processes [LAG07]. The modularization and standardization is the result of CSPs’ enthusiasm to exploit the economies of scale arising from external providers supplying services for CSPs’ common needs. Traditionally, all the operations for providing services were conducted within CSPs and vertically integrated conforming with organization silos. CSPs also developed their own operations and applications. But lately the situation has significantly changed. The markets have been liberalized and regulations have been rescinded [FTL09].

Changes in business reflect most directly on changes in business processes. Business process management technology with its ability to quickly automate business processes offers a core mechanism for enabling business flexibility and adaptability [Mah06]. Applications constitute the sketch of operations, operations constitute the sketch of a processes and processes constitute cornerstones for the daily business. TMF's strategy is to align them together using their own integration framework initiative - NGOSS. It provides the standard guidelines for effective business operations, enabling CSPs to assess and improve performance by using SOA for integration.

NGOSS has received lot of attention because CSPs have recognized it as core to their business instead of abstract set of other standards. More recently, NGOSS has been extended by Customer Experience Management (CEM) operations which nowadays constitutes the third core industry for CSPs together with OSS and BSS. Figure 7 presents the four sub-frameworks which together compose the whole NGOSS. Its main idea is to bring together processes, applications and data with Service-oriented architecture working as the integration base.

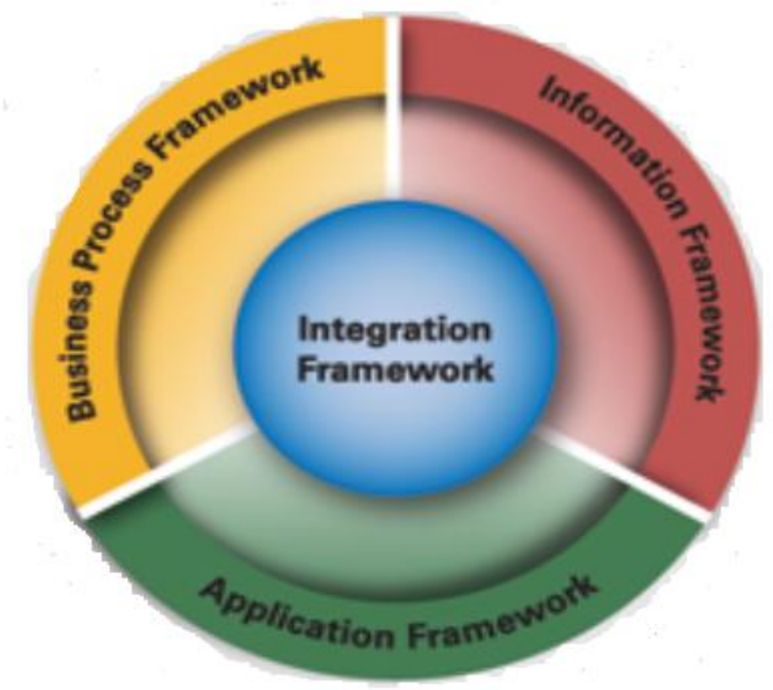


Figure 7. The overall of TMF’s NGOSS [TMF12].

Business Process Framework helps CSPs to move towards a new IT architecture based on standardized business processes. Telecommunications operators have recently modularized plenty of their operations complying with Business Process Framework. Processes have been categorized into four vertical functional groups and into four horizontal functional groups. Vertical groups are operations, fulfilment, assurance and billing. Horizontal groups are customer relationship management, service management and operations, resource management and operations and supplier/partner relationship management.

Information Framework offers a common data model which should enable more agile implementation of projects, enhance quality of services and leverage network data exchange across multi-vendor systems. Information Framework conformant interfaces help operators to avoid time consuming integration efforts due to standardized process and operation semantics. With significant pressure on operational and capital expenditures, CSPs need standard building blocks that enable business agility. To support this business need, Nokia Siemens Networks is working strongly and intently towards Information Framework compliance. The company has recently received TMF's Information Framework certification for its open, process-led OSS/BSS architecture and its business analysis application suite [NSN10a]. This is a concrete step towards the goal to push for a common data model to tear down data silos.

Application Framework provides business requirements and the solution design perspective for applications and helps an enterprise to advance its insights into the design and implementation phases. Application Framework offers a procedure to categorize processes and their associated information into well-known applications. Application Framework provides a common language for communities who specify, design, and deploy operation and business support systems, so that users can understand each other's viewpoints. The Application Framework has been divided into horizontal layers which are consistent with Information Framework and vertical columns which are consistent with Business Process Framework.

Integration Framework is the centralized core element which combines the other three frameworks together as an unique and solid way to manage CSPs' daily business. Integration Framework defines how the processes and information behind these systems can be automated. Integration Framework identifies the dependencies and unifies other frameworks in the Service-oriented fashion. Integration Framework offers a large set of reusable and implementable business services that are based on SOA and which behave like building blocks - each corresponding a standard business operation and dedicated to support the enterprise's service portfolio.

3.4 OSS/BSS Services as a Solution

Services inter-operate both in the organizational level and across company boundaries. The enhanced service integration will enable a rapid entry to OSS/BSS markets. The key strategy is to create the Service-oriented application architecture which enables both inter- and intra-organizational business processes. The multi-layer OSS/BSS solution has to be integrated with multiple networks and services. These networks and services are heterogeneous by the nature, and the multi-layer OSS/BSS solution should be adoptable and flexible enough to be integrated with the business processes of different CSPs. The key strategy is to provide end-to-end productized and verified business process models for fast system integration which improves operational efficiency through reduced complexity and improved data management. It also ensures extensive add-on applications for OSS/BSS functions on the CSP specific demand. The end-to-end process approach starts with business value and develops further to the data architecture and technical solution required to realize these benefits [Mal10]. The end-to-end process approach should be aligned with the meet-in-the-middle development approach.

Service offerings will be of interest to CSPs once they provide cost-advantages achieved through Service-oriented process models. Applications which can be used broadly across organization silos should be implemented first to comply with SOA. Since SOA is a common mode of software provisioning for horizontal applications across industries, it is expected that service offerings are more likely to appear in many OSS/BSS operations, such as in billing, revenue management and service fulfillment areas. When CSPs and vendors continue to standardize and consolidate their business processes and operations, they have better changes to automate and adapt also these operations into their existing service portfolio.

The natural evolution of the horizontal approach evolves mainly through standardization. This reduces obstacles to entry to the software market. The standardization of business processes and operations in the vertical industry enables providing software for many customers and reduces importance of special knowledge of vertical processes. Standard interfaces enable modularization of operations and business support systems and grant more opportunities for pre-integrated business processes. The entry of horizontal software vendors to the operations and business support systems market is likely to have a great impact on the competition and further evolution of business processes. For example, a mediation service could be the same within all the provided business processes. Moreover, some CSPs have specialized, by focusing mostly on operations which are most traditional for telecom operators. This business strategy has resulted in the appearance of both service and network providers [FTL09].

4 Service-Oriented Business

CSPs are going to outsource more and more development and operational activities of the software applications to the software vendors. A new service strategy increases the demand of cost-efficient service offerings. The most prominent approach is the Service-orientation and software vendors are eventually adopting this new service model as part of their core business strategy. The strategy of the Service-oriented business is to provide a service architecture and to offer a large number of simple elementary services whose interfaces are easy to invoke, test and validate.

This chapter concentrates on the Service-oriented architecture and specifies characteristics which services should adapt if they want to inter-operate in a distributed network while composing an end-to-end business process. Furthermore, the chapter introduces Web Services protocols which are standardized protocols for services to collaborate.

4.1 Service-Oriented Architecture

A service is a running instance of an application. Service-oriented architecture is a paradigm for utilizing and organizing distributed services which are controlled by different ownership silos [OAS06]. SOA offers a bunch of principles for building complex, loosely coupled, autonomous and granular applications, as well as paradigms for service orchestration and service composition. SOA is all about building standard interfaces to access different business functions used by various business applications and systems. These business functions can essentially be those ones that are frequently invoked by other applications and systems. SOA provides a way to integrate data and processes across internal and external partners. SOA offers a way to combine business services as unified service offered to the end users [KDK07]. Standard interfaces are constructed for enterprise-level usage and are used by many different organizations and applications spanning a great variety of business processes. This embodiment can be referred as implementation of the enterprise level SOA. In such scenario, the collaboration networks may consist of multiple vendors managing the network together, and of niche vendors filling in the technical gaps of the leading company. SOA is purposed to reduce complexity in application integration with a platform- and programming language independent service layer. SOA can be seen as a giant subroutine library, with the difference that the user must not copy it onto the computer, compile it and link it with his own programs [Sne05].

SOA is not any real product or technology but rather it can be understood as a software engineering concept. SOA is a standard procedure to create connections for enterprise systems. These systems offer their resources as services. The interfaces of those services work transparently. The services offer descriptions how the interfaces are to be connected and used. For example, a mediator that gather data from multiple sources and correlates data packets together as a more solid data packet, can market itself as a mediation service. The interface for a service is provided by an implementation independent manner. Once implemented, the mediation service is available for any users or services that require mediation functionality [Yrj11].

Services are the fundamental elements of SOA and the architecture offers methods bringing and combining needed resources and capabilities loosely together. A service description contains the information to interoperate with the service and describes it in such terms as service inputs, outputs, associated syntax and semantics. The service description also conveys what is accomplished when the service is invoked and all the preconditions required to use the service. In the context of SOA, the importance of Service Level Agreements (SLA) must be highlighted [PaH07]. Service level agreement is a contract between the service customers and the service provider. It is required to define what the service provider is offering to the customer and what are the expectations from the customer. The SLA is commonly a document describing the expectations between the participants. It may cover many qualitative and quantitative aspects, such as maximum performance and latency of the service, service availability and the possible penalty fees of the service failure [Can09]. Besides, Key Performance Indicators (KPI) can be used to monitor the pre-agreed behavior of services.

In general, organizations offer capabilities and resources when they play service providers' role. Parties, who will use service are called as service consumers. The actual service description allows prospective consumers to evaluate whether the service is suitable for their current needs. The key issue for the Service-oriented architecture is to facilitate the manageable growth of large scale enterprise systems and to reduce costs of interoperability. Through inherent capability to scale and evolve, SOA enables a large-scale service portfolio which is adaptable to different needs of specific problem area and process architecture. This can be achieved by adopting the horizontal layer strategy. The infrastructure of SOA encourages also to be more agile and responsive than the architecture which has been built on top of exponential number of point-to-point connections. Therefore, SOA can provide a solid foundation for business flexibility, adaptability and agility. The key feature that differentiates service from traditional software component is self-description. It separates the service specification from the service implementation and thus, supports loose coupling.

Traditionally, each business unit hosts its own local servers and its own team of IT specialists which can be used on demand. This approach grants that productivity of the individual business units can be maximized but in parallel it maximizes also the TCO. The costs of maintaining different solutions for each separate activity will become eventually greater than the costs of sharing common resources. The Service-orientation promises to cure this problem. The service architecture is an attempt to unify the business concept of distributed, autonomous business units, each taking care of different set of processes, with the concept of using common IT resources as services, which can be applied to many concurrent business processes working in parallel [Sne06].

Agile and flexible business process management is the main goal of Service-oriented business. There are two ways to adopt SOA which both can be used in parallel. The first way is to assess the current environment and encapsulate existing applications integration points using Web Services. The second way is tailored for new applications which should be constructed as set of Web Services. The real value through such a manner developed Web Services will be achieved by composing and orchestrating those services into pervasive business processes [JeD09].

Although Service-oriented architecture has the potential to bridge business and technical aspects closer together, their relevant usage still greatly depends on their context. The nature and the scale of business processes and legacy systems should always be taken into the consideration before changing the course towards SOA [Fet06]. However, including a "Support Web Services" check box on the service portfolio is not a sufficient action to adopt the SOA concept. There are huge differences in how software vendors support Web Services technology, and these differences can have dramatic impacts on the ease of deployment, interoperability, maintenance costs and risks analysis [Hil03]. Enterprises have to make a radical strategic change and to perform an enhancement walkthrough for all applications if they want to fully adopt SOA benefits. The long-term business and IT alignment strategy should aim to purchase new applications and systems from software vendors that are committed to provide the state-of-the-art services now and in the future. Enterprises can avoid the integration failures by using application suites and ready-made business process models that software vendors provide. These models provides state-of-the-art operations for the specific business area, such as operations support, billing, service assurance and fulfilment.

4.2 Service Characteristics

A service is an operation that is provided by an application and offered for another peer, client or server. The open question is to consider what is the appropriate level of a single operation that the application should provide. A service can be referred as an instance of the application. The only thing a consumer should know about the service is the service interface definition that specifies the behaviour and contractual agreement which is required to use the service.

Service abstraction is an important aspect in SOA. It separates the service description and the interface logic from the business logic. It is essential that enterprises develop a service layer which can directly interact and to be linked with existing business processes [PeP08]. The service layer is expected to contain various pre-integrated process models and composite services. The service layer is required to serve new business needs which require agile changes to existing business processes. For a large-scale business process transformation, involving multiple processes with complex point-to-point interactions, the process architecture without any service layer becomes very complex to implement and maintain. It is very difficult to get an overall picture how the business process is actually running and how the process flow is orchestrated. From this perspective, the service layer is necessary to be modularized into multiple simpler assets and to provide them as services [PeP08].

Services can be composed into composite services, and thus they can be reused in a context which was not recognized during the implementation phase [PTR05]. The smallest possible granularity grants the maximum flexibility. However, this approach exposes an extremely demanding role for service integration and orchestration. The higher is granularity of a composed system, the lower is the degree of vertical disintegration. The recommendation is to keep the service simple and to avoid unnecessary complexity which makes it demanding to build an appropriate service definition and test it properly. In other words, the interface should be kept as narrow as possible.

The control logic has to be included either in the Web Service or then within the business process. By default, the Web Service should remain untouched whenever possible. All changes should be done at the business process level, by invoking an alternative service, by changing the order of Web Service invocation, or by altering input and output parameters. The aim of the SOA is to replace existing applications altogether by means of fine grained services which the solution architect can compose together as a dynamic application on-demand [Sne05].

Figure 8 presents the composite service which has been broken down into simple service assets which form a small, isolated and distributed application with ready-made operations. This composite service has been built from various simple service assets exiting in various service layers.

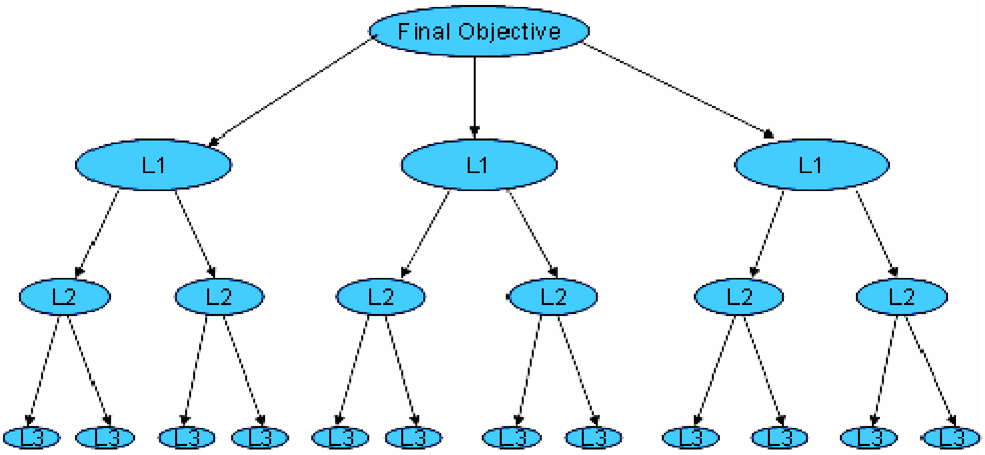


Figure 8. The hierarchy of multi-layered service assets [PeP08].

A single, autonomous computing asset can be referred as an unique service, whereas a composition of multiple computing assets can be referred as business process. In such an architectural environment, any organization can adapt changes flexibly by means of changing the interactions between various services or changing the interface definition of the services [PeP08].

Developers should make an appropriate decision plan and implement a strategy to develop fine-grained services, consequently increasing cohesion, and decreasing complexity and coupling. The granularity of services has an impact on number of attributes. For example, coarse-grained services reduces development efforts simply because it is easier for designers to generalize existing functionality into coarse-grained service interfaces. On the other hand, the creation of coarse-grained services increases service coupling and decreases cohesion. The side effect is lower system quality in terms of maintainability, flexibility, and overall efficiency. Therefore, SOA architects should make a trade-off regarding to expected granularity of services. Coarse-grained services can improve network performance since they require less communication than fine-grained services [PTR05]. However, each message in coarse-grained systems contains more information so the positive performance effect will be received from decreased level of XML document validation and parsing operations. To summarize granularity, it can be seen that more basic elements the component contains, easier it makes to re-use it. Due to this fact, granularity can be seen to trigger vertical disintegration and flourish horizontal integration.

Another important service characteristics is statelessness which means that any information regarding a consumer’s current engagement with the service is not maintained at the service provider's application. A service which does not depend on the state of the service consumer is called a stateless service. This makes all the service consumers equal from the functionality point of view, which means that the business logic of the stateless service is the same, no matter who requests it [Rah10]. In other words, a stateless service is a service that does not keep alive any state information between different service requests. Every local process has to maintain its own state and data. If a Web Service is invoked for the second time, the user cannot expect it to remember the results of the last invocation. The preservation of persistent objects is under the overlying business process' responsibility [Sne05, Sne06].

Services exhibit a complete autonomy from other services, which means that each service is implemented separately from other independent services. Together they create a loosely coupled system. According to the software system design principles, tightly coupled components should be placed in a same module [MTV08]. These modules normally contains some complex business logic hard to separate which makes the interface specification difficult.

In vice versa, loosely coupled components should be placed in separate modules. If services which are composing a comprehensive business process are decoupled using a message oriented middleware, it is extremely challenging to provide transactional integrity. The data replication across different systems provides loose coupling, but generates challenges to maintain synchronization.

4.3 Service Interoperability through Web Services

Web Services offer a technique to describe, discover and transport messages between services. This means that all the organizations of the enterprise are able to use the provided services for their own business purposes. This is a revolutionary change compared to old habits and how software within an enterprise is implemented, delivered and maintained. The simplest way to achieve service interoperability is achieved through Web Services, the standardization that is fully SOA compliant.

Web Services are a popular way to provide underlying technology and protocols for Service-oriented architecture. Web Services allow applications to be accessed from any neutral and transparent environment. A Web Services architecture is the most common way to design the architecture according to the SOA concept. WS are well-suited to abstract the data communication and programming interfaces of these applications and make the information available to other applications by means of XML formatted documents. SOA, using advantages which Web Services provide, makes information sharing across silos transparent [KDK07]. WS provide environment to build a large-scale processing ability from simple, small assets spread across intra-organizational and inter-organizational boundaries in a distributed network [PeP08].

Service-oriented architecture obeying Web Services promises to make applications easier to compose and replace, more flexible, and especially, shorten the timeframe to launch new services to market.

WS comprise of a suite of techniques. Web Serviced Definition Language (WSDL) is tailored for service definition, Universal Definition and Discovery Interface (UDDI) for service registration and discovery and Simple Object Access Protocol (SOAP) for transportation of messages between the service provider and the service consumer [W3C02].

The service interface description between the request and response messages is made by WSDL. Opposite to integration through software code, this declarative description of the operations that services offer, enables services autonomy as well as easy adaptation and evolution of the business logic when business systems need to be updated [Can09]. The deeper is the functionality of the individual Web Service, the more complex is the interface and its definition. The service procedure requires not only more input parameters but it also produces more results, all of which have to be specified by the interface definition. The WS call becomes more and more fragmented, with many layers of nested data and parameter lists, until it resembles a standard program itself. At this point, it is just simpler to code the service itself rather than spending so much efforts to build up a WSDL interface [Sne05]. Complex interfaces require deep black-box testing period for service verification and validation. It may require more efforts to test the WS interface than just program the service itself. In overall, WS should be kept short and simple in order to provide reusable assets which can be composed into business processes with maximal benefits and with minimal efforts.

The other core functionality of WS is a service registry which contains a list of service descriptions. According to these descriptions, another peer, client or server can decide which service fits best to required business needs and choose the service accordingly by invoking that service. A service registry has to be taken into use to provide the place in which the WS definitions can be stored. All the business units and organizations should be informed about the existence of the service registry. A service registry provides a configurable, scalable and secure registry for Web Services. It enables services to be discovered, managed, provisioned and governed across the enterprise. A service registry should enable service providers to publish and advertise their service interface capabilities. It should also allow clients to find and retrieve services that meet their criteria and to consume services they want and let service administrators to control the service visibility, access and lifecycle.

Service registry capabilities support service lifecycle management and SOA environment governance. Service registry must support UDDI discovery enabling registry queries to be fully compliant with WS and to provide functions and flexibility which can help IT groups to institute and enforce standards and best practices in their SOA deployment projects. UDDI provides a framework to classify business, services, and technical details about the services the user may want to expose. Publishing a service into a registry requires detailed understanding of the service type and the data syntax and semantics representing the service in the registry. A registry entry has certain associated properties and these property types are defined when the registry is generated. The user can publish a service into a registry and make it available for other departments and organizations to be discovered and used. Like Walker writes in his article "*Innovation is also about creatively reusing something that already exists and not just creating something new*" [Wal07].

The third core functionality is SOAP which binds the services together. SOAP transports messages which are assembled by wrapping the contents of the header and body variables inside an envelope. The information inside the envelope is an XML formatted document. Behind the WS protocols there is a transportation layer which usually uses HTTP and TCP/IP working as transmission and routing protocol. The standardized transport layers for Web Services are HTTP and HTTPS because they are the only protocols having standard WSDL bindings.

Above WS there have to be agreements how services should interoperate together. It is not enough to let WS to send and receive messages without any reliable communication channel. For that reason, there are important technical aspects to guarantee consistent business transactions and reliable messaging in parallel with security, availability, scalability and performance aspects.

5 Technical Requirements in Middleware Platform

A middleware framework is an architectural pattern which provides extensible models for applications within a silo. Besides, a middleware framework provides the core elements, relationships, structural and dynamic integrity, as well as extension points for modifying the framework for a given application. Applications play very dominant role in today's organizations and most of the crucial business operations are executed by them. Application integration has now become a critical issue for an enterprise [Jab05]. Therefore, the middleware framework is the key element which usually dictates whether management of business processes becomes a success or failure.

This chapter introduces some of the most important technical requirements which SOA integration platform working as middleware framework needs to fulfill, thus it is capable to deploy and manage services. Firstly, some challenges related to data consistency is introduced. Then, Enterprise Service Bus (ESB) is introduced which offers a service provision layer to integrate applications together. Later on, different types of middleware messaging is introduced, and finally the chapter ends with the comparison between Domain Specific Notations (DSN) and general notations.

5.1 Challenges in Business Transactions

WS transactions support thin clients representing autonomous parties whose accessibility cannot be determined beforehand. The area of process and service composition requires a support of the transaction management. Therefore, it has to be possible to compose business process transaction models in a way that all the transactional requirements of each participant can be satisfied [Wan08]. Integration and coordination increase the amount of real time data and this type of interoperability can be referred as business transactional exchange. The presence of SOA should further enhance the benefit by performing real time communication and information delivery easier due to the inherent standard-based interoperability feature. Thus, enterprises with SOA would be in the better position compared to competitors to achieve better transactional performance [KDK07].

The client layer contains end-users, peers and services. Those services exist within or across the enterprise borders. The service layer supports loose coupling and communicates using WS. Interfaces in the service layer do not run the actual application code but they market their interface definitions. Below the service layer, the application layer runs the actual application code. The application layer can be further divided in model, view and controller parts.

Application servers maintain large amounts of data for relational databases and XML databases. These databases require strict concurrency control mechanisms to provide sufficient data consistency. The ideal way is to generate a new persistent data layer which works closely together with applications but stays still independent. This manner decreases the communication costs by keeping data close to its use.

Figure 9 illustrates the hierarchical layout of the layers. The persistence layer provides a non-volatile storage in the nature of relational- or XML database and heterogeneous file systems. The model square left in the application layer should contain the resource manager’s and transaction manager’s roles which are vital for database connectivity.

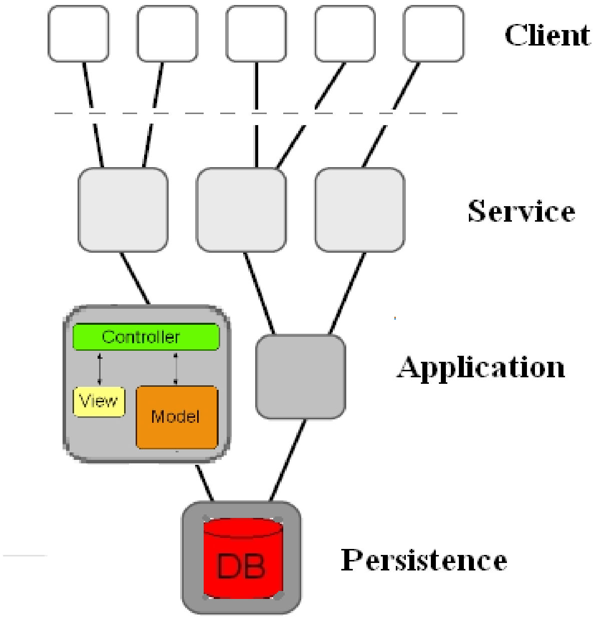


Figure 9. The multi-layer middleware architecture.

Middleware architecture systems are used for transaction processing. These systems control transaction submissions to backend databases and to other transactional subsystems. Strict ACID (Atomicity, Consistency, Isolation, Durability) properties can be loosen to improve application availability, performance and scalability by using these services appropriately [WVG08].

The standard transaction oriented workflow may consist of multiple parallel, serialized short-running requests which are managed globally by the centralized resource manager. Besides, these short-running transactions are managed locally by multiple transaction managers. The heart of the transactions is X/Open XA [Wan08], which defines the standard protocols that allow coordination, commitment, and recovery between transaction managers and resource managers. Hewitt describes pretty well their roles saying *"Transaction managers, coordinates transactions across different resource managers. Typical XA resources are databases, messaging queuinq products such as JMS or WebSphere MQ, mainframe applications, ERP packages, or anything else that can be coordinated with the transaction manager. XA is used to coordinate what is commonly called a two-phase commit (2PC) transaction. The classic example of a 2PC transaction is when two different databases need to be updated atomically. XA proves its value when an application encounters unexpected situations. Intentionally killing a BEA WebLogic Server instance, causing a duplicate key error, or restarting a database simulates situations that can happen in production. If XA has been properly configured, all resources should complete or roll back the same transactions"* [Hew04].

While WS conversations are normally long-living and durable, there are circumstances where it is acceptable to leave them in memory. This can occur in situations where only the last fulfillment step is crucial. Asynchronous communication is vital to support long-living business operations which move back and forth between interoperating services. Between tight and loose coupling, there is a centralized coordinator which mandates the use of certain proprietary mediation technologies over which XML messages flow. The coordination can exist between organizations within or across the enterprise. Therefore, the middleware platform should provide a distributed transactional environment which extends outward from backend databases.

5.2 Enterprise Service Bus

Enterprise Service Bus (ESB) is an integration backbone for Service-oriented applications. ESB enables loose coupling of clients, peers and other services. Main purpose of ESB is decoupling of heterogeneous communication end points in a distributed environment, with the deep focus on integration, system robustness, scalability, flexibility, quality of service and service-level agreement enforcement. ESB increases availability of the service infrastructure which forms a set of standard-based services that are reusable in composite applications and business processes. Applications plug into the ESB and share data with other applications [JeD09].

The ESB governance exposes the traditional enterprise service bus operations extended with scalable Web Services [Bau10]. ESB architecture includes complex message flows in which messages are routed to other services which are composed into larger workflow. Individual service routes, publishes, or calls out next service in the workflow. The reason for a multiple services design is to support modularity of the end-to-end message flow. The individual service in the workflow needs to communicate efficiently and securely. Additionally, it has to allow transactions and transactional behavior to be propagated and to fulfill sufficient ACID properties.

Some of the most important challenges with ESB include difficulty of interfacing and integrating with a complex infrastructure, strict performance requirements and security concerns. In generic SOA environments, the security aspect is solved by its own architectural layer with dedicated manners. It is possible to implement operator's security policy using WS and integrate it to the operator's own security management system. However, if security is hard-coded into applications, the overall security architecture becomes fragmented, hard to manage, error prone and expensive to meet all security policies customers may have.

Service consumers call service providers via ESB. Once the request is received, it is routed via a pipeline through various stages to a business service that is managed by an external application. Separate stages in the pipeline can transform the content of the message, make routing decisions based upon header and message content, and enrich the content of the message. ESB can allow service consumers to use one protocol and interaction style while the business service can use a completely different protocol and interaction style.

Functional features of ESB, expressed in a layered structure, includes three layers. A description and basic operation layer which also can be referred as messaging layer, is concerned with service publication, discovery, selection and binding together with message transport, brokering, routing, validation and transformation. Therefore, it offers multiple messaging options using Web Services.

A composition layer uses versatile notations that allow service integration, preconfiguration, modeling, coordination and orchestration. The composition layer can also take care of measurements gathering even if the actual monitoring and further decisions are performed by the management layer. One major criterion for a powerful service modeling is the ability for easy design of the business operations and maintenance. It is essential, that not only developers can understand how the process is working by looking at the source code level, but the operator should also be able to get an accurate, up-to-date and easily understandable presentation of the process.

The management layer offers features like dynamic service rules and policy configuration and centralized monitoring of the service usage and performance. Embedded service management includes functionalities with unified service interface for maintenance and management. Via service-level agreements, ESB supports configurable monitoring and enforcement of SLAs for messages, operation metrics and message pipelines. These responsibilities are mainly taken care of by business intelligence solution. The rule-based approach is an excellent architectural alternative for applications requiring agility and transparency. They are areas where either frequently changing legal regulations exist, or where companies need to react very quickly to market changes, such as competitor behavior, or areas where the level of customization is especially high due to vertical silos and their specific applications. These applications are considered agile if substantial modifications and customizations can be delivered into production quickly and without programming. An application is considered transparent if an auditor or business analyst can view its code and easily determine that the application precisely implements business policies accordingly. Due to new legislative requirements, all industries have strong transparency requirements nowadays. This trend is expected to continue and strengthen globally.

ESB should offer two separate main components. One component is the mediator and the other one is the service bus. These two components cover different sets of ESB requirements and therefore they can coexist in the integrated system in a complementary manner. The mediator is positioned as an intra-composite mediation component within an application, whereas the service bus is an enterprise-wide ESB, focused on inter-application communication. The mediator provides functionality that can be seen as a service, available within the service bus that provides filtering, routing, and transformation capabilities. This mediation functionality includes also subject-, content-, and itinerary-based routing with declarative routing rules, but may also provide interface versioning, handling transparently routing of requests to a correct version instance, and management of version instances.

The service bus is a standard-based integration platform that combines messaging, data transformation, intelligent routing and is reliably able to connect and coordinate the interaction between different applications. The purpose of the service bus is to make reusable services widely available for users, business processes, and other services. The service bus is designed to connect, mediate, and manage interactions among heterogeneous services, legacy applications, and multiple enterprise service bus instances across the enterprise wide service network.

Services do not need to care about how messages are actually routed to their partner services. They simply receive a message from the ESB and process that message. The ESB gets the message delivered anywhere it needs to go in a reliable way. ESB assumes that services are generally autonomous. A state and availability of a service at a certain moment of time cannot be guaranteed leading to a asynchronous communication mode.

Building SOA based applications has always been a demanding challenge. It is not so easy to decide, what is an appropriate granularity level of a service and when it is reasonable to join or split services. While there are many standards defining how services should be invoked, managed and monitored, in parallel there are quite few standards defining how services should be developed. Recently, there have been standardization efforts focusing on how applications should be constructed and composed in order to build applications based upon reusable assets. The goal of those standardization efforts has aimed to simplify creating SOA based applications.

One of the key aspect of ESB is to centrally manage and control point-to-point connections. Traditionally, the point-to-point integration approach has typically bundled integration and business logic together existing in either or both the end points. The SOA based integration architecture separates the interface integration logic from the business logic. The benefit is significant because it is easier to change the integration logic and also the integration end point system itself [Jed09]. Therefore, maintenance costs have become higher than in the SOA mode due to exponential growth of point-to-point connections but now costs should stay at the constant level. The overall architectural change from point-to-point connectivity towards ESB can be seen in Figure 10.

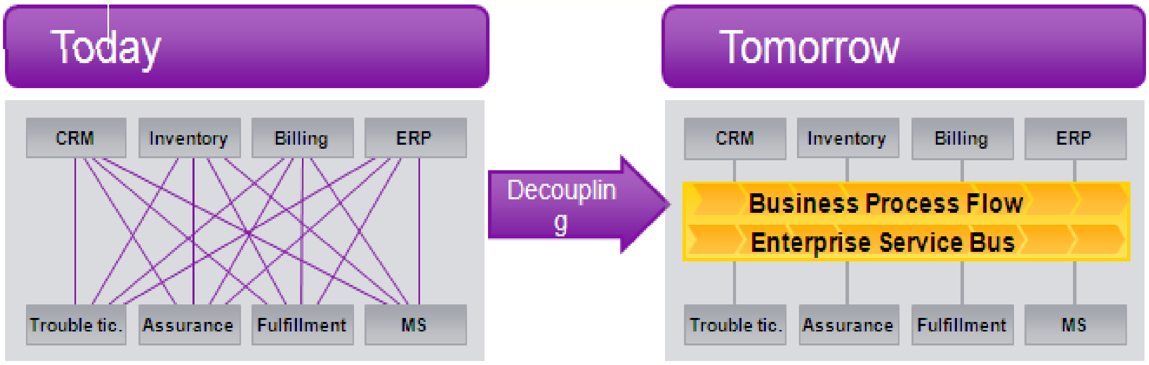


Figure 10. The architectural change from point-to-point connectivity towards the ESB.

ESB enables the integration architecture to model business processes resembling the manual document flow in the organization. The document-based approach for SOA integration is that an XML-formatted document is sent from its source integration end point to ESB where the document is transformed into a ESB message format. The document might be enriched with additional information and then based on the content of the document it is sent to the destination integration end point. What ESB is doing in this chain is transformation, routing and mediation between integration end points. ESB represents significant mediation layer in process integration. With ESB, integrators are adding significant layer to the SOA architecture which adds enhanced availability, scalability, security and performance benefits to the traditional integration architecture [JeD09].

Web Services provide the best-of-proof technique to support SOA approach but the fact is that most of the legacy systems do support neither SOA nor Web Services. While Web Services interfaces are the integral part of the ESB architecture, all applications do not have to be modified to become true Web Services to connect ESB. There are adapter frameworks that provide standards-based access to non-qualified Service-oriented middleware systems. Database adapter, FTP adapter, JMS adapter, and multiple business suite adapters are some examples of adapters leveraging the adapter framework. Adapter frameworks are required for legacy systems and bottom-up approach for non-standardized SOA applications instead of top-down approach. The service integration is achieved through multiple protocols, client API techniques, legacy messaging environments, and third-party application adapters.

5.3 Messaging in Middleware Platforms

While lacking the robust and comprehensive service infrastructure, system architects have been forced to use a variety of middleware techniques to support complex communication, such as object request brokers and remote procedure calls. Information system developers have also hardcoded complex integration logic as many point-to-point connections to Web Services in order to integrate disparate applications and processes. This has inevitably resulted in highly complex services within the enterprises' IT environment.

Message-oriented middleware generally provides a messaging functionality and it have to resolve to external technologies to meet appropriate interoperability and integration requirements [EsJ07]. Besides, it should totally separate the interface logic from the application logic making inter-operating applications fully autonomous and loosely coupled. Message-oriented middleware supports synchronous request/response, asynchronous publish both for one-to-one and one-to-many, as well as asynchronous request-response message delivery techniques.

ESB needs to support multiple service types ranging from pure Web Services to more generic, not pure XML services. The service type is selected by the individual doing the service registration when the business service is created, and it defines protocols that can be used to communicate with the service end point. ESB needs to include three types of services. With SOAP services messages are assembled by wrapping the contents of the header and body variables inside an envelope [W3C00]. If the body variable contains a piece of reference XML, the referenced content is not substituted into the message. With non-SOAP services messages obey XML, but they can be of any type that the service configuration allows, such as JMS. With pure messaging services a request message can be of one data type and a respond message of another data type.

5.4 General Modeling Notations

Generic tools and notations are based on the architectural paradigm which is further based on horizontal division of functionality. Same functionality is solved always with the same tool irrespective of the silo which unifies the integration work. With common notations companies can decrease operational expenditures. For common tool the same resource group can provide support for all solutions irrespective of the silo. One of the key drivers for common tools are reusable assets. Re-usability is one of the biggest saving potential a company has at the application level. General modeling notations ensure that people who know one modeling tool can easily deploy and understand templates from another.

For each Domain Specific Notations (DSN) there are always need to dedicate a separate group of resources. For DSNs there is no re-use between tools that are each supported by their own silo. The core concept of DSN is to have an environment, where the solution can be crafted using concepts and words that are commonly used within that silo and only those. Using a DSN communication service providers will end up having a different development environment for each solution. This will lead that each development environment is serving narrow business area, maybe even only one business operation. Each enterprise has its specific and tailored operations and applications to support daily business. There are large amount of point-to-point connections between these applications. When more operations and applications will be composed together to support enhanced business processes, the amount of point-to-point connections will grow exponentially. This will result to complex, time-consuming and costly maintenance operations. For that reason general modeling notations feel more natural because their concepts are generic in nature.

It is important to possess required capabilities to orchestrate strategic relationships and collaboration networks. A special challenge in the Service-oriented architecture is the large scale and diversity of workflow languages, standards, and engines. It is widely agreed that workflows are essential for business process modelling, re-engineering and automation but there is little consensus about possible features a given workflow management system needs to provide or characteristics about business workflows [Can09]. A modelling notation offers an abstraction of the process definition and modelling from the actual designing tools and systems which are required for the actual implementation. The next step to evolve business process models is to integrate them together with more comprehensive business process modelling and workflow management systems which automate the process execution.

The key for powerful modeling is visualization. Business Process Modeling Notation (BPMN) is a standard which can visualize business processes as graphical workflows [OMG11]. The main approach is designing instead of programming. BPMN is a tailored for process orchestration and it can be used for integrating applications and legacy systems, composing coarse-grained services from fine-grained services, building process centric composite applications and automating business processes. BPMN describes how to compose separate business activities into the complete business process. Each of these activities offers a different service and are invoked via their interfaces. The workflow communicates via these interfaces and defines how to orchestrate applications.

In horizontal partitioning, data sets are partitioned, mapped, and processed by processing nodes in a mutually independent manner whereas in vertical partitioning, data is processed in a pipelined manner on the processing nodes in succession [Can09]. In the horizontal mode, data is delivered to other nodes via XML documents whereas in the vertical mode data is delivered internally within a silo in a pipelined manner. The orchestration depicts a role of the specific participant in the collaborative network whereas choreography depicts the roles of all the participants in the collaborative network.

Composite services have two separate aspects complementing each other. Orchestration of Web Services enables coordination of services by assigning an orchestrator, which is the central controller responsible for invoking and combining workflow's sub-activities. As the opposite, Web Service choreography defines inter- and intra-organizational collaboration of each service to realize the system’s target goal without any central controller. In other words, both ways depict service composition but from different coordination perspectives - centrally and de-centrally.

BPMN can be used to realize orchestration and choreography to achieve interoperability between applications. Figure 11 presents the workflow composition and coordination perspectives. The workflow consists of one pool and the pool has been separated to parallel swim lanes. Each swim lane simulates the role of the specific participant. Dashed lines crossing swim lanes deliver data messages among the participants and lines within the swim lanes delivers data in a pipelined manner.

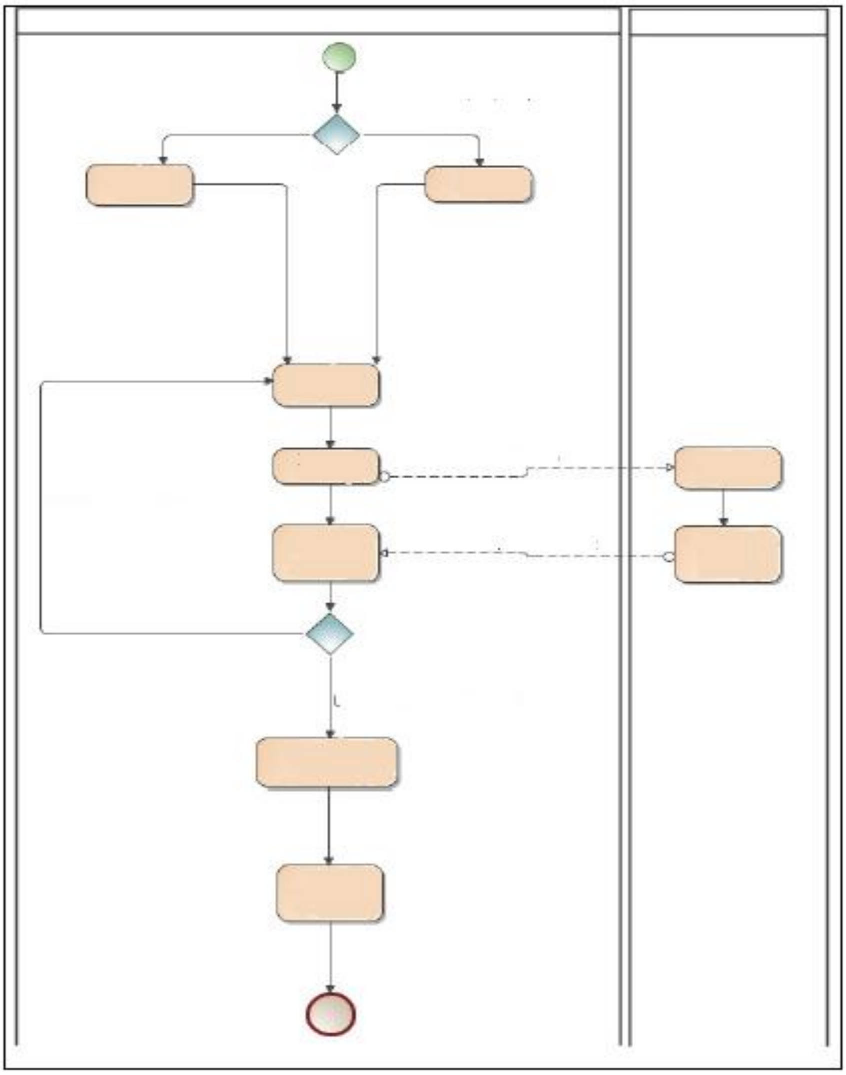


Figure 11. Workflow coordinated horizontally and vertically .

System integrators can design and simulate any type of process simply by dragging and dropping process elements based on the BPMN standard onto role-based swim lanes. A single integrated process model allows developers to make the process executable without requiring a translation between modelling and execution languages.

Once business representatives decide that a process blueprint is ready for sharing, IT personnel can access and edit it within their environment. Rich process definitions get generated from the blueprint promoting rapid and meaningful process automation. The executable process is always in lock step with the business process model. Business users can create and change process models while IT users can view and modify these processes in parallel. The blueprint should also support bidirectional synching enabling both business and IT to work on the same process at the same time. Business level changes can automatically be merged with any changes done by developers to ensure that the implemented process is in-line with the expectations of the business users. Furthermore, IT can make changes to the blueprint that then become visible to business users as proposals for improvement. Further, process metrics collected from the execution engine can be fed into business intelligence to perform analysis which is based on real-time data. This is required to achieve continuous process improvements. This approach enables unparalleled collaboration between business and IT. It will also provide an executable process model on which the whole enterprise can refer. Additionally, it will ensure that the executed process is always the same as the modelled process.

Business Process Execution Language (BPEL) offers a workflow engine, whereas BPMN provides process-modeling integration techniques [OAS07]. BPEL manages all processes and their resources – applications, services, systems and people. BPEL orchestrates the process execution flow, communicates via XML documents and invokes the Web Services over the standardized Web Service interface. BPEL enforces business rules and audits each workflow step to ensure correct process execution. Applications are broken down into services of different granularities and these services are orchestrated on the business process level.

BPEL is purposed also for process orchestration and choreography. Orchestration realizes a process as a partial order of activities under the control of a central conductor whereas choreography realizes a process as an exchange of messages among participants. BPEL establishes links to partners, defines the link types, input parameters to be sent and output parameters to be received, and invokes the Web Services. A WSDL interface can be generated by the BPEL interpreter with the input and output parameters, partner link, the operation name, the messages and variables, the port type and its input and output messages and finally, the SOAP binding description [Sne05].

6 Evaluation of Service-Oriented Business Process Models

Process models offer operational and ready-made business processes. Process models behaving as workflow patterns simulate abstract, commonly recurring business flows, thus helping to reduce costs for developing and integrating enterprise applications [Can09]. This chapter introduces Process Automation Enabling Suite which is Nokia Siemens Networks proprietary and licensed solution offering a large asset of service infrastructure components for building, deploying, and managing comprehensive business processes. Furthermore, the chapter also highlights the importance of business intelligence by means of the agile and flexible management of customer experiences and insights. Finally, some best practises are going through which present results received from the customer experience management and how Service-oriented process models as well as other more common process integration efforts have been adopted in various integration projects.

6.1 NSN Approach - Process Automation Enabling Suite as a Solution

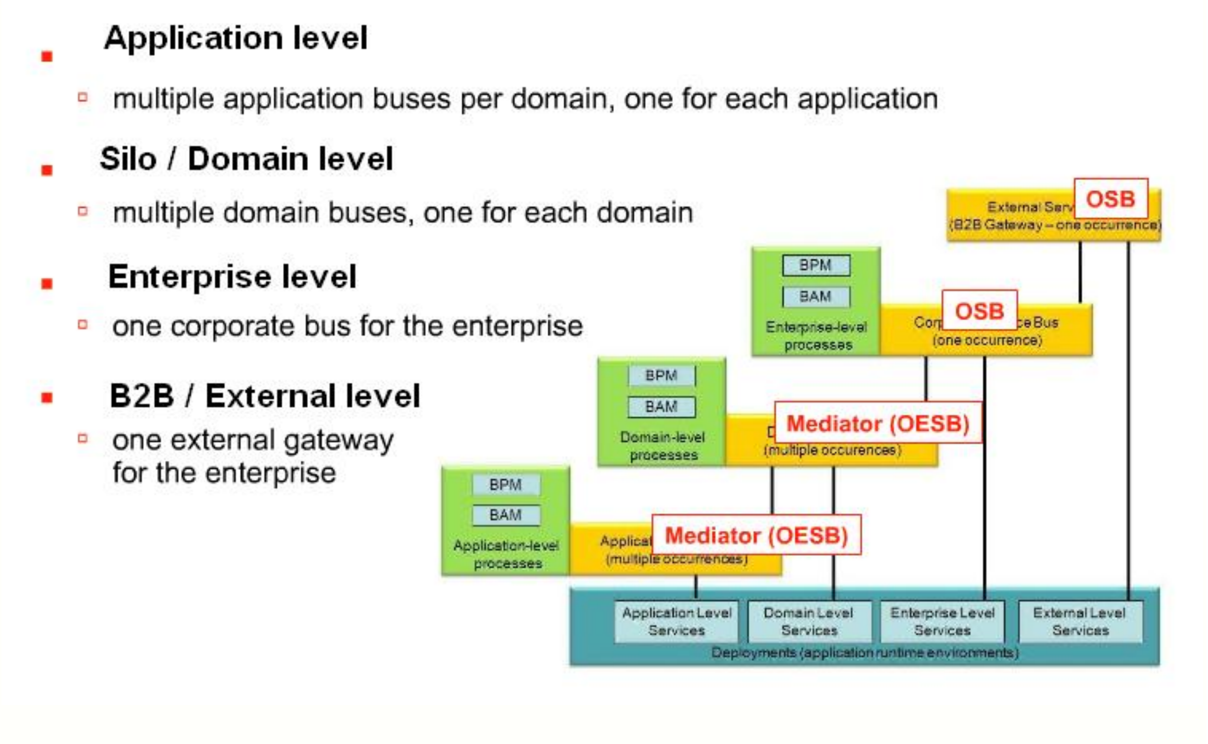
Process Automation Enabling Suite consists of large asset of service infrastructure modules for building, deploying, and managing Service-oriented processes. PES is a Nokia Siemens Networks proprietary and licensed solution. PES enables system developers to set up and manage applications and to orchestrate them as composite services and business processes. PES works as a mediator between business and IT to better co-operate with common process models and languages. With hot-pluggable services, organizations can easily extend and evolve their architecture instead of replacing existing solution or investing in new one.

PES offers a set of tools used to model, implement, execute, monitor and optimize business processes through the process lifecycle. PES is used for enterprise application integration as well as for business process management. PES is mainly intended for persons having a need for basic understanding of the capabilities and concepts of modern SOA based process management toolsets.

PES offers a business process modeling tool for high-level process analysis and representation. This modeling notation is BPMN. PES allows business users to model independently of the implementation or independently of the execution platform's constraints. PES modeling environment has bigger scope, not only related to modeling and design, but also including business analysis to analyze quality, risks, productivity, and costs using state-of-the-art metrics. PES offers process automation and simulation for high-level analysis of real end-to-end operator processes to provide effective cost-benefit analysis. PES supports publishing and sharing processes over the web, hide confidential parts of a process, and to perform process level auditing.

PES can be seen as a management principle as well as a suite of software components including leading BPM technology to bring processes, systems and people together. PES contains an engine for process orchestration. This engine is BPEL. It provides also a graphical user interface for defining the logic. PES offers Enterprise Service Bus for decoupling applications from each other including operations like protocol conversions, data conversion, and content based routing.

Figure 12 presents a multi-layer infrastructure from PES ESB point of view. Application and silo level collaboration is based on Oracle Mediator (OESB). This level of communication can be referred as intra-organizational collaboration. Besides, the external level of collaboration is based on Oracle Service Bus (OSB) and purposed mainly for inter-organizational collaboration.



- **Application level**
  - multiple application buses per domain, one for each application
- **Silo / Domain level**
  - multiple domain buses, one for each domain
- **Enterprise level**
  - one corporate bus for the enterprise
- **B2B / External level**
  - one external gateway for the enterprise

Figure 12. Multi-layer PES ESB architecture [Sch09].

PES ESB fades the operational and technical boundaries. PES ESB improves communications – both intra- and inter-organizational - so that the data in one end-system could automatically appear in the other end-system, without having any human intervention. Besides, PES contains Web services manager for configuring security policy at run time and human workflow engine for adding manual steps to automated processes. PES includes a rules engine for externalizing simple decisions to a human readable database that are likely to be changed at runtime. PES contains a service registry for specifying at run time on what physical node each service is running by providing physical endpoint addresses. PES service registry repository function is used for service governance processes and service lifecycle support. PES provides a configurable, scalable and secure UDDI compliant registry of Web services and related artifacts and enables services to be discovered, managed, provisioned and governed across the enterprise.

PES supports business activity monitoring for real time report generation which offers business users a versatile process monitoring and analytics tool that allows to analyze better business process information in real time. This is done by computing higher-level aggregates from event data. Thresholds can be set, causal relationships between different event types discovered and temporal event patterns analyzed. Once the analysis has been performed, PES provides a real-time, multi-channel deliver of the analyzed information. This enables problem resolution and business process optimization.

The core strategy behind PES is to provide an enterprise wide service ecosystem. Getting solutions based on PES will tremendously trim down the integration costs during solution deployment but still expose a possibility to replace individual applications of the solution with others. The key elements of PES are SOA fashioned Process Integration (PI) middleware and pre-integrated business process models provided as Process Integration Packs (PIP). Furthermore, Business Intelligence (BI) offers methods and tools to monitor and manage operational business processes.

PI framework grafts a new or modified functionality into existing process in a lightweight and easy-to-deploy manner. PI provides a comprehensive, SOA based foundation framework for adaptive application integration. PI is a complete solution to orchestrate loosely coupled and autonomous business processes beyond and across enterprise applications. PI makes the integration of enterprise application seamless. When integrated with such third party applications, an end-to-end solution could be built which can also cover processes like purchase ordering, invoicing, billing and customer lifecycle management.

PI facilitates centralized and flexible coupling of different applications and network elements by centralizing the orchestration of a process flow into the integration layer. Users achieve reduced TCO because the maintenance of integrations is clearly more efficient than what a de-centralized point-to-point scenario could ever offer. The productized SOA fashioned integration reduces the implementation time and time to market for those users who are launching new products and services.

Process integration is an essential part for enterprises to align their processes to receive fully advantages and opportunities provided by SOA. Process integration is related to development of applications which are linked together and provided as pervasive and automated services. Integration of applications may contain not just internal services but integration of the whole process from external sources related to supply chain management or financial services from multiple organizations. It is important that the organization responsible for the integration work chooses right processes for its SOA efforts based on its potential contribution to add business value [TTC06].

PIP provides a collection of the best practice process implementation models across different process areas, like service delivery, billing, assurance, network management, sales, marketing and customer care. PIP offers pre-intergated and packaged set of business processes over the PI middleware framework. These best-of-breed business processes are designed to fulfil CSPs' business requirements. Furthermore, these pre-integrated process models are defined based on TMF's NGOSS. They help the scoping work in any customer project and support collaboration with other vendors and system integrators.

PES will simplify the adaption of converged charging and billing and customer relationship management applications, leveraging CSPs to adapt their tariff models to diverse customer needs and launch new market campaigns based on customer insights and experiences. Furthermore, PES also supports efficient detection of business critical service degradation. A business solution consolidates and analyses usage, service quality and business performance data from all kinds of classes and service types. Besides BSS, in OSS PES helps to provide a comprehensive view of all network elements and services.

PES working as the key element for Nokia Siemens Networks' comprehensive business solution, has been certified against TMF's Information Framework [NSN10a]. Besides this certification, the company has also been rewarded for its automated approach to service management [NSN11]. These awards cite Nokia Siemens Networks for its effective use of TMF's NGOSS which conformance enables the whole telecom industry to reduce TCO while improving agility, operational efficiency and customer satisfaction. Figure 13 presents the overall architecture of NSN comprehensive two-layer business solution. Self Care and Customer Administration rectangles are separate COTS offered with or without PES whereas Business Analytics plays to role of BI.

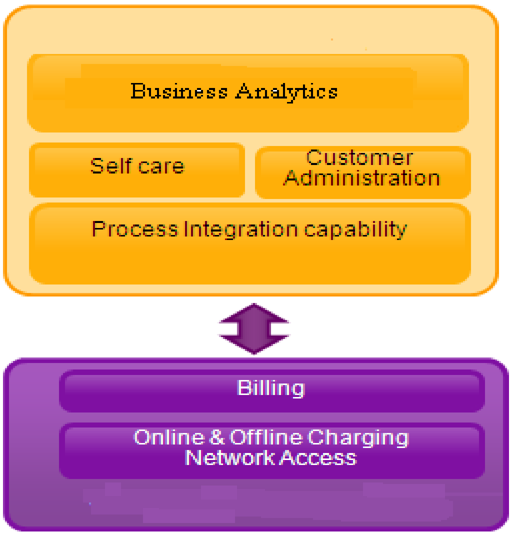


Figure 13. Nokia Siemens Networks' comprehensive business solution for CSPs [Don09].

With horizontal two-layer architecture, CSPs can truly offer the entire service portfolio to any customer across the market segment and overcoming historical silos of diverse systems and processes that previously limited service offerings by separating network service types. These service types are presented in the Table 2.

payment type	network type	data type	charging type	access type
prepaid	mobile	voice	online	content
postpaid	fixed	data	offline	access

Table 2. Network service types.

PES provides an option to seamlessly integrate the OSS/BSS operations layer with any third party enterprise layer applications that may be present in the CSP's IT environment. The pre-integrated process models provide reusable and extensible integration adjusting to any needed changes over the solution life time. This inbuilt solution layer provides economies of scale for further integration with the underlying OSS/BSS operations. Together they reduce the overall number of data models and interface mappings because application designers need to know only two data models - the product specific and the canonical ones. PES is based on use of WS and SOA technologies that provide an IT industry standard framework. From technical, semantical and pragmatical perspectives PES offers fully support for them.

Processes are strongly interrelated with customer's organisation, marketing programs and business management. Customer involvement is hence intensive in most project phases, especially in the scoping phase where customer's existing business processes are mapped and streamlined and the future processes are designed. Management's strong participation, target setting and approvals are required on top of normal technical definition work.

Once the SOA architecture requirements are defined, CSP can start building shared service infrastructure using PES. SOA initiative should be iterative and consist of multiple phases [JeD09]. The first phase builds internal basis for SOA infrastructure. This phase would focus on opportunistic SOA projects and building internal SOA capabilities with the focus on BPMN and BPEL. The second phase externalizes SOA infrastructure. This phase should make SOA capabilities available to the extended enterprise and focus on system integration. It should also standardize inter-enterprise integration and remove all remaining point-to-point integrations. The third phase adds missing components of SOA architecture. Self Care, Customer Administration and Business Analytics – if purchased – can now be integrated as CSP has enough business processes and SOA has been adopted across the organizations.

Business intelligence is a complete solution for building real-time operational views to monitor business processes and services, service levels, and track KPIs from processes and services, with capabilities to take automatic or manually invoked corrective actions. BI enables business users to build interactive, real time views, and proactive alerts. It leverages the latest web technology to deliver a rich, interactive personalized operational views from which real time data and personalized reports are delivered to end-user via a standard web interface. The user can also set personalized alert conditions that can be triggered and delivered to the user through e-mail, short messaging, phone, or via other convenient channel. Users have the ability to take the appropriate corrective actions.

BI is built on a message-based, event-driven architecture, specifically designed for needs of real-time analytics and reporting applications. It is the most appropriate solution that provides real-time visibility in the enterprise, delivering information within seconds. Business intelligence enables business users to improve their decision making process by getting a real-time view of the business events occurring in their enterprise and using this information to improve the efficiency of their business processes. For example, monitoring the purchase ordering process to ensure an up-to-minute view of customer service requests.

The improved visibility enables more effective monitoring of the business processes allowing faster reaction time for changing environments. Due to BI it is not only possible to monitor KPIs, which could affect business relevant SLA in real-time, but it also enables a fast drilldown to the roots of the possible agreement violations.

BI provides versatile process monitoring and analysis tools to access real time data. This is done by computing high-level aggregates from event data. Thresholds can be set, causal relationships between different event types discovered and temporal event patterns analyzed. Once the analysis has been performed, BI provides a real-time multi-channel delivery of analyzed information. This enables problem resolution and business process optimization. BI takes the business level monitoring approach which helps an administrator, who may be confronted by tens or hundreds of events and alarms, all as a result of a single malfunction. BI allows correlating events from multiple sources to detect the root cause.

6.2 PES Case Study - Importance of Customer Experience Management

Customer Experience Management (CEM) is an overall psychological impact of customer interactions with CSP and the services customer use. This includes direct experience from products quality and network services offered as well as indirect experience from brand reputation, user experiences, and information collected from the multiple sources on the web. Tough competition and unpredictable customer demands have made it extremely important to understand customer needs and perceptions, and to build stronger and more profitable customer relationships [Pok11]. CEM provides a limited suite of Business Intelligence purely concentrating on mobile subscribers' insights and experiences. From PES point of view, CEM's main responsibility is to provide real-time information for it so that business analysts and technical architects can quickly react on changed business needs and create new service offerings accordingly. In other words, output results of CEM are input arguments for PES.

CEM has become within the last few years a major issue within the telecommunication industry. Most of the CSPs tried to compete many years with their customers only by reducing the prices. Whilst all offers became about the same, customers made their purchase decisions based on the price. Meanwhile it became clear that in order to increase profitability, it is also crucial to understand the customer behavior, their wishes and expectations and offer services that are valuable for them [PoK11]. Thus, the key success factor and the business strategy must be centralized towards the customer orientation instead of the network centralized strategy. Besides customer orientation, the relevant data is often contributed after a customer activity has been performed, failing to provide a valuable real-time view due to its dislocation from the actual flow of the business process. CEM promises to solve both these challenges.

Enterprises continuously struggle with challenges related to information fragmentation in large business organizations. Every big company collects, processes, generates and stores data. Initially, data is stored in binary format and when it is converted to human readable format, it actually transforms into information. Furthermore, when information is further processed and some special particles are be picked up from the information stream and bound sensibly together, the information transforms into knowledge. Finally, when knowledge is further adapted correctly in some special circumstances, it transforms into intelligence as seen in Figure 14 which presents the sphere of understanding consisting of the four elements in which CEM operates.

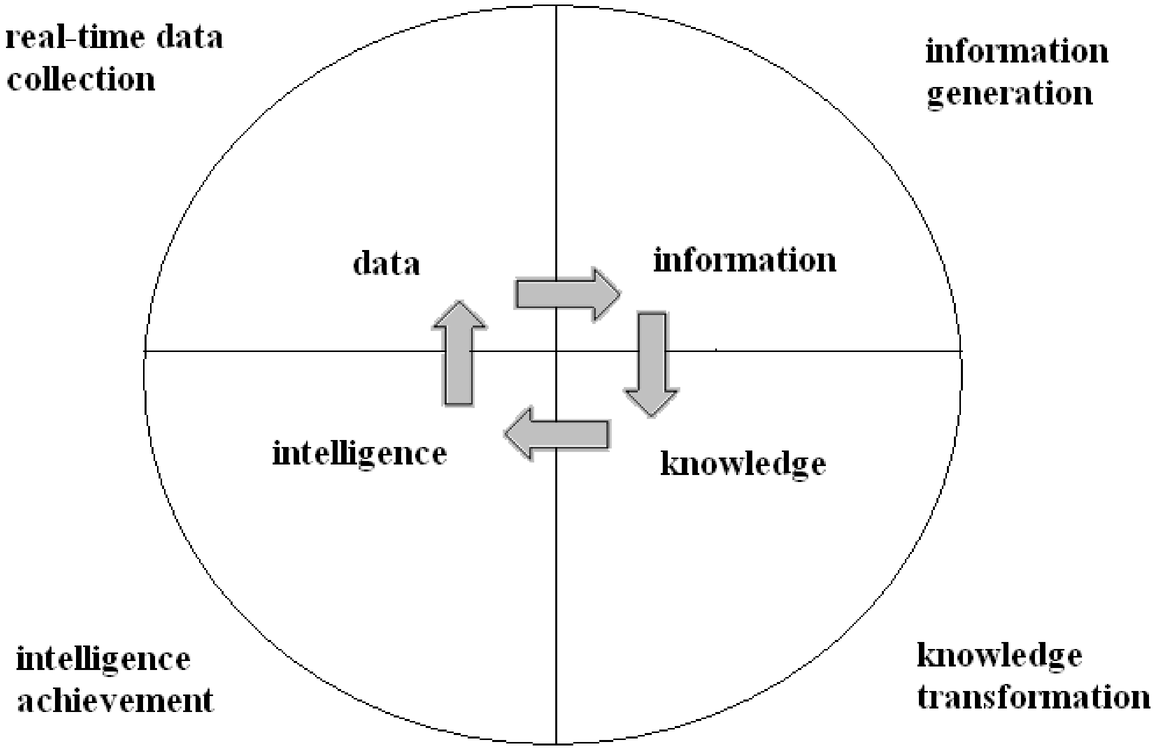


Figure 14. The CEM sphere of understanding.

On the real-time data collection phase, customer and subscriber data is coming from multiple different sources, such as from the network and devices, charging and billing systems, CRM, trouble ticket systems and care systems. Real-time raw data is constantly transformed into customer information, knowledge and insights, which helps CSP to evaluate valuable future opportunities. Customer insights can then be further analyzed to improve business performance [PoK11]. When real-time raw data and historical data patterns are consolidated together for exposure to applications as unique views, data transforms into information.

On the information generation phase, getting the correct data to the right audience in the right format and time, different organizations can efficiently and accurately analyze, profile and segment customers, thus achieving proper insights required for customer centric actions across multiple scenarios. These deep customer analysis gatherings allow a detailed understanding of user's perception, behavior and usage patterns and financial value [PoK11]. During this phase information transforms into knowledge.

The knowledge transformation phase enables customer experience management, efficient data mining and optimized data browsing. When enterprise architects perform a deep analysis of processes and operations in a given silo they can provide an evolution roadmap with recommendations how to implement target solution [PoK11]. When enterprise architects can efficiently adapt the knowledge into special circumstances, knowledge transforms into intelligence. Intelligence is the highest level of understanding. Furthermore, intelligence is the mode of understanding when user can extract, transform and load multiple models and patterns and still further adapt them in some special conditions and forecast the future correctly.

The next study – the PES case study - is based on two large NSN studies and an advisory report from the TM Forum's Management World 2011 conference and exhibition. An acquisition and retention study [NSN10b] was performed by Nokia Siemens Networks in 2010. The purpose of the study was to understand relevant factors which are important for customer acquisition and retention. For this study, more than 20,000 users were interviewed in 17 countries. The study revealed that 40% of high-value customers were dissatisfied with their current operators, and they were likely to switch operators within a year.

Especially, customer's service experiences were one of the top acquisition and retention drivers worldwide and end-users were in search of a better experience. In other words, end-users were not only being lured away by competing offers. Instead, they were being driven away from their CSPs due to dissatisfaction with their existing service. This suggests that an investment in understanding and reducing dissatisfaction can significantly reduce churn in most markets. In advanced markets like

Europe and the US, rather than cost, the actual quality has nowadays become the most important retention driver. There are similar situations in many mature markets and it is just a matter of time when end-users start to evaluate quality more important aspect than pure costs. Self-care options for managing offers, ability to get suitable answers to end-user queries, managing enquiries are all disappointing, which contributes to the lowest level of satisfaction. The study also revealed that the key drivers for retaining customers and to generate customer loyalty and also to win new customers is to empower the customers to satisfy and administer their needs and find answers to their queries around-the-clock and to improve the customer experience by up-lifting the quality of the service.

Besides, Business Evolution Study [NSN10c] was also performed by Nokia Siemens Networks during the same year. This study was carried out via interviews with top opinion leaders and decision makers from 47 CSPs worldwide. The objective of the study was to understand the challenges of communication service providers in terms of improving their customers' service experience. The study revealed that customer insights start to be a ultimate key topic for CSPs and that key challenges to be solved concern the availability of real-time data, data in silos, mismatch of collection versus usage for insights and the structure of internal processes.

The advisory report based on TM Forum's World 2011 exhibition and conference [McG11a] stated that network equipment makers should follow the lead of Nokia Siemens Networks in introducing customer experiment management solution that utilizes multiple components from the equipment makers' portfolios. Besides the importance of customer experiment management, the same conference also highlighted the importance of evaluating new business process models [McG11b]. Furthermore, the same report also stated that pleasing existing customers would eventually be more of a competitive factor than searching and finding new ones. The similar customer behavior analysis has also been made by other researchers [Mal10].

6.3 Best Practises Received from Process Integration

The “Alignment IT to Business Processes” survey among BPM customers revealed that more than two thirds of respondents are connecting SOA and BPM, supporting the strategy that synergies can be received by combining these two approaches. BPM helps to justify and fund SOA investments, whereas SOA enables BPM to scale quickly and effectively. Although BPM can be implemented without SOA, using pre-integrated, reusable services reduces costs and time associated with system integration. BPM offers a higher level abstraction for defining, monitoring and managing businesses processes whilst SOA separates BPM from having details of the underlying systems. Furthermore, three fourths of chief information officers estimated that their company’s IT architecture will be based on SOA, Web services and related technologies within the next five years. The most important drivers for SOA based on the survey can be seen in Figure 15 [BEA08].

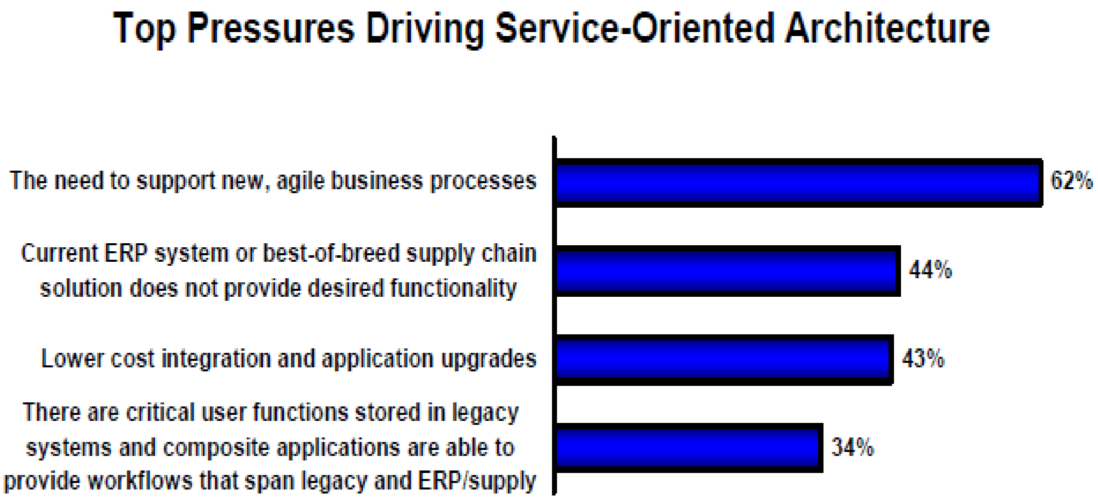


Figure 15. Survey of Aligning IT to Business Processes [BEA08].

Furthermore, the “BPM Lifecycle Assessment” survey revealed that companies which combine SOA and BPM are able to achieve greater reuse, provide better control over business processes than with BPM alone. According to these surveys, the link between SOA and BPM is clear – the support of agile business processes is a primary driver behind many SOA projects. The benefits received from BPM and SOA integration based on the survey can be seen in Figure 16 [BEA08].

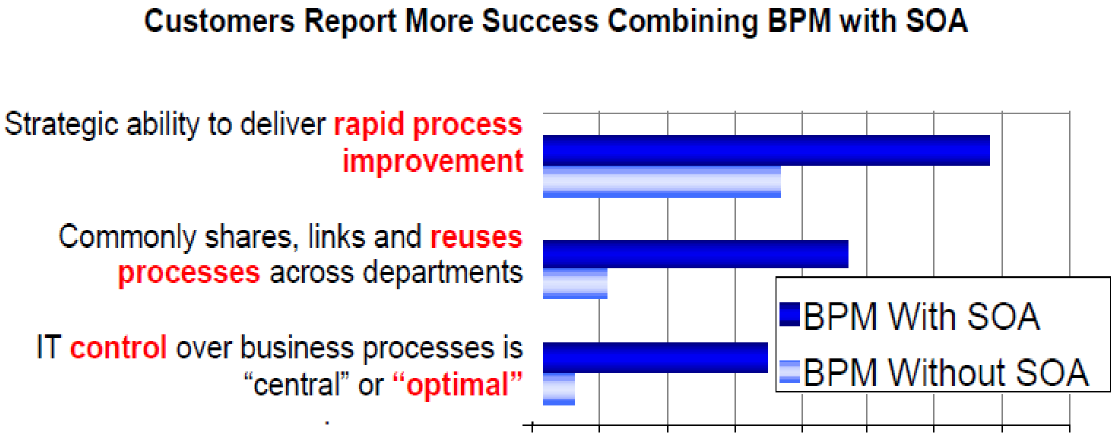


Figure 16. BEA BPM Lifecycle Assessment Survey [BEA08].

Wilson, Daniel and McDonald in their study recommend to leverage the model of best practice service assets embedded in COTS [WDM01]. This will reduce costs and minimize the risk involved in development of bespoken software. They evaluated it important to create an explicit IT strategy which should integrate disparate projects and to align a single organizational view of customers, products and competitors. Furthermore, they also highlighted the importance to prototype new business processes and not just to concentrate on the technology itself. Effective marketing demos normally had detailed and profound implications for internal and external processes and relationships. They required prototyping just as much as the actual IT working behind them. If the testing were ignored or improperly performed, then technological limitations may have constrained necessary business process modifications.

King and Burgess in their research[KiB07] proposed that customers should first be acquired via clear communication of a powerful value proposition, understanding the customer needs and offering value added services. These pro-activities were recognized as key factors that determine the success or failure of companies. Furthermore, the same survey also highlighted the importance of deep social dimension within the organization for a success. These results seemed to highlight the importance of close collaboration with CEM and PES - continuous interaction with the customer to gather data, which can then be analyzed to provide the knowledge required to create more personalized services.

Dezdar and Sulaiman estimated in their survey that minimum customization in parallel with business process re-engineering, top-managerial level support and appropriate project management were the three biggest key success factors to implement ERP system properly [DeS09]. In parallel, Mahajan in his study estimated that SOA implementation achieved remarkable time benefits not only because of the non-modular nature of the solution, but also because of actively performed non-automated testing efforts. The black-box testing was evaluated very suitable testing method for Web Services due to their interface definitions. The same study also stated that by wrapping service interfaces across existing ERP based accounting operations and by integrating them to the ESB, allowed the project to quickly integrate payment engine with the accounting operations system without writing a single line of code [Mah06].

Laukkanen, Sarpola and Hallikainen estimated that the impact on resources in ERP adoption were deeply emphasized if ERP software customization is required. Most companies' were unwilling to get involved with resource constraining and long-lasting implementation, deployment and customization process of ERP system modules. This was one of the most important criterion together with the ERP system module compatibility with the company's current business processes. All the companies irrelevant their sizes kept it important to develop business and integration capabilities, the necessity to replace the old legacy systems and the efficiency of improvement and the reduction of costs, respectively [LSH07].

Simon, Laszlo and Goldschmidt performed a case study [SLG07] to investigate if co-operation of SOA fashioned applications from different vendors can be achieved. Their preliminary assumption was that the software market will remain diversified on the long term, leading to that the biggest challenge for the widespread distribution of SOA is interoperability among applications of different vendors. They had problems to install and configure application servers and SOA tools provided by multiple software vendors. Also the first handbook and sample settings they applied, turned out to be totally useless. Their analysis highlighted that without the appropriate settings and non-functional requirements, seamless interoperability cannot be achieved. Furthermore, they emphasized the necessity for further standardization to enable out-of-box interoperability and portability of SOA tools.

A case study performed by Auer et al. revealed that one of the major advantages which their research participant gained by SOA was the reuse of separate SOA assets leading that all the processes of the company are currently supported by SOA [Aue11]. As a result, the enterprise managed to stabilize IT costs at the constant level, while number of users and business transactions were more than tripled. Besides, another success story was related to a large financial company which sought to build an IT system that was extensible and adapted to changes exposed by different processes which were coupled with other processes [Mal10, p.57]. Agility was achieved by using a layered architecture and by the service concept which finally revealed to be the biggest key success factor acting as front-end service interface for the actual business logic.

Another enterprise operating in the telecommunication industry, evaluated that for them SOA enabled enterprise wide communication in terms of bridging between IT and other departments. Within their enterprise, all continuous IT projects were usually initiated by a change request coming from the business side, which required a static business case containing a five-year TCO calculation. However, the same company also stated that they still do not have an overall strategy behind the company's SOA initiatives to transform the whole architecture towards Service-orientation. Instead, SOA is nowadays primarily used as a connecting layer between different silos with standard software, such as, ERP and CRM systems.

Furthermore, CRM researchers highlighted the importance of knowledge management capabilities, willingness to change processes and share information, ability to understand cultural heterogeneous to develop a customer-oriented organization, and technological readiness together with system integration capability. However, from ERP success point of view, these key success factors were estimated not so important [KiB07].

Interestingly, ERP researches evaluated minimum customization as a key success factors while CRM researches evaluated it not important. The reason for that seems to be the external orientation to customer business and ability to understand cultural heterogeneous which often requires trade-offs between COTS and solution specific implementations. In other words, intra-organizational collaboration should adopt COTS whereas inter-organizational collaboration can also lean to customization efforts if they are estimated necessary. Another especially surprising result was that project team's competence and management skills were estimated important by ERP but not so important by CRM. The possible reason for that could be that even if customization efforts are more easily allowed in CRM projects, CRM operations and systems are more standardized than ERP systems which means that especially high professional project competence and management skills are not required when integrating COTS controlling CRM operations.

Top-management support was evaluated highly important key success factor both for CRM and ERP projects as well as interdepartmental co-operation, communication and data sharing. The support seems to enable project organizations to better understand cultures of the different, geographically distributed organizations. This manner should enable more effective process improvements to be introduced which, in turn, improves knowledge management capabilities and willingness to share data across organizations. In vice versa, the reluctance to share data between organizations will lead to problems with systems integration which, in turn, reduces marketing, sales and local care team members' ability to deliver improved service to customers [KiB08].

7 Conclusion

Service-oriented business process models provide the best way for organizations to enact standardized business process workflows. They exploit re-usable, well-defined service models that are executed by the SOA middleware. Nokia Siemens Networks Process Enabling Suite provides the future-proof foundation for evolution of enterprise systems. The horizontal two-layer architecture with telecommunication and enterprise layers overcomes historical silos of diverse systems and processes and help CSPs to achieve full convergence at network, customer and service levels. The tangible benefits will be achieved by automating manual and overlapping activities, reducing implementation costs due to reusable business models, reducing maintenance costs and leveraging integration efforts for future demands. Furthermore, there are multiple intangible benefits which are quite challenging to measure, such as the ability to co-operate with external partners and increasing customer satisfaction with more business critical information.

The thesis noticed that, with the development of information technology towards a larger business scope and larger scale, the future trend of business process management is correspondingly following a direction to address the need for more services and collaboration in a distributed, heterogeneous and cross-organizational environment. This need is essential especially in an area of rapidly increasing business, which often involves multiple organizations and entities globally and dynamically establishes business relationships and collaborative operations over the internet.

The recommended solution to compose services together working as business processes is SOA which grants an efficient way to create and modify service offerings. In such scenario, the collaboration networks can consist of multiple large OSS/BSS vendors managing business processes together, and of niche vendors filling in the technical gaps of the leading company. Capabilities of being able to customize standard operations and techniques to serve the customer specific needs will be the most important point for the niche vendors to survive.

The scalability of the OSS/BSS service portfolio is an important factor to enable service offerings, since vendors need to have resources to provide solutions covering several business areas. The complete set of OSS/BSS processes can be offered for any CSP as a readily pre-integrated operations and business support systems. Otherwise, the cost benefits from services can be diminished as a result of high costs when integrating diversity software applications together from multiple partners. Enterprises have to break their organization silos and to expose applications to interoperate with others and compose business processes spanning several organizations within and across the enterprise borders. It has to be realized that the SOA adoption will not be any enormous effort. Therefore, enterprises just need to grab on it and perform the transformational process. However, the decision, strategy and transformational leadership have to come from the enterprise's top-management level and transformational messages have to be communicated and channelled correctly. The transformational process will not move towards a new strategy by itself. To alleviate the transformational process the BOIT enterprise framework was introduced. The effective management of business requires a balance among all the four areas of the BOIT framework. The service level alignment model was chosen for the transformational strategy to align business, IT and organizations closer together. This strategy provides the best possible service to organizations by developing the appropriate basis to redesign IT infrastructure while transforming towards SOA.

The new service strategy actually means that all the enterprise-level products and applications has to be transformed towards horizontal layers. On the top, some vertical customization efforts and value added services can be tailored just for customer specific needs if still evaluated necessary. The horizontal approach can be achieved by means of standardizing and consolidating the operations and business support systems and adopting TMF's NGOSS initiative. CSPs play a very important role in the future's business models. Their expectations and needs have an important impact on how services of business process models should be defined. For that reason needs and demands targeted to customer experience management is highly increasing to quickly and accurately provide necessary customer feedback.

Service-oriented process models and the SOA middleware together with customer experience management will become an increasingly popular alternative to extend vendor’s service portfolio. Together, they will dictate whether management of CSP’s business processes becomes a success or a failure. CSPs can be satisfied via highly qualified core products extended via the delivery of tailored applications and services which clearly satisfy customers' expectations. CSPs need to use business intelligence to gather data, which then can be analyzed to provide knowledge required to create more personalized services and to achieve deep and continuous interaction with end-customers. Thus, business processes should be accurately defined by process models so that enterprises can quickly react, construct and provision new services.

It is inevitable that many enterprises need to go through a transformational process from vertical silo systems to horizontally layered systems in which the system integration is enabled by SOA. The more organizations can build a Service-oriented working culture, where everybody focuses on continuous optimization of horizontal business processes, the more successful company’s business and IT alignment will be. Organizations with successful SOA deployments don’t purely concentrate on technology, but see continuous business process management as a critical success factor.

Process Automation Enabling Suite offers the best solution to adopt TMF's NGOSS initiative due to its pre-integrated Process Integration Packages. However, it has to be realized that business processes are not static, leading that the current packages can possibly be outdated in the future. In that scenario, the value of SOA becomes extremely important due to its ability to quickly react on new business demands. Therefore, PES offers an agile way to modify PIPs and adopt them to meet new requirements.

As a conclusion, PES promises the opportunistic future for those who can do this right. On the top of process models and SOA, there are other challenging research areas like the concept of Software as a Service. It is a continuum for services and an alternative way to deliver services to customers without the customers' responsibility to actually host these services by themselves. This exposes a great variety of new alternative business challenges related to SOA and those possibilities should be further investigated in the future.

References

ACD10 Almonaies, A., Cordy J., Dean, T., Legacy Systems Evolution towards Service-Oriented Architecture. School of Computing, Queens University Kingston, Ontario, Canada, 2010, URL http://research.cs.queensu.ca /home/cordy/Papers/ACD_MigToSOA_SOAME10.pdf (31.1.2012).

Aue11 Auer L., et al., Exploratory Case Study Research on SOA Investment Decision Processes in Austria. Springer-Verlag, Lecture Notes in Computer Science, Volume 6741/2011, pp. 329-336, 2011, URL http://www.springerlink.com/content/r150nt35520061n1 (31.1.2012).

BEA08 BEA Systems, The 2008 State of the BPM Market White Paper, 2008. URL http://bpmintro.files.wordpress.com/2008/04/state_of_the_bpm_ market_ white _ paper_2008.pdf (5.9.2012).

Bau10 Baude F., et all., ESB Federation for Large-Scale SOA. SAC'10, March 22-26, 2010, Sierre, Switzerland. URL dl.acm.org/citation.cfm?id= 1774597 (5.1.2013).

Can09 Candan, K., et al., Frontiers in Information and Software as Services. IEEE International Conference on Data Engineering, Lecture Notes in Business Information Processing, Volume 74, Part 5, pp. 283-300, 2011, URL http://www.springerlink.com/content/k372460752333438 (31.1.2012).

Cet07 Cetin S., et al., Legacy Migration to Service-Oriented Computing with Mashups. IEEE International Conference on Software Engineering Advances, 2007, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp? arnumber =4299904 (31.1.2012).

Coo00 Cooper, B., et al., Data Warehousing Supports Corporate Strategy at First American Corporation. MIS Quarterly Volume 24 No. 4, pp. 547-567, 2000, URL http://www.jstor.org/pss/3250947 (31.1.2012).

Cus08 Cusamano, M., The Changing Software Business: Moving from Products to Services. IEEE Computer Society, Volume 41 No. 1, pp. 20-27, 2008, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=444 5598 (31.1.2012).

Des09 Dezdar S., Sulaiman, A., Successful enterprise resource planning implementation : taxonomy of critical factors. Emerald Group Publishing Limited, Industrial Management & Data Systems, Volume 109 No. 8, pp. 1037-1052, 2009, URL http://www.emeraldinsight.com/journals. htm?issn=0263-5577 (31.1.2012).

Don09 Donhefner, D., Unified Charging and Billing Solution. Unified – Next Generation of Charging Systems in Mobile Networks. 2009. Springer-Verlag, IFIP Advances in Information and Communication Technology, Volume 308/2009, pp. 1-7, 2009, URL http://www.springerlink.com/ content /v16740w2k7732176 (31.1.2012).

EsJ07 Estevez, E., Janowski, T., Buiding Dependable Messaging Infrastructure for Electronic Goverment. Proceedings of the Second International Conference on Availability, Reliability and Security, IEEE Computer Society, Wien, Austria, 2007, pp. 948-958, URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4159896 (5.1.2013).

Fet06 Fethi, A., et al., A service-oriented architecture for financial business processes. Springer-Verlag, Information Systems and E-Business Management, Volume 5, No. 2, pp. 185-200, 2006, URL http://www.springerlink.com/content/e0332r574706010u (31.12.2012).

FTL09 Frank, L., Tyrväinen, P., Luoma, E., Analyzing the Current Phase of the OSS/BSS Software Market. Springer-Verlag, Contributions to Management Science, Part 2, pp. 67-88, 2009, URL http://www.springerlink.com/content/l352382q51174183 (31.12.2012).

Gom08 Gomez, P., Service Oriented Architecture as a Strategy for Business Improvement in the Enterprise. Massachusetts Institute of Technology, System Design and Management Program, Massachusetts, USA, 2008, URL http://dspace.mit.edu/handle/1721.1/43122 (31.1.2012).

Gre06 Grefen, P., Structured e-Business. Eindhoven University of Technology, The Netherlands, 2006. URL http://www.student.tue.nl/x/i.bayraktar/docs/EBASReader2009.pdf (31.1.2012).

Hew04 Hewatt, W., XA Transactions. SYS-CON Media, Weblogic: Article, 2004, URL http://weblogic.syscon.com/node/44439 (31.1.2012).

Hal02 Hallikainen, P., et al., Selection of ERP in Finnish SME's. Helsinki School of Economics, Helsinki, Finland 2002, URL http://citeseerx.ist.psu.edu/viewdoc/download?doi= 10.1.1.195.7280&rep=rep1&type=pdf (31.1.2012).

HeV99 Henderson, J. C., Venkatraman, N., Strategic Alignment: Leveraging information technology for transforming organizations. IBM Systems Journal, Volume 32, No. 1, 1999, URL http://dl.acm.org/citation.cfm?id= 166 3583 (31.1.2012).

HiS10 Hipps, B., Sarbiewski, M., The Applications Handbook – A Guide to Mastering the Modern Application Lifecycle. Hewlett-Packard, California, USA, 2010, URL http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA3-1949 ENW.pdf (31.12.2012).

Jab05 Jablonski, S., Processes, Workflows. Web Service Flows: A Reconstruction. Data Management, LNCS, 3551, pp. 201-213, 2005, Springer Berlin Heidelberg, URL http://dl.acm.org/citation.cfm?id=2167728 (5.1.2013).

Jac03 Jacobs, D., Distributed Computing with BEA WebLogic Server. Proceedings of the 2003 CIDR Conference, BEA Systems, California, USA, 2003, URL http://www.cidrdb.org/cidr2003/program/p26.pdf (31.1.2012).

JeD09 Jefic, B., Devost, M., Transforming your organization using Oracle Fusion – is it worth it?. Royal College of Physicians and Surgeons of Canada, 2009, URL http://affinita.ca/files/bjefic_mdevost_wp_1.pdf (31.1.2012).

KDK07 Kumar S., Dakshinamoorthy V., Krishnan M., Does SOA Improve the Supply Chain?. An Empirical Analysis of the Impact of SOA Adoption on Electronic Supply Chain Performance. Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07), Waikoloa, Hawaii 2007, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4076720 (31.1.2012).

KiB07 King, S., Burgess, T., Understanding success and failure in customer relationship management. Elsevier, Industrial Marketing Management, Volume 37, No. 4, pp. 421-431, 2007, URI http://www.sciencedirect.com/science/article/pii/S0019850107000430 (31.1.2012).

Kil04 Killström U.,et al., Opportunities and Threats from Business Model Perspective. The National Institute of Information and Communications Technology, WPMC Archives, 2004, URL http://www.lgalli.it/wp-content/uploads/2008/05/ killstrom-et-al-opportunities-and-threats-from-business-models-perspective.pdf (31.1.2012).

Kot07 Kotter, J., Leading Change: Why Transformation Efforts Fail. Harvard Business Review, 2007, URL http://cerc.stanford.edu/leading _change.PDF (5.1.2013).

Kyt10 Kyte, A., A Framework for the Lifetime Total Cost of Ownership of an Application. Gartner Research, 2010, URL http://www.gartner.com/id= 1331651 (31.1.2012).

LAG07 Liu, X., Azmoodeh, M., Georgalas, N., Specification of Non-functional Requirements for Contract Specification in the NGOSS Framework for Quality Management and Product Evaluation. Proceedings of the Fifth International Workshop on Software Quality, IEEE Computer Society, Washington, USA, 200, URL http://ieeexplore.ieee.org/xpl/ mostRecentIssue.jsp?punumber=4273464 (5.1.2013).

LFP09 Luoma, E., Frank, L., Pulkkinen, M., Overview of Telecom Operator Software Market. Springer-Verlag, Contributions to Management Science, Part 1, pp. 35-42, 2009, URL http://www.springerlink.com/content /v377t 08472604132 (31.1.2012).

LHF11 Luoma, E., Helander, N., Frank., Adoption of Open Source Software and Software-as-a-Service Models in the Communication Industry. Springer-Verlag, Lecture Notes in Business Information Processing, Volume 80, Part 2, pp. 70-84, 2011, URL http://www.springerlink.com/ content/ u18175473250286v/ (31.1.2012).

LMP10 Luoma, E., Mazhelis O., Paakkolanvaara, P., Software-as-a-Service in the Telecommuncation Industry: Problems and Opportunities. Springer-Verlag, Lecture Notes in Business Information Processing, Volume 51, Part 1, Part 4, 138-150, 2010, URL http://www.springerlink.com/content/ q12418p6640x3r54 (31.1.2012).

LSH07 Laukkanen, S., Sarpola, S., Hallikainen, P., Enterprise size matters: objectives and constraints of ERP adoption. Emerald Group Publishing Limited, Journal of Enterprise Information Management, Volume 109, No. 8, pp. 1037-1052, 2007, URL http://www.emeraldinsight.com/ journals.htm?ssn=1741-0398 (31.1.2012).

Mae09 Maes, S., Pragmatic Approaches to True Convergence with or without IMS. Proceedings of the IEEE Network Operations and Management Symposium Workshops, pp. 44-51, 2009, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4509926 (31.1.2012).

McG11a McGarvey, J., 2011 Management World Part II: News and Announcements. Current Analysis, Management World exhibition and conference report, Dublin, May 2011, URL http://www.currentanalysis.com/COMPETE/Login.aspx?ReturnUrl=%2fCOMPETE%2ffrontend%2fReport.aspx%3frid%3d80306&rid=80306 (31.1.2012).

McG11b McGarvey, J., 2011 Management World Part I: Same Message, More Details. CurrentAnalysis, Management World exhibition and conference report, Dublin, May 2011, URL http://www.currentanalysis.com/COMPETE/Login.aspx?ReturnUrl=%2fCOMPETE%2ffrontend%2fReport.aspx%3frid%3d80295&rid=80295 (31.1.2012).

Mah06 Mahajan, R., SOA and the Enterprise – Lessons from the City. Proceedings of the IEEE International Conference on Web Services, pp. 939-944, 2006, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=40321 22 (31.1.2012).

Mal10 Malla, P., et al., Business Platforms for Next-Gen Enterprise Packages. Infosys, SETLabs Briefings, Volume 8, No. 1, 2010, URL http://www.infosys.com/ infosys-labs/publications/Documents/business-platforms-enterprise-services.pdf (31.1.2012).

MTV08 Mazhelis, O., Tyrväinen, P., Viitala, E., Analysing software integration scenarios: the case of telecommunications operations software. World Academic Press, International Journal of Management Science and Engineering Management, Volume 3, No. 3, pp. 200-210, 2010. 2008, URL http://www.worldacademicunion.com/journal/MSEM/msemVol03No03paper04.pdf (31.1.2012).

NSN10a Nokia Siemens Networks, Press Release. Nokia Siemens Networks, 2010, URL http://www.nokiasiemensnetworks.com/newsevents/pressroom/press-releases/nokia-siemens-networks-pushes-for-a-common-data-model-to-tear (31.1.2012).

NSN10b Nokia Siemens Networks, Customer Acquisition & Retention Study. Nokia Siemens Networks, 2010, URL http://www.nokiasiemensnetworks.com/news-events/press-room/ press-releases/nokia-siemens-networks-adds-depth-to-customer-experience-management (31.1.2012).

NSN10c Nokia Siemens Networks, Business Evolution Study. Nokia Siemens Networks, 2010, URL http://ru.nokiasiemensnetworks.com/news-events/press-room/press-releases/ nokia-siemens-networks-adds-depth-to-customer-experience-management (31.1.2012).

NSN11 Nokia Siemens Networks, Press Release. Nokia Siemens Networks, 2011, URL http://www.nokiasiemensnetworks.com/news-events/pressroom /press-releases/nokia-siemens-networks-wins-tm-forum-s-solution-excellence-award-2011(31.1.2012).

OAS06 OASIS, Reference Model for Service Oriented Architecture. Organization for the Advancement of Structured Information Standards, 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf (6.4.2012).

OAS07 OASIS, Web Services Business Process Execution Language Version 2.0. Organization for the Advancement of Structured Information Standards, 2007. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html (4.1.2013).

OMG11 OMG, Business Process Model and Notation, Open Management Group, 2011. http://www.omg.org/spec/BPMN/2.0/PDF (4.1.2013).

Pah07 Papazoglou, M., Heuvel, W-J., Service Oriented Architectures: approaches, technologies, and research issue. The VLDB Journal, Springer-Berlin Heidelberg, 2007, pp. 289-415, URL http://link.springer.com/article/10.1007%2Fs00778-007-0044-3 (5.1. 2013).

PeP08 Perumal, S., Pandey N., Process-based Business Transformations through Service Computing. World Academy of Science, Engineering and Technology, 2008, URL http://www.waset.org/journals/waset/v47/ v47-86.pdf (31.1.2012).

PoK11 Poldrugac, M., Komadina, M., Customer Experience Management. IEEE Proceedings of the 34th International Convention, pp. 346-350, 2011, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5967079 (31. 1.2012).

RaH10 Rabie, Y., Hassanein E., Enhancing Service Design Patterns in ERP Systems using Patterns. Proceedings of the 7th International Conference on Informatics and Systems, pp. 1-5, 2010, URL http://ieee xplore ieee.org/xpls/abs_all.jsp?arnumber=5461825 (31.1.2012).

Ram09 Ramiro, J., CONTRIBUTION TO OPERATION SUPPORT SYSTEMS AND SERVICE MANAGEMENT ARCHITECTURES FOR USER-CENTRIC TELECOMMUNICATIONS SERVICES OVER NEXT GENERATION NETWORKS. Poly-Technical University of Madrid, 2009, URL https://noppa.lut.fi/noppa/opintojaksot/ct30a8201/materiaali/tesis_im_del_alamo_2009.pdf (31.1.2012).

Sch09 Schmutz, G., Oracle Service Bus vs. Oracle Enterprise Service Bus vs. BPEL. Trivadis, 2009, URL http://www.trivadis.com/uploads/tx_cabagdownloadarea/osb_vs_oesb_bpel.pdf (31.1.2012).

SLG07 Simon, B., Laszlo Z., Goldschmidt B., SOA Interoperability, a Case Study. Budapest University of Technology and Economics, Hungary, 2007, URL http://mycite.omikk.bme.hu/doc/37283.pdf (31.1.2012).

Sne05 Sneed, H., Wrapping Legacy Software for Reuse in a SOA. CiteSeerX, 2005, URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi= 10.1.1.87.1642 (31.1.2012).

Sne06 Sneed, H., Migrating to Web Services A Research Framework. Proceedings of the 11th European Conference on Software Maintenance and Reengineering, Amsterdam, the Netherlands, 2006, URL www.cs.vu.nl/csmr2007/workshops/4-%20SneedSOAPaper.pdf (31.1.2012).

Spr00 Sprott, D., Componentizing the Enterprise Application Packages. ACM, Communications of the ACM, Volume 43 No. 4, NY, USA, 2000, URL http://dl.acm.org/citation.cfm?id=332074 (31.1.2012).

TMF12 TeleManagement Forum, TM Forum Frameworx 11.5. Telemanagement Forum, 2012. URL http://www.tmforum.org/Frameworx11/10779/home.html (17.3.2012).

TTC06 Teo, L., Teh, D., Corbitt, B., Service Oriented Architecture (SOA): Implications for Australian University Information Systems Curriculum. Proceedings of the 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, pp. 91-94, 2006, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1624805 (31.1.2012).

TyF08 Tyrväinen P., Frank, L., DIFFUSION OF TELECOM SOFTWARE SERVICES AND PRODUCTS – FROM LEGACY SERVICES TO FOCUSED PRODUCTS. IADIS, International Telecommunications, Networks and Systems, 2008, URL http://jyu.academia.edu/ PasiTyrv%C3%A4inen/PApers/380486/Diffusion_of_Telecom_Software_Services_and_Products_from_Legacy_Services_to_Focused_Products (31.1.2012).

W3C02 Champion, M., et al., Web Services Architecture. World Wide Web Consortium, 2002, URL http://www.w3.org/TR/2002/WD-ws-arch-20021114 (31.1.2012).

Wal07 Walker, L., IBM business transformation enabled by service-oriented architecture. IBM SYSTEMS JOURNAL, Volume 46, No. 4, 2007, URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.2041&rep=rep1&type=pdf (31.1.2012).

Wan08 Wang, T., et al., A survey of the history of transaction management: from flat to grid transactions. Springer Netherlands, Journal of Distributed and Parallel Databases, Volume 23, No. 3, pp. 235-270, 2008, URL http://www.springerlink.com/content/0667826840552700 (31.1.2012).

WDM01 Wilson, H., Daniel, E., McDonald, M., Factors for success in customer relationship management (CRM) systems. Journal of Marketing Man /10.1362/0267257022775918 (31.1.2012).

WVG08 Wang, T., Vonk, J., Grefen, P., A survey of the history of transaction management: from flat to grid transactions. Springer Netherlands, 2008, pages 235-270. URL http://link.springer.com/content/pdf/ 10.1007%2Fs10619-008-7028-1(5.1.2013).

Yrj11 Yrjölä, J., Using Service Oriented Architecture Platform on Cloud Computing Infrastrucure. Tampere University of Technology, 2011, URL http: //URN.fi/URN:NBN:fi:tty-2011082514769 (31.1.2012).