Aleksi Volanen

# Scenario Analysis of Software Component Market in Telecommunication Industry

**Faculty of Electronics, Communications and Automation**

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Espoo 10.5.2010.

**Supervisor:**
Prof. Heikki Hämmäinen

**Instructor:**
PhD Jan Lindström

**Aalto University**
School of Science and Technology

AALTO UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author: Aleksi Volanen

Name of the Thesis: Scenario Analysis of Software Component Market in Telecommunications Industry

Date: 10.5.2010 Language: English Number of pages: 8+82

Faculty of Electronics, Communications and Automation
Communication and Networking
Professorship: Telecommunications Management

Supervisor: Prof. Heikki Hämmäinen

Instructor: PhD Jan Lindström

The share of software is constantly increasing in the telecommunication systems and telecom system vendors do not find all the needed software development resources in-house anymore. Telecom system development has also a constant goal to cut down the development costs and decrease the time-to-market. The recent development of technology platforms used in telecom systems has enabled the increased use of commercial ready-made software components as part of them. This has created a software component business in the telecommunication industry. The software component business has its unique characteristics which distinguish it from traditional software product business.

This thesis examines the technical and economic opportunities and issues component technologies and component business introduces. Moreover the software component market was studied using *Five Forces* analysis to reveal the current market environment and identify the underlying elements of uncertainties. From the set of identified uncertainties the most important factors, which would have the biggest effect on market structure changes, were recognized based on the techno-economic study and industry expert interviews. The analysis resulted in two important uncertain variables; *level of partnering* and *the level of openness in component-based telecom systems*. Combining these two variables resulted in the plausible future scenarios of the software component market in the telecommunication industry.

As a result, four different scenarios were constructed to define the different future markets of software components in the telecommunication industry. The final conclusion was that the software component market is evolving towards horizontal market structure as the utilization of standardized software components increases. Major software companies have however means to slow the horizontalisation and thus increase the revenues collected from component business by engaging in partnerships with major telecom infrastructure providers. Moreover, the structural changes in the software component market were found to happen fairly slowly.

Keywords: Software component, Software component business, OEM, Scenario analysis, Telecommunication industry

| AALTO YLIOPISTO<br>TEKNILLINEN KORKEAKOULU | DIPLOMITYÖN<br>TIIVISTELMÄ |
|---|---|

Tekijä: Aleksi Volanen

Työn nimi: Skenaarioanalyysi ohjelmistokomponenttimarkkinoista tietoliikenneteollisuudessa

Päivämäärä: 10.5.2010 Kieli:Englanti
Sivumäärä: 8+82

Elektroniikan, tietoliikenteen ja automaation tiedekunta
Tietoliikenne- ja tietoverkkotekniikan laitos
Professuuri: Telecommunications Management

Työn valvoja: Prof. Heikki Hämmäinen

Työn ohjaaja: FT Jan Lindström

Ohjelmistojen osuus tietoliikennejärjestelmissä kasvaa jatkuvasti eikä järjestelmävalmistajilla itsellään ole välttämättä kaikkien ohjelmistokehityksen osa-aluiden tietämystä. Järjestelmäkehityksessä on myös jatkuva tarve vähentää kehityskustannuksia ja lyhentää kehitysaikaa. Tietoliikennejärjestelmissä käytettävien teknologia-alustojen kehitys on myös osaltaan mahdollistanut kaupallisten ohjelmistokomponenttien käytön lisääntymisen osana tietoliikennejärjestelmiä. Näin tietoliikenneteollisuuteen on syntynyt ohjelmistokomponenttiliiketoimintaa, joka poikkeaa tavanomaisesta ohjelmistotuoteliiketoiminnasta.

Tämä diplomityö tarkastelee ohjelmistokomponenttien teknisiä ja kaupallisia mahdollisuuksia sekä haasteita. Lisäksi ohjelmistokomponenttimarkkinoita tutkittiin kvalitatiivisella *Viiden Voiman analyysillä,* jotta markkinoiden nykytila sekä markkinoihin vaikuttavat epävarmuudet pystyttiin toteamaan. Todettujen epävarmuuksien joukosta tunnistettiin ryhmä tärkeimpiä muuttujia, joiden vaikutus ohjelmistokomponenttimarkkinoiden muutokseen arvioitiin olevan suurin työn alkuvaiheessa suoritetun teknillistaloudellisen tarkastelun sekä asiantuntijahaastattelujen perusteella. Lopputuloksena määriteltiin kaksi tärkeintä toisista riippumatonta muuttujaa (*yhteistyökumppanuuksien aste* sekä *komponenttipohjaisten tietliikennejärjestelmien avoimuus*). Nämä kaksi muuttujaa yhdessä määrittelevät tulevaisuuden ohjelmistokomponenttimarkkinoiden eri skenaarioita tietoliikenneteollisuuden alalla.

Työn tuloksena esitetään neljä skenaariota, jotka määrittelevät ohjelmistokomponenttimarkkinoiden tulevaisuuden rakenteen. Lopullisena päätelmänä huomattiin, että tietoliikenneteollisuuden ohjelmistokomponenttimarkkinat konsolidoituvat standardeihin perustuvien komponenttien lisääntyneen hyödyntämisen myötä. Suurilla ohjelmistotoimittajilla on kuitenkin mahdollisuus hidastaa tätä kehitystä sekä lisätä komponenttiliiketoiminnan tuottavuutta partneroitumalla suurten tietoliikennejärjestelmätoimittajien kanssa. Lisäksi markkinarakenteiden muutosten havaittiin tapahtuvan melko hitaasti.

Avainsanat: Ohjelmistokomponentti, Ohjelmistokomponenttiliiketoiminta, OEM, Skenaarioanalyysi, Tietoliikenneteollisuus

## Acknowledgement

This master's thesis project has been the single most demanding personal project I have ever undertaken. The process of accomplishing this thesis has been challenging but has definitely learned me a lot.

First, I want to thank Professor Heikki Hämmäinen for supervising this work and giving valuable insights throughout the project. I'm also grateful to Jan Lindström for his willingness to instruct me during the course of my work. I also want to thank Mirva Antila and Joni Lehtomäki who gave the opportunity and support to conduct the major part of this project during the working hours.

I would also like to thank all the interviewees for their comments and information provided for this study.

Finally, I want to show my gratitude towards my family, friends and especially to Hanna - the most important part of my life.

Espoo, May 2010

Aleksi Volanen

# Table of Contents

**Acknowledgement ........ ii**

**Table of Contents ........ iii**

**Abbreviations ........ v**

**List of Figures ........ vi**

**List of Tables ........ vii**

**1 Introduction ........ 1**
1.1 Background ........ 1
1.2 Problem Statement ........ 2
1.3 Objectives and Scope ........ 2
1.4 Used Methods ........ 3
1.5 Outline of the Thesis ........ 4

**2 Theoretical framework ........ 6**
2.1 History of Software Components ........ 6
2.2 Defining Software Components ........ 6
2.3 Advantages of Component-based Production ........ 8
2.4 Different Types of Software Components ........ 9
2.5 Layered Architecture for Component-based System ........ 11
2.6 Component Technologies ........ 13
2.6.1 Windows Communication foundation (WCF) ........ 13
2.6.2 Common Object Request Broker Architecture (CORBA) ........ 15
2.6.3 Web services ........ 17
2.7 The Business of Software ........ 19
2.8 Characteristics of the component business ........ 20
2.9 OEM in Software Component Business ........ 23
2.10 Pricing ........ 24
2.11 Intellectual Property Rights and Contracting ........ 26
2.12 Summary ........ 27

**3 Industry Framework ........ 29**
3.1 Telecommunications Software Industry Structure ........ 29
3.2 Software in Telecommunication Systems ........ 33
3.3 Consolidation of Telecommunications Software Market ........ 37
3.4 Software Component Sourcing Practices ........ 38
3.5 Summary ........ 39

**4 Analysis of Software Component Business Dynamics ........ 41**
4.1 Methods Used in Analysis ........ 41
4.1.1 Porter’s Five Forces ........ 41
4.1.2 Porter’s Scenario Analysis ........ 42
4.1.3 Interview ........ 46
4.1.4 Industry Expert Interviews in This Study ........ 46
4.2 Five Forces Analysis ........ 47
4.2.1 Rivalry Among Existing Competitors ........ 47

4.2.2 Bargaining Power of Buyers ........ 48
4.2.3 Bargaining Power of Suppliers ........ 49
4.2.4 Threat of Substitute Product and Services ........ 49
4.2.5 Threat of New Entrants ........ 50
4.3 Identifying Industry Uncertainties ........ 51
4.4 Determining the most important uncertainties ........ 54
4.5 Causal Factors behind Uncertainties ........ 54
4.6 Dimensioning the Scenario Variables ........ 55
4.7 Constructing Internally Consistent Scenarios ........ 57
4.8 Summary ........ 58

**5 Industry Scenarios of Future Software Component Business .. 60**
5.1 Final Scenarios ........ 60
5.1.1 Niche Market ........ 60
5.1.2 Fragmented Market ........ 62
5.1.3 Vertical Solutions ........ 64
5.1.4 Component Revolution ........ 66
5.2 Strategies for Component Supplier to Confront the Scenarios .... 68
5.3 Choosing the Right Strategy ........ 71
5.4 Summary ........ 73

**6 Conclusions ........ 74**
6.1 Summary of the Findings ........ 74
6.2 Discussion ........ 75
6.3 Areas for Further Study ........ 76

**References ........ 77**

**Appendix A: Industry Expert Interview Structure ........ 81**

## Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| BSS | Business Support System |
| COM | Component Object Model |
| CORBA | Common Object Request Broker Architecture |
| COTS | Commercial of The Shelf |
| CSP | Communication Service Provider |
| DCOM | Distributed Component Object Model |
| ETA | Electronics, Telecommunication and Automation |
| HTTP | Hyper Text Transfer Protocol |
| IDL | Interface Definition Language |
| IEEE | Institute of Electric and Electronic Engineers |
| IP | Internet protocol |
| IPR | Immaterial Property Right |
| IT | Information Technology |
| ITV | Independent Telecom Software Vendor |
| LAN | Local Area Network |
| MOTS | Modifiable of The Shelf |
| MSHP | Major Software & Hardware Provider |
| NDA | Non Disclosure Agreement |
| NEP | Network Equipment Provider |
| OEM | Original Equipment Manufacturer |
| OMG | Object Management Group |
| ORB | Object Request Broker |
| OSI | Open Source Initiative |
| OSS | Operation Support System |
| PSTN | Public Switched Telephone Network |
| RPC | Remote Procedure Call |
| SaaS | Software as a Service |
| SI | System Integrator |
| SOA | Service Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| TCP | Transfer Control Protocol |
| UML | Unified Modeling Language |
| WCF | Windows Communication Foundation |
| WS-* | Web Services specifications |
| WSD | Web Service Description |
| WSDL | Web Services Description Language |
| WS-I | Web Services Interoperability Organization |
| XML | Extensible Markup Language |

## List of Figures

Figure 1: Software component as a part of system solution versus standalone software deployment. Adapted from Helander (2004) .... 3
Figure 2: The structure of this thesis ... 5
Figure 3: OMG's example UML presentation of a component (OMG 2009) ... 7
Figure 4: Layered system architecture (adapted from Jacobson et al. (1997)) ... 11
Figure 5: Component interconnection options in Windows Communication Foundation ... 15
Figure 6: Simplified CORBA architecture ... 16
Figure 7: Web services architecture (W3C 2010) ... 18
Figure 8: Supply chain and pattern of derived demand for industrial products (Chisnall 1995) ... 22
Figure 9: High, Low and Differential Pricing (Shapiro & Varian 1999) . 25
Figure 10: Value network of telecommunications software industry .... 32
Figure 11: World telecom infrastructure market value and shares by segments in 2009 excluding mobile handset market (Market value in million Euros), (Idate 2010) ... 33
Figure 12: High level structure of communication network ... 35
Figure 13: Vertical processes and horizontal layers of CSP's processes (Tyrväinen et al. 2007) ... 36
Figure 14: Porter's five competitive forces (Porter 1980) ... 42
Figure 15: Porter's scenario analysis process (Porter 2004) ... 44
Figure 16: Simplified value-chains in scenario: Niche market ... 61
Figure 17: Simplified value-chains in scenario: Fragmented market ... 63
Figure 18: Simplified value-chain in scenario: Vertical solutions ... 65
Figure 19: Simplified value-chain in scenario: Component revolution . 67

## List of Tables

Table 1: Summary of characteristics of different types of software components ........ 10
Table 2: List of interviewees ........ 47
Table 3: Uncertain elements of structure in telecommunication industry's software market ........ 52
Table 4: Independent scenario variables in telecommunications software business ........ 54
Table 5: Causal factors determining the uncertainties in telecommunication industry’s software component market ........ 55
Table 6: Range of assumptions in telecommunications software industry scenarios ........ 56
Table 7: Level of openness of component-based telecom systems and market structure of telecommunications software component market ........ 58
Table 8: Analysis of scenario: Niche Market ........ 61
Table 9: Analysis of scenario:  Fragmented Market ........ 64
Table 10: Analysis of scenario:  Vertical Solutions ........ 66
Table 11: Analysis of scenario:  Component Revolution ........ 68

# 1 Introduction

## 1.1 Background

Software development is nowadays highly important or even crucial part of product development in many traditional industries e.g. telecommunications- and electronics industries which largely embed software as part of their products. Telecommunication networks for example form huge distributed systems whose different parts increasingly contain specialized software components. Many of these large software-intensive systems are already so complex that original manufacturer is unable to have competence in all fields of software engineering needed. Telecommunications- and electronics industries have mainly used sub-contracting in software development outsourcing. Shift towards increasing the usage of off-the-shelf software components can be seen as software development outsourcing evolution. The needs to cut costs and reduce the time-to-market of the original product are the main drivers behind the increased use of commercial off-the-shelf software components. The evolution of system software e.g. operating systems and employment of layered software architecture in system product design are also helped to enable the use of readymade third party software components in applications. The software component market is relatively new area of business in software industry and has got business and academic communities' attention since late 1990's. Software component market practices can be compared with Original Equipment Manufacturing (OEM) seen in other fields of industry such as car manufacturing. Software component business however has its own special characteristic mainly related to the immaterial nature and technical complexity of software solutions.

Software component business introduces challenges of a new kind in all areas of management of a software company when compared with for example a software vendor who is currently focusing mainly on one-off-license sales. Many different forms of inter-corporation co-operation can be seen in the software component market. These co-operation types differ from project based sub-contracting production to the acquisition of commercial off-the-shelf components and thus software companies and component vendors have many options to position themselves within the value network.

The motivation for this study original derives from a local software division of a multinational information technology corporation. Their recent acquisition of a small Finnish software company has brought a new software product family to their product portfolio. The acquired company has a long track-record of conducting software business in the telecommunication sector and they had been successful in software component business. For acquiring corporation this type of business model is currently not so widely adopted and shows the signs of a potential channel for growth also for other software product families in their portfolio. The fact that big telecommunication companies have a strong presence in Finland and near-by markets increases the interest of the corporation's local country organization to seek out for new business opportunities.

## 1.2 Problem Statement

The main research problem is presented in the following question:

*What is the state of the software component market in the telecommunication industry?*

The subsequent research questions derived from the main research question are:

*How is the software component business likely to evolve in telecommunication industry?*

*What would be the most suitable market position and expansion routes in software component market in telecommunication industry for a major software vendor?*

## 1.3 Objectives and Scope

The main goal of this thesis originating from the case company's current situation is to explore the opportunities of expanding software component sales in the selected market by reviewing the status of the software component business within telecommunications industry. The objectives in this study are first to build understanding what kind of solutions different component software technologies enable, then gain understanding of the current market situation in telecommunication industry and finally develop and analyze future industry scenarios. The Industry scenarios could be in turn used to support decision making when evaluating the attractiveness of different go-to-market options of a specific software product family.

The technological and business aspects of software intensive products merchandised in telecommunications industry are in the main scope of this study and the software component business is studied mainly from the component supplier point of view. Although all parties in software component business are taken into account, the main interest is to cover the component supplier's direct customer relationships and not to cover further customer relationships like the one shown on Figure 1 between "customer" and "end-user". The techno-economical representation of the focus areas of this thesis is illustrated in Figure 1.

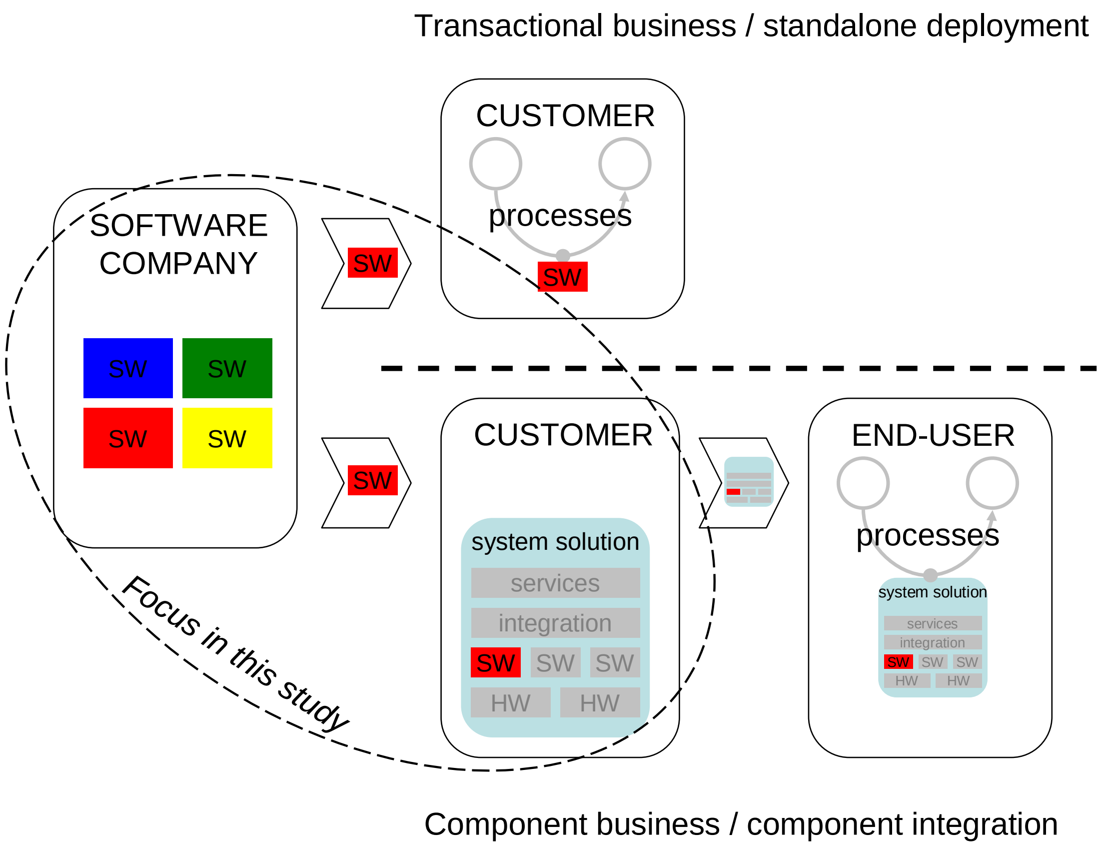



**Figure 1: Software component as a part of system solution versus standalone software deployment. Adapted from Helander (2004)**

## 1.4 Used Methods

The telecommunications software component industry's market analysis is completed by applying the famous Five Force model introduced by Michael E. Porter (2004) to expose competitive environment and sources of competitive advantage. The theoretical research done in the first part is based on sources from literature, articles and the Internet.

The empirical data was gathered by conducting a series of industry expert theme interviews. Interviewees from both software component vendor and component buyer side were selected to comply with the scope of this study. Interviews were used to broaden the view of current industry structures and to define the most important uncertainties affecting the future of the software component business. Data from market studies and interview are then broth together to construct plausible near future industry scenarios using the Porter's scenario analysis methods. Both methods used in this thesis are introduced in the beginning of the chapter they are applied.

## 1.5 Outline of the Thesis

First part of this study includes backgrounds behind this research giving the reader an overview what is the context and purpose of the thesis (Chapter 1), an overview of software components, existing component-based technologies and relevant business concepts (Chapter 2) and a market study (Chapter 3). The first chapter is also a research plan of this thesis describing research questions, objectives, scope and used methodologies. Chapter 2 gathers the most relevant literature on the concept of component software, software business models, original equipment manufacturing business and their indications in software component manufacturing business. The first part of the thesis (Chapter 3) includes also a market study to reveal the current business configuration of the software market in the telecommunications industry. The second part combines the findings from market analysis done in the first part with input from industry experts and finally the industry scenarios are created. In the third part of this study the industry scenarios created in part two are introduced and analyzed and possible strategic choices for a software component vendor are discussed. Finally the last chapter concludes the whole research, discusses the relevance and validity of the results, and gives ideas for further studies. The structure of this thesis is illustrated in Figure 2.

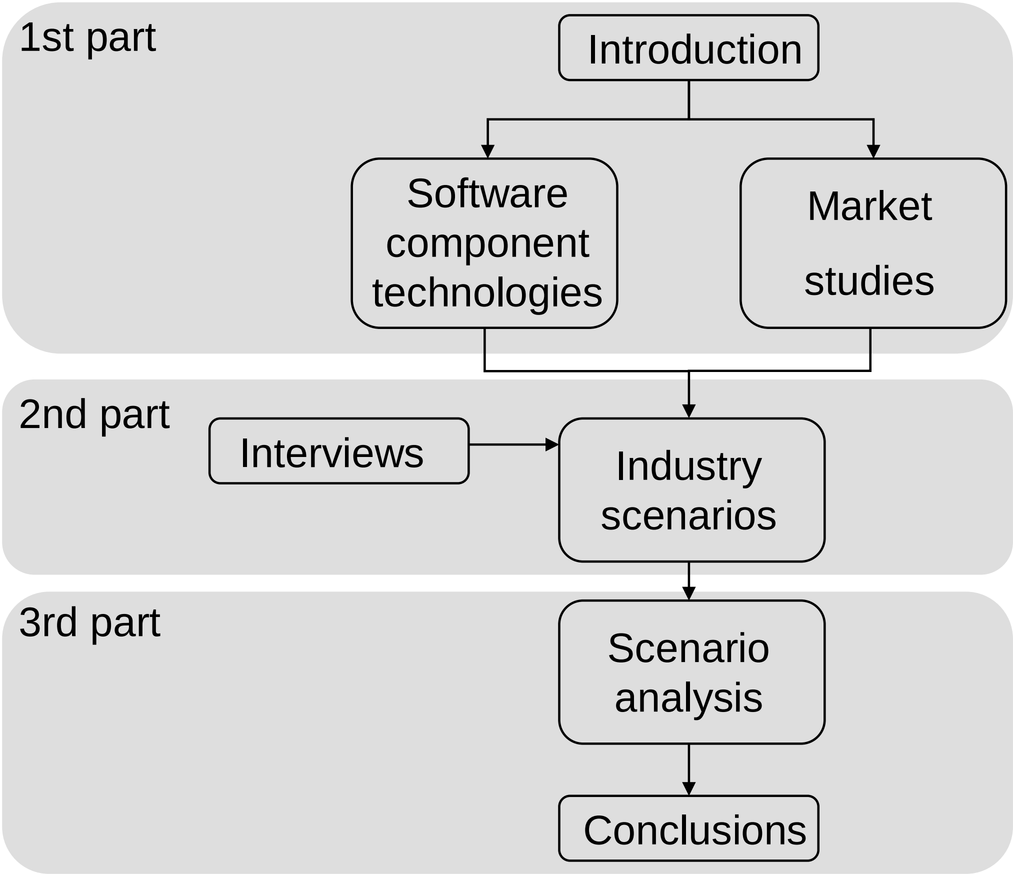



**Figure 2: The structure of this thesis**

# 2 Theoretical framework

This chapter gives an overview of the fundamental concepts behind what is called a software component business in this paper. The idea that larger software products can be developed and assembled using standard ready-made components and that a software component vendor can make a profitable business selling these components are used as a starting point for this study. The overview in this chapter is given discussing the earlier academic literature about software components and related software business topics. This chapter clarifies what software component-based product development is, what are the benefits and shortcomings it introduces and what is the economic foundation to do business with software components.

## 2.1 History of Software Components

The idea of using software components to build larger software systems is not a new one. In NATO congress held in Germany 1968 Doug McIlroy predicted that mass-produced software components could be the answer to overcome software engineering problems discussed in the same congress (Naur & Randell, 1969). In the past 40 years McIlroy's predictions were first realized in form of object-oriented programming, which was studied by Brad Cox. Cox researched object-oriented programming paradigm and how it could be implemented. Cox used the term "software-IC" (Integrated Circuit) to describe the analogy between software and hardware engineering. Later Cox implemented his ideas in the form of the Object-C programming language (Cox, 1986). The latest innovation in component models were made in early 1990's when IBM and later Microsoft introduced their own models named accordingly System Object Model (SOM) and Component Object Model (COM). Different component models are trying to increase the language independence and thus portability of software components and objects from one environment to another. The latest dominant component technologies are further discussed in section 2.6.

## 2.2 Defining Software Components

In the Object Management Group's (OMG) Unified Modeling Language (UML) specification (OMG 2009) a component definition is general enough to cover both the logical and physical component. OMG's component definition is used a starting point of defining software components in this study, because OMG's definition has the right level of abstraction.

OMG (2009) defines a component to be a reusable, autonomous unit within a system or a subsystem. A component provides one or more interfaces through which the component can be accessed by other component in the system. According to OMG a component can also be dependable of one or more interfaces. The internals of a component are hidden and thus are not directly accessible from outside of the component and therefore a component could be said to be encapsulated. According to OMG (2009) components should be treated as independently as possible to attain flexible reuse in different systems. Components are connected together via their provided or required interfaces. In OMG's definition any difference is not made between small or large components. An example UML presentation of a component with two provided interfaces and three required interfaces can be seen in Figure 3.

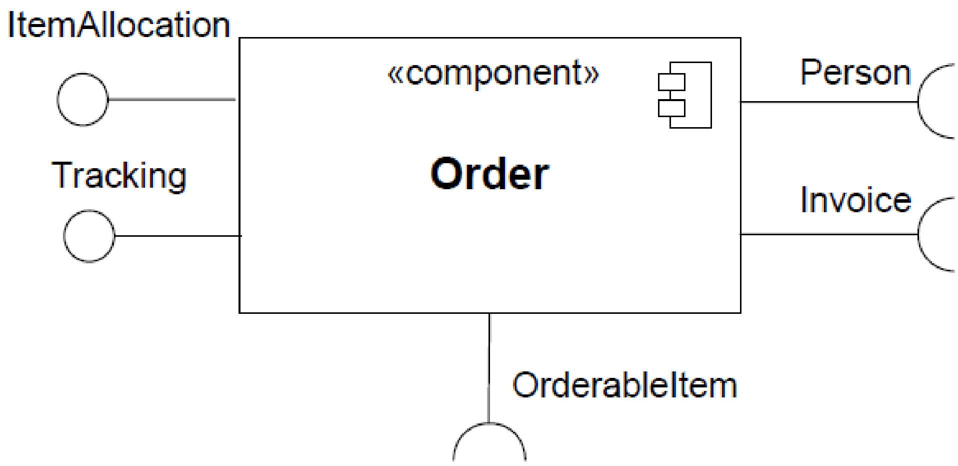



**Figure 3: OMG's example UML presentation of a component (OMG 2009)**

Szypersky et al. (2002) defines a software component as follows:

> *A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.* (Szyperski et al. 2002)

From above definition we can see that software component is a unit, which contains the composition of functions. The component's functions can be accessed via a standardized or some other way commonly agreed interfaces. According to Szypersky et al. (2002) the independent deployment means that a component should be separated from its environment and other components and that a component can never be deployed partially. Third parties, which are component buyers, in Szypersky's definition have no access to component's construction details, which can for example mean that third party don't have access to the software component's source code and thus cannot modify the component.

Considering the above definitions and the context in this study, the following characteristics defined by Helander (2004) are chosen to define a software component:

- Software component is a reusable computer program
- that is accessible through specified interfaces,
- is integrated into a larger software-based system product as an individual operational part, and
- is not valued by the end customer as a standalone application.

The definition we have chosen to describe software components covers software products that are mainly package software products. Software component in this study can refer to a product that offers a set of functionalities which alone are not valued by end customer regardless that the product can be deployed as a standalone solution. Example of such a standalone software product could be a database management system. Nevertheless, the above definition we have chosen includes also other types of software components, such as a set of compiled classes which are linked into some package. Difference between independent and component software deployments is illustrated in Figure 1 (see Chapter 1.3). The end customer mentioned at the fourth point in the listing above refers to the end customer of the system product as shown in Figure 1. Figure 1 also points out the focus we have in this study and illustrates the delivery chain of the software component.

## 2.3 Advantages of Component-based Production

Component-based approach in industrial systems manufacturing is already widely adapted way of building bigger systems in traditional industries, like in car manufacturing. Introduction of industrial or de facto standards among each industry has made possible to use ready-made components in building bigger systems and thus enabled the component business to emerge. Venyard (2001) has generalized the fundamental business idea of components as a sale of some firm's assets, which are packaged into black boxes. The component buyer can then apply component provider's know-how to its own product immediately and according to Venyard (2001) reduce the overall costs and risks of production. According to Venyard the flexibility of business processes and operations is also increased while their complexity is reduced. In other words component-based manufacturing of industrial systems reduces time-to-market of the system product, costs, risks and complexity of operation of original manufacturing company while it increases flexibility to make changes to system products, which are build using standard components.

According to Szyperski et al. (2002) the software industry is following other industries adopting the component-based approach in product development mainly because the software component-based approach provides the same benefits as already proven in other industry areas. Szyperski's answer to question about what motives the use of software components is simply put: "… components are the way to go because all other engineering disciplines introduced components as they became mature – and still use them."

Niemelä et al. (2000) have made industry case studies covering Finnish software industry about the utilization of industrial component software. They found out that although commercial software components had been used in software development, the case companies still had difficulties exploiting the benefits of component based development, because of variation in components' quality, lack of needed functionalities and poor quality of documentation. In their study Niemelä et al. also found out that at the time of study component buyers were expecting a component based development to bring them more efficiency, quality and business benefits. These expectations were based on opinions that the software component business and component based development practices were still on a very immature level about ten years ago.

## 2.4 Different Types of Software Components

Software components can be divided into three different types based on how much control a component acquirer has over what functionalities component includes and how the acquirer can modify components. Institute of Electric and Electronic Engineers' (IEEE) standard 1062 introduces the two types of software components: Commercial of The Shelf (COTS) and Modifiable of The Self –components (IEEE 1998). The third important type of software components are the open source components.

COTS-components can be said to be the most mature type of components. According to IEEE's (1998) definition they are stable products which have already been widely tested in commercial implementations. COTS-components also have well-defined and documented functionalities and their capabilities and limitations are known. COTS-component vendors are not usually willing to modify the internal workings of their component product according to buyer requirements, because COTS-components are typically build to solve generic problems. In general COTS-components source code is not disclosed to component buyers or other outside parties.

MOTS-components are like COTS-components with a difference of vendors' willingness to make modifications to them (IEEE 1998). This fact makes MOTS-components acquirer-specific and modifications made are not usually applicable in other contexts. A MOTS-component can for example include both common functionalities, which are implemented in several similar MOTS-component vendor components, and acquirer-specific functionalities. In this case component vendor and acquirer could agree to share access only to the source code of the modified part. MOTS-component development could also be done by the joint development of the vendor and the acquirer.

Open Source Initiative (OSI) defines open source software to be freely distributed, used, modified and improved and that derived works must also comply these terms (OSI 2010). All these terms are defined in open source licenses. Open source components can be developed by distributed groups of people or open source components can be derived from commercial software products where software vendor has released certain pieces of its software to the open source domain. Commercial software vendors also benefit if they participate in open source projects, because they can make good use of open source community's contribution to their own products. International Business Machines Corporation is for example participating in more than 120 open source projects[1].

In this thesis, the main interest is on COTS-components, because they are by definition those kinds of software products which business logics are the focal research topic of this study. The main differences between COTS-, MOTS- and open source-components are summarized in Table 1.

**Table 1: Summary of characteristics of different types of software components**

| | COTS | MOTS | Open Source Components |
|---|---|---|---|
| modifiable | no | partly | yes |
| free of charge | no | no | yes |
| acquirer has access to source code | no | no/partly | yes |
| documented | yes | yes | partly |
| supported by vendor | yes | yes/partly | no |

[1] IBM's open source efforts: http://www.ibm.com/developerworks/opensource/newto/#9 (visited 14.2.2010)

## 2.5 Layered Architecture for Component-based System

In order to compose components from different sources to form a complete product, component-compliant technologies and techniques of some kind need to be established. As discussed earlier in this paper, a component can only be accessed through well-defined interfaces and that's why a component-based software system needs a software platform and architecture which enables component integration. Moreover the interconnection of components and other pieces of software is the core tasks of integration. One solution to enable the use of software components is to establish layered system architecture. In layered architecture a software system is divided into several layers of functionalities. Each layer offers services to the upper layer based on its functionalities and in turn make use of lower layer services. Layers interact with each other through well-defined interfaces. Simple layered architecture is illustrated in Figure 4.

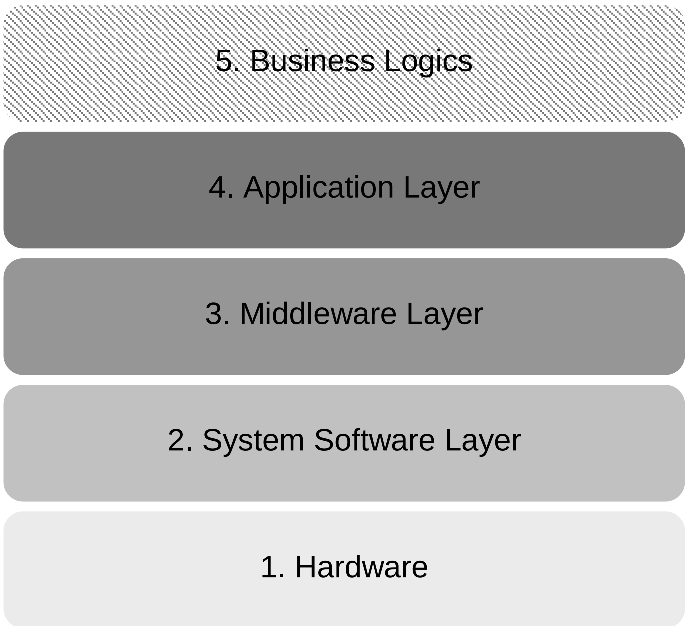



**Figure 4: Layered system architecture (adapted from Jacobson et al. (1997))**

Fundamentally, the architecture is used to define overall system structure, the interfaces of the different layers and components and the interaction patters of the system (Jacobson et al. 1997). According to Jacobson et al. (1997) a well defined and proper architecture is the key to manage complexity in component-based software systems and later to maintain the integrity of the system. If architecture is ill-fitting or poorly defined for a particular component system, it may lead to difficult and expensive development. This is due that the components seldom naturally fit together without proper definition of their use in the system.

Jacobson et al. (1997) suggest that layered architecture is a good way to tackle the problems in component-based development. In layered architecture, the software is organized into different layers. In each layer, a component system may be in turn build from components. The upper layers include more context specific components and the lower layers more generic components. On each layer, the components are depending on the services lower layer components offer so each layer is building on top of each other and adding more specialized features to the system. Jacobson et al. (1997) remains that in a layered architecture a component should only be dependent on the lower layer services, but not vice versa. This important notion about dependencies is made, because lower layer components are perceived to be more resilient than upper level components, that is when a component system evolves it is more probable that an upper level component is changed or substituted with another one than more general lower lever component.

A typical layered architecture of a component-based large system is illustrated in Figure 4 and it has five different layers, which are from top to down:

*Business logics* - layer is an abstraction of real-world use-cases and processes for which the software is designed to be an application for.

*Application layer* contains application specific components, which are tailored to solve some specific problem. For example in the context of this study this layer could include network element monitoring application components.

*Middleware layer* is probable the most important layer in component architecture, because it provides services for components to interact with each other. According to OW2 consortium (2010) middleware include intermediate software which provides the following services: hide the connection of distributed components, hide the components heterogeneity to enable components from different operating system and network technology domain to interconnect with each other, provide high-level interfaces for upper layers to use the two earlier mentioned services and provide common services to perform general tasks such as database management, load balancing and application server functions.

*System Software Layer* contains operating system software and other hardware specific software components, such as network interfaces, that are needed to run the hardware system.

*Hardware* – layer consists of computer hardware components, which in most cases is some kind of multipurpose server hardware.

## 2.6 Component Technologies

Apart from component-based architecture, technologies enabling the composition of components written in different programming languages are essential in component-based software development. The goal of these different technologies and standards is to allow components from heterogeneous environments to communicate together by for example calling each other's operations (Jacobson et al. 1997). The component technologies discussed in this study are commonly implemented in the middleware layer (see Figure 4).

Current component technologies can be divided into two fundamental groups. The first group of technologies emerged in mid-1990 as distributed systems and component-paradigm gained popularity. Technologies in the first group are large industry originating frameworks defining all services component-based software systems needs. Component technologies belonging to this group have common ability to enable components to exchange information in binary format over a communication network. Windows Communication foundation (WCF) and Common Object Request Broker Architecture (CORBA) are considered belonging to the first group and are further discussed in this chapter. The second distinguished group of component technologies has emerged as network protocols and techniques from Internet-domain have been adopted to interconnect distributed components in machine-to-machine interaction. Descriptive to technologies in the second group is that they transport information between components in higher level format and are more independent of what type of communication networks exits between interconnected components than technologies in the first group. Protocols and specifications used to build Web services, namely Simple Object Access Protocol (SOAP), Representational state transfer (REST), Web Services Description Language (WSDL) and WS-* specifications, are further discussed in this chapter as representatives of the technologies in the second group.

### 2.6.1 Windows Communication foundation (WCF)

Windows Communication Foundation is a proprietary software development toolkit developed by Microsoft. WCF is part of Microsoft's .NET framework and was released with .NET framework's version 3.0 in

2006. WCF is a suite of protocols and methods of supporting communication between distributed components (Chappell 2007/1). The Microsoft .NET framework and WCF can only be run on the Microsoft Windows operating system, such as Windows Server 2008. WCF is backwards compatible with former Microsoft's component technologies such as Distributed Component Object Model (DCOM), which was a component model supporting low level binary format communication between distributed software components. In WFC, a service oriented approach in component interaction and interoperability is used meaning that server-side components expose services through interfaces to the client-side components to consume them (Microsoft 2010) (Microsoft 1996).

WCF supports all currently used inter-component communication paradigms: inter-process communication between components running on the same machine, binary format wire communication between components distributed over a computer network, high-level message based web-services type of communication and message queuing communication. Components developed or running on other than Microsoft platforms can be interconnected to WCF components by using Web services which use Simple Object Access Protocol (SOAP) along with WS-* specifications or by RESTful Web services mainly over Hyper Text Transfer Protocol (HTTP) (Chappell 2007/1)(Chappell 2007/2). Figure 5 shows an example of distributed components, or services as they are called in WCF, which are running on three different machines running different platforms. In Figure 5 machines are connected to the client machine via local area network and Internet. A used connection type of each component client a component is connected to can also be seen in Figure 5. Web-services protocols and specifications are discussed later in this chapter.

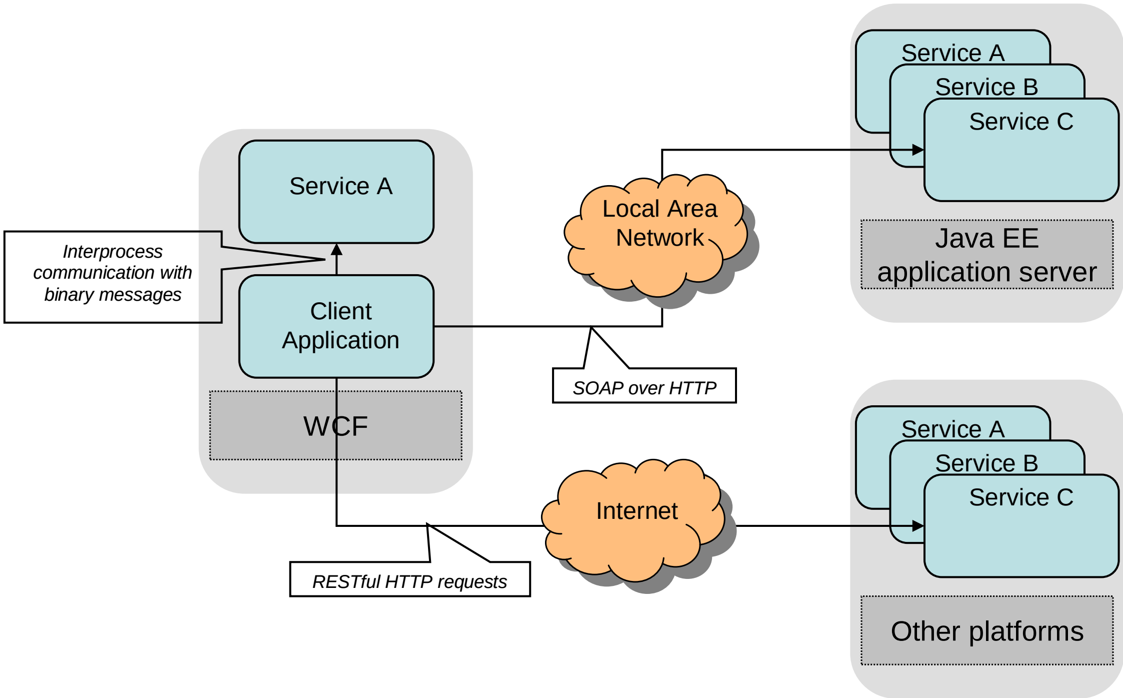



**Figure 5: Component interconnection options in Windows Communication Foundation**

WCF itself does not provide means for programming language neutrality and support for multiple programming languages are established in other parts of .NET framework (Microsoft 2010). Notable about WCF is that, although it is a collection of all the Microsoft's previous component- and web services-based technologies, it is not trying to provide one universal solution to interconnecting components, but rather offer all available composition options under one toolkit.

### 2.6.2 Common Object Request Broker Architecture (CORBA)

Common Object Request Broker Architecture (CORBA) is a middleware technology standard by Object Management Group (OMG). CORBA provides means for a heterogeneous client and a server distributed over a heterogeneous network environment to interconnect. In other words, CORBA allows components written in different programming languages and running on different servers running different operating systems to interoperate with each other. First versions of CORBA specifications were published in early 1990's and current version 3.1 was released in 2008 (Vinoski 1997)(OMG 2008)(Chung et al. 1997).

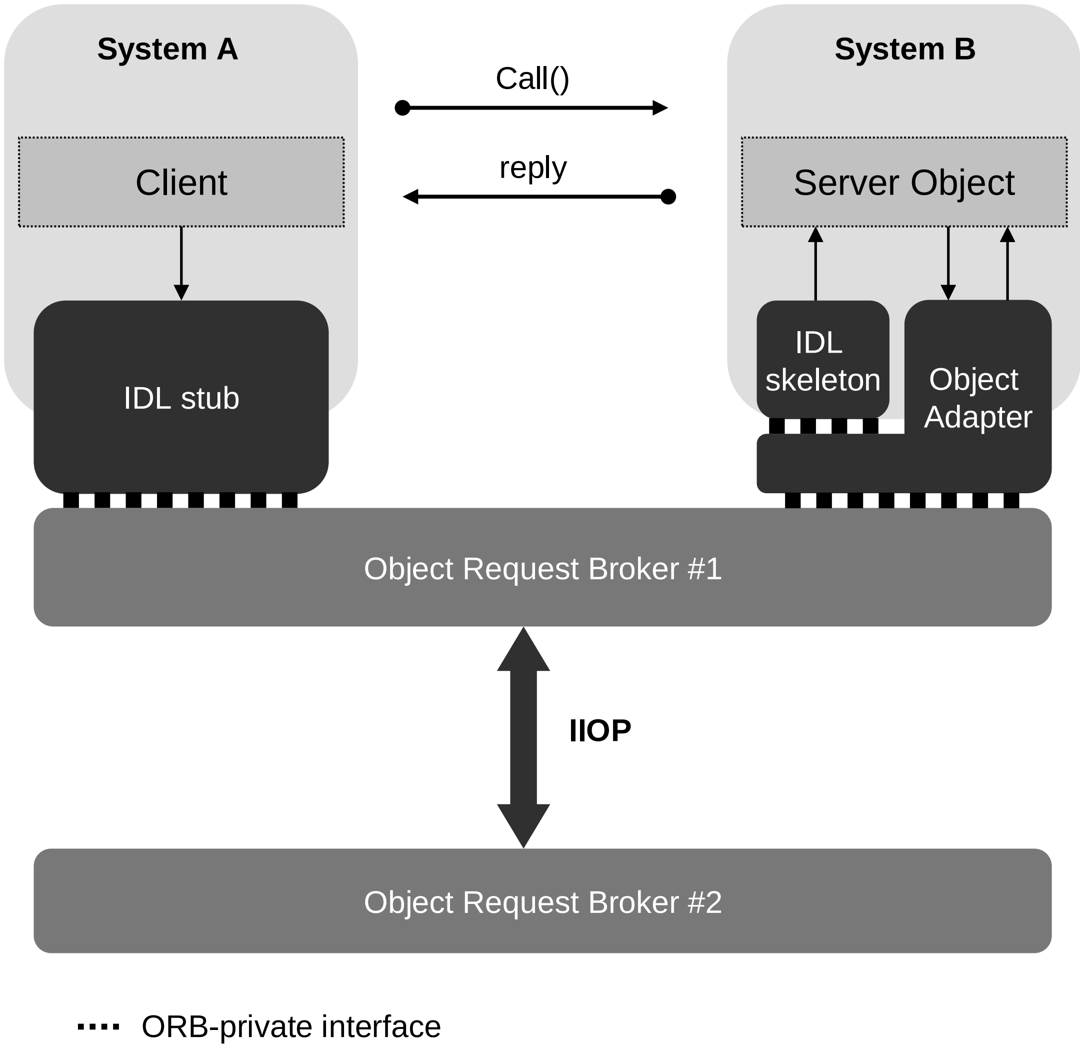



**Figure 6: Simplified CORBA architecture**

Figure 6 shows a simplified CORBA architecture, where systems A's component has a CORBA client which is calling system B's CORBA object. In Figure 6 a client call and server object reply are illustrated at high level, but actually all communication between objects are delivered through Object Request Broker (ORB). ORB is the basis of CORBA architecture and it provides functionality to deliver remote procedure calls (RPC) back and forth between CORBA-objects. ORB hides object location, the execution state, implementation and underlying communication mechanisms from other objects. All objects are accessed through interfaces on both client and the server side. In CORBA interfaces are defined using Interface Definition Language (IDL). IDL is not programming language specific. An interface defines object's supported operations and types. From interfaces written with IDL can also inherit multiple other IDL interfaces enabling re-use of already defined interfaces when creating new ones. Each CORBA implementation comes with IDL compiler, which is used to create client and server side programming language specific proxies. These proxies are called client stubs and server skeletons in CORBA and are can be seen in Figure 6. Stubs and skeletons are actual language mappings used in server object or client implementation to map IDL defined operations to equivalent programming language function (Vinoski 1997). The third ORB interface illustrated in Figure 6 is the object adapter. The object adapter adapts the implementation of object interface to ORB services by exporting a public interface to object implementation and a private interface to a skeleton (OMG 2008). Last CORBA's functionality shown in Figure 6 is Internet inter-ORB protocol (IIOP), which specifies transfer syntax and a standard set of message formats for two ORB's to interconnect over standard TCP/IP[1] connection (Vinoski 1997).

All CORBA defined object request interfaces are not shown in Figure 6 and thus not introduced here. More detailed description of interfaces and their functionalities is out scope of this thesis.

### 2.6.3 Web services

In this paper Web services refers to architectural definitions by World Wide Web Consortium (W3C). Web services are the latest large scale industry consortium standardization efforts in the area of machine-to-machine interoperability. Web services have the same goal as WCF and CORBA to provide standardized means for application running in heterogeneous environments to interoperate with each other over a communication network. W3C distinguishes two major classes of Web services: services which use Representation State Transfer (REST) methodology and arbitrary services which just expose arbitrary set of operations. The REST model's purpose is to define uniform server-side interface semantics. A REST-compliant interface should provide semantics for create, retrieve, update and delete operations (W3C 2010). With Web services a Service oriented architecture (SOA) can be implemented.

Web services are based on client-server paradigm and client implementations of web services are called requestor agents and server side web service implementations provider agents. Web service interfaces are described in Web Service Descriptions (WSD), which are machine-processable specifications. WSDs are written using web service description language (WSDL). In WSD Web service interface's message format, data types, transport protocol, transport serialization formats

[1] Transfer Control Protocol (TCP) / Internet Protocol (IP)

and service's network locations are defined. The requestor agent then uses information in WSD to connect, send and receive messages to and from the provider agent (W3C 2010). Web services architecture is illustrated in Figure 7 below.

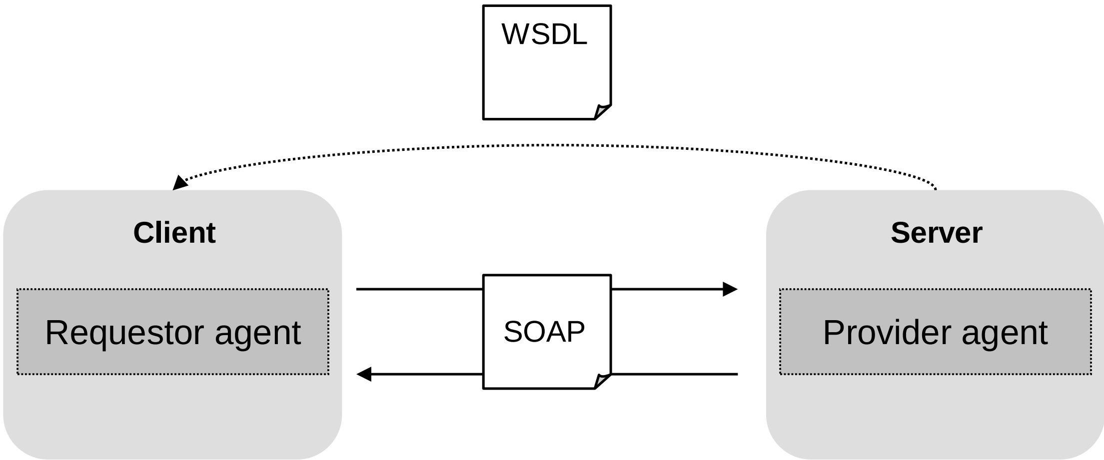



**Figure 7: Web services architecture (W3C 2010)**

Although W3C does not require certain protocols or technologies to be used, a common way to implement a Web service is to first describe the Web service's interfaces with WSDL and then use Simple Object Access Protocol (SOAP) over an HTTP connection to establish a connection between client and server machine. SOAP is a text-based protocol to carry messages over a network and is independent of underlying protocols. SOAP messages are written with extensible markup language (XML), which is a textual data format defined by W3C (W3C 2007).

Apart from W3C, other industry consortiums exists as well who are providing specifications and recommendations for the use of Web services. Such consortiums include Web Services Interoperability Organization (WS-I), which is an open industry organization. WS-I delivers recommendations how Web services should be implemented using a certain set of protocols and technologies and test tools to test Web services implementations compliance with WS-I's recommendations (WS-I 2010). Other important consortium in Web services standardization is the Organization for the Advancement of Structured Information Standards (OASIS). OASIS has multiple working groups concentrating on Web services standardization. Web services working groups have standardized many functions needed when implementing Web services like security and certain industry specific messaging (OASIS 2010).

## 2.7 The Business of Software

Software is a digital information product with no physical characteristics and therefore it has special implications to business around it. Although it is not completely clear whether software is more like a product or a service, software have many characters of physical products. For example, unlike a service, software product can be returned back to seller when it is not used anymore by agreeing that buyer do not have the right to use the software anymore. According to Shapiro & Varian (1999) production of information goods, like software products, has distinctive cost structure compared to productions of physical products. Producing the first version of a new software product cause the fixed costs resulting from software development. These fixed costs are also largely sunk, because labor costs from development cannot be recovered if software product fails in the markets. Variable costs are typically insignificant in comparison with fixed costs, because there do not exist production capacity problems with software and distribution costs are nearly zero. Other cost consists of marketing, selling and supporting the use of software product. Because software has characteristics of information product there exist only few sustainable market structures in software business (Shapiro & Varian 1999).

Hoch et al. (1999) has identified two main categories in software markets, namely professional software services and software products. Moreover the software product market is separated into two sub-segments: enterprise solutions and packaged mass-market software. Firms operating in these two latter segments typically implement very different business models. In order to reach mass-markets firm must have very general product offering, which satisfy the needs of masses. Software firms targeting mass-markets must also support a large scale selling and distribution channel in their business model. Respectively firm offering enterprise solutions should perhaps have more specialized and probably more expensive sales resources and domain specific product offerings.

Rajala (2001) have listed four major elements, which should be considered when analyzing or constructing a business model for a software firm. These elements are: product development approach, revenue logic, marketing and sales approach and servicing and implementation approach. Product development approach is the element of a business model where a process to deliver a value proposition is defined. Rajala (2001) represents multiple applicable revenue logics for software business. Revenue logic determines how software product's sales value is appropriated to the software company. According to Rajala (2001) the gamut of revenue logics in software business is diverse, but three revenue logics listed by Rajala are currently widely used in product oriented software business. Three most used revenue logics are as follows.

> *Licensing*, which means selling a right to use the software. A license fee can be based on a number of users or computers the software is installed in.
>
> *Profit sharing* is a form of licensing where license fees are dependent on customer's performance when using the software. A customer for instance can re-sell the software as a part of its own product and then share the profits with a software supplier gained from selling the product. Alternatively, the license fee could be a fixed fee paid per every sold customer product.
>
> *Loss leader* refers to a model where a core software product is offered for free, but all related and supplementary products and services are chargeable to buyers who use the free product.

Marketing and sales models in software business concentrates mainly on selecting strategically best fitting sales channels for a certain company or a product. A software company can use different sales channels with different products. Possible sales channels can be divided into direct and indirect channels. In direct sales, a company has an immediate customer relationship with a buyer, whereas in indirect sales some intermediary company is involved. Intermediary company's role in a sales channel can diverge greatly from being a distributor who operates a market place for software products to a strategic partner who have expertise in a certain customer domain or technology. Lastly the service and implementation approach of the business model include activities resulting in a working solution for a customer.

In developing a business model, it is a matter of a right combination of options on each four elements mentioned above. According to Rajala (2001), each option in different element has interdependencies with each other. In software project business for example a direct sales efforts targeting a partnership with a customer could be the best combination of different options when the product development approach is considered along with the sales channel and the sales approach. Similarities to famous 4P marketing mix can be seen in Rajala's model. 4Ps in marketing mix stands for a product, price, promotion and place, which are four fundamental activities a marketer should address in order to execute effective marketing of a product or a service (Kotler & Keller 2005).

## 2.8 Characteristics of the component business

As explained briefly in the first chapter, the main focus in this study is on component business where components are sold to customers who integrate them into a larger system product and then sell on the system as their own product. These kinds of component vendor’s customer organizations are in many industries called original equipment manufacturers or shortly an OEM company. By component business we mean the trade relation between a component supplier, also known as an OEM supplier, and a component buyer or an OEM company as they are commonly referred to.

Component business is fundamentally characterized by the derived demand in industrial markets. Derived demand is a market force which defines the demand for components and derives from end-users or consumers along a supply chain towards components and raw material suppliers (see Figure 8). Derived demand affects the component supplier’s marketing strategies, because a component supplier should gain understanding about the needs of not just its direct customers, but also its indirect customers who are namely the end-users and consumers. There can be multiple levels of companies involved in supply chain between a component supplier and an end-user and that makes it more complicated to the component supplier to influence the demand on all levels. This said, the main focus of the component supplier should however be in building expertise in understanding the business an OEM company is doing and what are the motives and practices of an OEM company when they use components to build larger systems (Chisnall 1995). Seppänen et al. (2001) also mentions two additional aspects influencing on component demand. These are joint demand, where demand for two or more components has mutual influence on each other’s demand, and cross-elasticity, which basically means that a sale of one component has effect on a substitute component’s price.

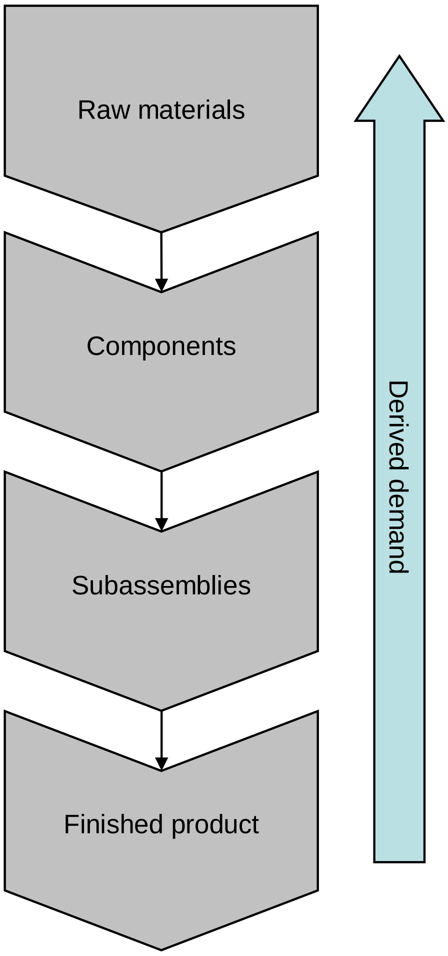



**Figure 8: Supply chain and pattern of derived demand for industrial products (Chisnall 1995)**

If the main concern for a component supplier lies on managing the demand, then it is evident that for an OEM company a component business is mainly a matter of supply chain management. The basic goal of supply chain management is to optimize the supply of components, so that a right kind of component would be available in right quantities on a right time and place. Supply chain management covers a rather wide set of OEM company's activities the essential ones being: evaluation and selection of suppliers, building supplier relationships, managing contractual agreements about component supply, establish information exchange practice with suppliers and finally manage supply chain risks (Waters 2003).

## 2.9 OEM in Software Component Business

Since component technologies emerged in the early 1990's the component-based software development and intra-organizational software reuse practices has been the main area of research in a software component domain (see for example Mohagheghi et al. (2008)). Although the acquisition of third party software components has got some attention in literature, the management related studies have mainly concerned only on the buyer's point of view (Meyers & Obendorf 2001).

Seppänen et al. (2001) has studied the state of the software component markets especially in electronics, telecommunication and automation industries and found out that although in the time of their studies the in-house software production was largely component-based, the business on open component markets was still immature and no clear market structure was found. Companies acting on the market were facing problems in interacting with each other. The main source of immaturity of software component market was claimed to be the lack of industry standards and management guidelines. Additionally Seppänen et al. revealed that because software industry is still young in comparison with traditional industries, like automotive industry, it has not yet structured to OEM style supply channels or networks. Predictions were also made about the software component market following other industries in supplier-buyer relationship development from horizontal networks to the more vertical alignment of suppliers. In vertical network component buyers want to have fewer and more competent component suppliers with more responsibility thus reducing the number of direct buyer-supplier relationships. It was still unclear whether intermediaries of some kind will exist in software component markets and what value they would provide for component suppliers and buyers. In their studies Seppänen et al. (2001) found out that the most of the relationships between the suppliers and the buyers were long lasting and close, which was considered the main reason why intermediaries were not present.

In the study of Helander et al. (2002) the software component market was analyzed from the value creation point of view. This study found out that companies acting on software component markets had not yet convinced about the benefits of inter-company software component business. The main barriers of effective use and selling of components were the difficulties in component sourcing, pricing issues and difficulty of analyzing the life-time costs of component use. Buyers also had problems finding right type of suppliers and components as well as concerns whether they had recognized all suitable suppliers on the market when selecting the supplier. Other critical factor in component sourcing was unwillingness from both the supplier and the buyer side to exchange enough information about the component and the system product to which the component was planned to be integrated. Suppliers and buyers had both fear of revealing too much proprietary details and thereby leveraging valuable know-how. Pricing and life-time cost analysis difficulties are related to each other. It was hard for a supplier to analyze and predict what would be the buying volumes of a certain buyer and this had effect on component pricing strategies as well. Besides these monetary value benefits, both component suppliers and buyers should also consider the value of non-monetary effects of component use and selling, like the effect on the market position and learning. The value these non-monetary benefits are creating is even more difficult to estimate than the clearly monetary benefits. The final proposal of Helander et al. (2002) is that, based on the need for close and co-operative relation between component supplier and buyer and the complexity and the vast amount of different software components seen in market, “perfectly” working software markets will very unlikely come true.

## 2.10 Pricing

The product price is in a simplified manner inversely proportional to the sales volumes. This relationship can be seen in Figure 9. In other words, the higher price leads to reduced demand and thus the lower volume of sales and vice versa. The terms and conditions under which the information good, e.g. a software product, is sold have influence also on the demand for the product. The more the terms and conditions give rights to the buyer the more the buyer is willing to pay for the product, but again the higher price will cut down the overall sales volume of the product. Figure 9 illustrates the idea of differential pricing, where the same product has a different price depending on the characteristics of the target customer (Shapiro & Varian 1999). Shapiro & Varian (1999) introduces three different means to implement a differential pricing of information goods:

*Personalized pricing*, where different price could be set for each customer,

*Versioning*, where a product line of different versions is made available and

*Group pricing*, where different prices are set for different kinds of customer groups.

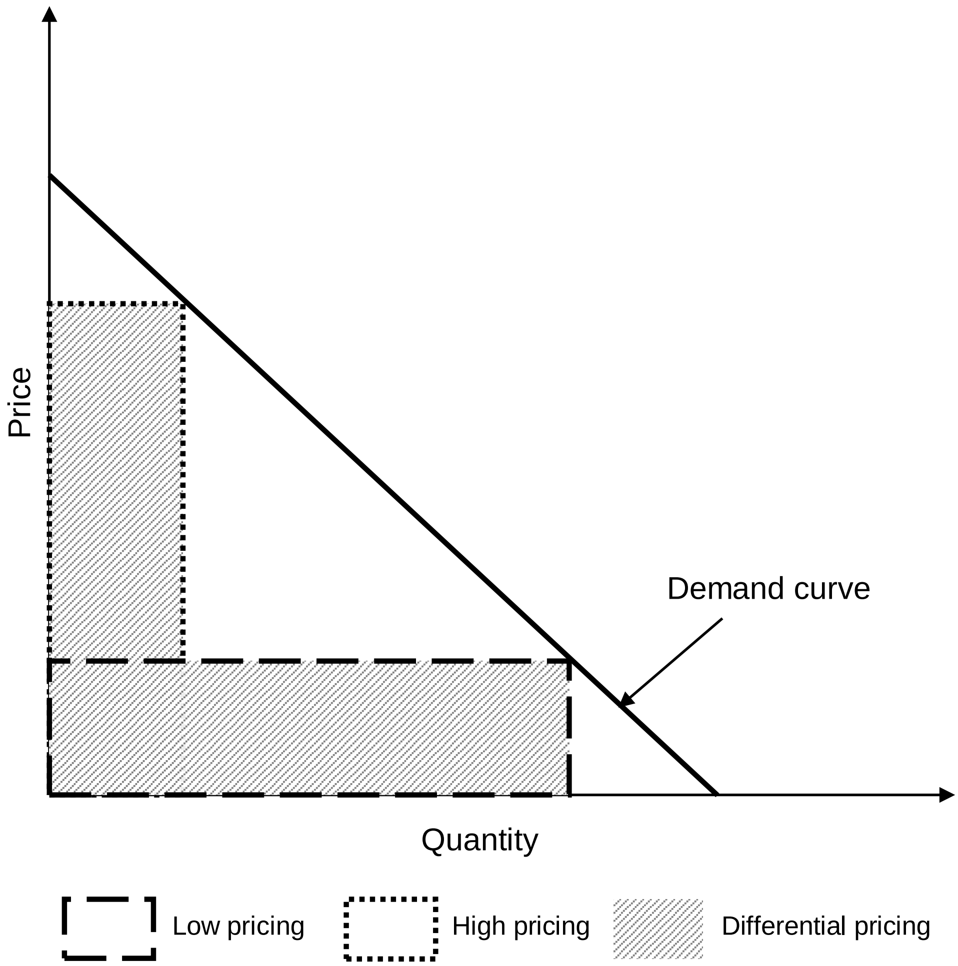



**Figure 9: High, Low and Differential Pricing (Shapiro & Varian 1999)**

Setting the right price for a software component product can be rather complicated compared with a standalone software product. Because the final products', which contain the same software component, complexity and "size" and thus the final price may vary radically and the pricing of the software component may need flexible pricing models. Therefore the software supplier should differentiate its component price based on customer's final product's pricing and sales volume estimations.

If the same software product is sold both as component and standalone product, the pricing decision becomes even more complicated. In this case, a software supplier is in situation where it must usually reveal list prices of some kind for the standalone products and at the same time use differential pricing in component sales. According to Shapiro & Varian (1999) the published list prices will set the highest price the supplier can charge any customer. Multiple pricing options with the same software product also increase the complexity and thus can weaken the reliability of sales revenue predictability.

## 2.11 Intellectual Property Rights and Contracting

Immaterial property rights (IPR) has increasing importance in software business, because of the globalization and rapid growth of the software markets. Immaterial property rights can be protected by several legislation and they are common for all digitally stored and thus immaterial property. The most important legal means to protect one's immaterial rights are copyright, patent, a trade secret and contracts. The most used protection mean are the copyright laws, which protects the software from unauthorized copying, using and exploiting. Copyright laws give protection only to source and object code of software, but ideas, procedures or methods are not protected by copyright laws. The other important protection mechanism is patenting. With patenting a product, process or computer program can be protected so that a patent holder has exclusive rights for a defined period to capitalize on the patented innovation. Patents are not automatically granted and innovation needs to be novel, non-obvious and useful in order to be eligible for patenting. Software patents are used mainly in the United States and recently the European Union has rejected a directive proposal allowing software patents[1]. Trade secrets, such as procurement or software development processes and tools used in them, are also protected by law. When companies have intentions to do business together, they usually write a so called non disclosure agreement (NDA) before any further negotiations are done. With NDA companies explicitly agree not to disclose any information which could be regarded as a trade secret (Warsta 2001).

Contracts between a software supplier and a buyer have an essential role in software business and probably have even more importance in software component business. By a contract a supplier and a buyer are basically giving each other a promise how they will act when the product or service and monetary or other compensations are exchanged. A contract binds both parties to comply with the terms by introducing penalties of some kind in case of not acting in compliance with the agreement. One important principle in contracting is the freedom of contracting, which means that contracting party has a freedom to choose with whom to contract, what is the content of the contract, how the contract is put into effect and which country's laws are applied in case of disagreement. The main contract type in software component business is a license agreement. Besides the right to use the licensed software, a license agreement also usually determines duration for the right, environment the software is allowed to be used in and how many copies of the software licensee are allowed to make and use (Warsta 2001).

[1] Patenting Software vs. Free software - What should the European Union do? - Briefing Paper http://people.ffii.org/~jmaebe/epecosci0502/SoftwarePatent.pdf (visited 26.3.2010)

Seppänen et al. (2001) have discovered several contracting issues in software component business. They found out that the main reason for contractual issues is the remarkable differences between companies' legal expertise in contracting. Smaller companies are not so interested in developing their contracting policies and processes or do not have enough resources to do so. The contracting party who has more legal competence has also more bargaining power when contracting and smaller companies are usually forced to follow the terms the more legally competent party demands. Seppänen et al. also found problems in agreeing upon the payment and the maintenance of the software goods. These areas are especially problematic in software component business. The pricing of software components is usually based on profit sharing (see Chapter 2.7) where a software component supplier gets royalties based on the final product's market success. Profit sharing also commits the supplier to work towards the final product's market success. Balance in profit sharing could be attained by mixing both one-time fixed payment and royalty-based fees. By doing so a supplier could get up front payments to cover its marketing and support costs, but possibility to collect significant royalties would also give guarantees to OEM company about supplier's continuous commitment. Support and maintenance services are important to an OEM company at least in case of COTS-components, because the OEM company may not have possibility to fix any defects they may discover in the component. On the other hand it can be difficult to contractually agree exactly about how the support and maintenance responsibilities are shared between a component supplier, an OEM company and an OEM company's end-customers.

## 2.12 Summary

30 years have passed since the first ideas of software components were implemented in object-oriented languages. Different definitions of software components vary widely depending on what context is applied. Recent literature seems to define software components more abstractly than earlier studies and definitions are in correspondence with the then dominant component technology. Shortly, a software component can be said to be an independent sub-system, which can be accessed through well-defined interfaces. The component ideology was first implemented in object-oriented programming languages followed by comprehensively standardized component models. The latest Internet-originating standardization efforts, which are manifesting the component idea, are defining how SOA ideology can be implemented using software components.

Components can be divided into three identified groups: commercial non modifiable closed source code components (COTS), modifiable commercial components (MOTS) and open source components. In order to develop software systems using components an architecture, which adapts to the use of components, must be established. A layered architecture, where service interfaces are defined between different layers, is supporting well component-based development. Overall there is no dominating technology for implementing component-based systems and it seems that component ideology will be increasingly adopted on an architectural level rather than in an individual component's implementation.

The business of software and especially software components has many characteristics, which are not present in traditional industries. These special characteristics derive mainly from the fact that a software product is an intangible asset, which re-production costs are nearly zero. Demand for software components is derived from end-user demand giving the software component business a supply chain nature. Some of the problems in software component business can be solved with the same supply chain practices as seen in traditional industries although there are naturally no physical logistical problems to be solved with software products. The main issues in software component business are in balancing between supplier and buyer interests. Suppliers and buyers have issues in software component contracting mainly because both sides eagerly protect their immaterial property rights attached to software products.

Markets for software components have emerged in the past ten years although they are not largely visible at least in academic literature. There is also evidence that companies in electronics, telecommunication and automation (ETA) industries have increasingly moved away from in-house software re-use practices towards using 3rd party software components as part of their own products.

# 3 Industry Framework

This chapter describes the industry domain where the software component market is mainly studied in this thesis. The industry framework is laid out by a telecommunication industry overview, which includes examination of current telecommunications software industry structure and current software intensive telecommunication product segments. Furthermore, this chapter explains the derivation of the demand for software components within telecommunications industry.

The whole market structure of telecommunication industry is first reviewed by identifying key players acting in telecommunications software market and their buyer-supplier relationship types. Then different telecommunication systems are reviewed to understand where commercial software components could potentially be used. Finally different component sourcing options are discussed and evaluated in different buyer-supplier relationships.

## 3.1 Telecommunications Software Industry Structure

In order to understand the market dynamics of context industry the companies which are present on the market needs to be identified. Telecommunications software industry is covering a wide set of different types of companies offering varying products and services and so any comprehensive and standard view of industry structure is hard to present. We have divided companies acting on telecommunications software markets into five different strategic groups. It must be noted that all consumer electronic device manufacturers, such as mobile handset manufacturers, and thus all software products and components used in them were excluded from our industry examination. This was done because the market where communication service providers are the end-users to software components is the main object of study in this thesis. The grouping is based on structure introduced by Leinonen (2007). We have adapted Leinonen's model considering the recent consolidations, which are further discussed in the chapter 3.2, and current telecommunications software product offerings of each strategic group. The different strategic groups, into which most of the companies seen on the market fit, are introduced below.

**Communication Service Providers (CSP)**

CSPs are companies offering communication services to consumers and corporate clients in a communication network they own and operate. The CSP group includes telecom operators who offer broadband, fixed line, mobile and other wireless communication services. CSPs form also the end-user group of the software component market.

Larger CSPs implement usually a multi vendor strategy in network infrastructure sourcing in order to lower the operational risks and risk of vendor lock-in. According to Leinonen (2007) CSPs' sourcing strategies with software systems, which they use to support their daily operations, range from a partial in-house development to full purchasing. The software for these systems is sourced from all types of software vendors seen on the market and the integration level of the purchased software solutions is depending on what kind of sourcing strategy CSP is using. These operation support systems, which are discussed in more detail in the next chapter, have evolved over time and compatibility with the legacy sub-systems have to be maintained resulting in huge integrations costs for CSPs. Example companies in this group include China Telecom, Vodafone and AT&T.

**Network Equipment Providers (NEP)**

NEPs are the old-timers of the telecommunication industry. They provide network infrastructure products, supporting software solutions and related services for CSPs. Development of network infrastructure equipment is requiring huge research investments from NEPs. After the rapid boom of mobile communication services at the end of the last millennium NEPs have been forced to adapt to the declining revenue margins in network equipment sales by constant cost-savings. One mean to cut down development costs has been the outsourcing of research and development to sub-contractors. NEPs are also providing infrastructure management services to CSPs by taking over parts of CSPs infrastructure and managing them on their own premises.

Because NEPs have a common need to cut down development costs by using not only outsourced development services, but also 3rd party software components in their products and they have a substantial part of overall telecommunications infrastructure markets, NEPs can be considered to the most important customer group for software component suppliers. Example companies in this group include Ericsson, Alcatel-Lucent and Nokia Siemens Networks.

**System Integrators and Sub-contractors (SI)**

Systems integrators are companies who offer integration services mainly for CSPs. System integrators are needed to make the overall solution to work on CSP's environment. Sub-contractors in turn are the telecommunications software contract manufacturers and they provide solutions as a service to NEPs and CSPs. Sub-contractors also offer integration services and are therefore equated here to system integrators. SI might use all other vendors to create their own offerings. Dominating SIs has global presence in the market and they operate in multiple industry sectors. Economies of scale and accumulated industry knowledge are the main competitive advantages SIs has. Apart from sub-contracting and integration services, companies in this group also offer infrastructure management services to CSPs and some of the companies have even started to offer ready-made software solutions, however it is unclear how mature and productized these solutions are. Example companies in this group include Accenture, Tieto and Sasken.

**Independent Telecommunication Software Vendors (ISV)**

ISV group include companies who offer telecom specific highly productized solutions to CSPs and SIs and are also important software component suppliers for NEPs. Descriptive to ISVs' solutions are that they are innovative and use the latest technology enablers to differentiate they offerings. Typically, ISVs are acting on market niches where they can possess some technological advantage. ISVs are substantially smaller companies than other players on the telecommunications software market and therefore they have been obvious targets for acquisitions. Example companies in this group include Amdocs, Comptel and NetCracker.

**Major Software & Hardware Providers (MSHP)**

MSHPs are the heavy-weight information technology (IT) corporations who offer the general purpose server hardware and runtime environments as well as comprehensive portfolios of enterprise software solutions to all industry sectors. As discussed earlier in this paper, the middleware solutions MSHPs are providing are increasingly used in the telecommunication domain and it could be said that top three MSHPs are dominating the middleware platform market with their integration and database management solutions. The fact that MSHPs have been the companies who have developed the first general purpose IT system including e.g. operating systems and database management software have established them to be important software component suppliers too. Through acquisitions and mergers MSHPs have also extended their software offerings to increasingly cover application layer solutions like customer relations management and business intelligence. MSHPs are, along with NEPs and SIs, big players in CSPs' infrastructure management business. Example companies in this group include IBM, Oracle and HP.

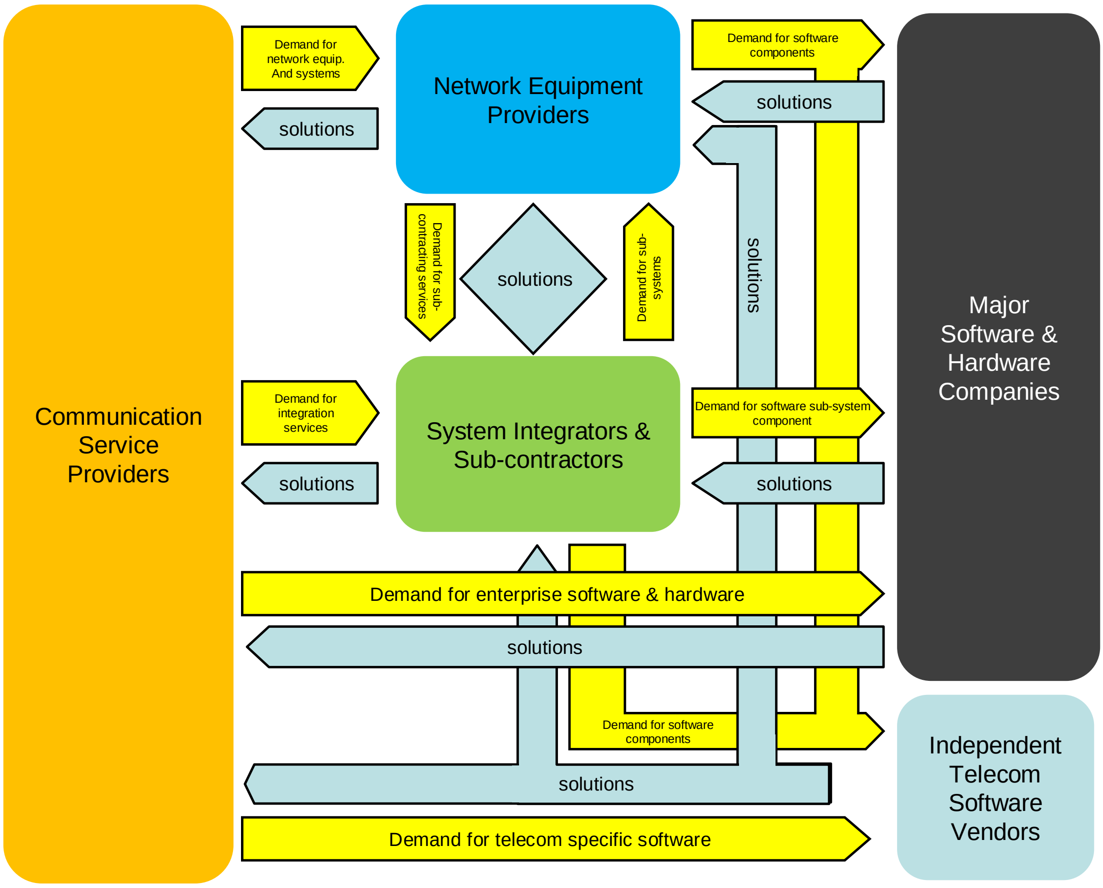



**Figure 10: Value network of telecommunications software industry**

Figure 10 illustrates the rather complex value network of telecommunications software industry including all strategic groups introduced above. From the Figure 10 can also be seen that the market structure of the software components business does not have a clear supply chain structure, because all vendors have also direct customer contact with CSP along with component supplier role. This might result in situation where component supplier and its customer are offering competing solutions to same end-user company.

In Figure 11 the global market values of different telecom equipment segment are illustrated. According to Idate (2010) CSPs worldwide invested around 131 billion Euros to their infrastructure in 2009. From Figure 11 we can see that enterprise equipment, infrastructure services and mobile access equipment segments together covered over 60% of whole infrastructure market's value. Notable in Figure 11 is that software intensive OSS/BSS market value was only 5% of total infrastructure markets in 2009. Between years 2006 and 2008 the world telecom infrastructure market grew only 4 to 5 percents annually and year 2009 showed about 2 percent decline in total market value (Idate 2010). Idate do not expect any change to grow rates in four forthcoming years.

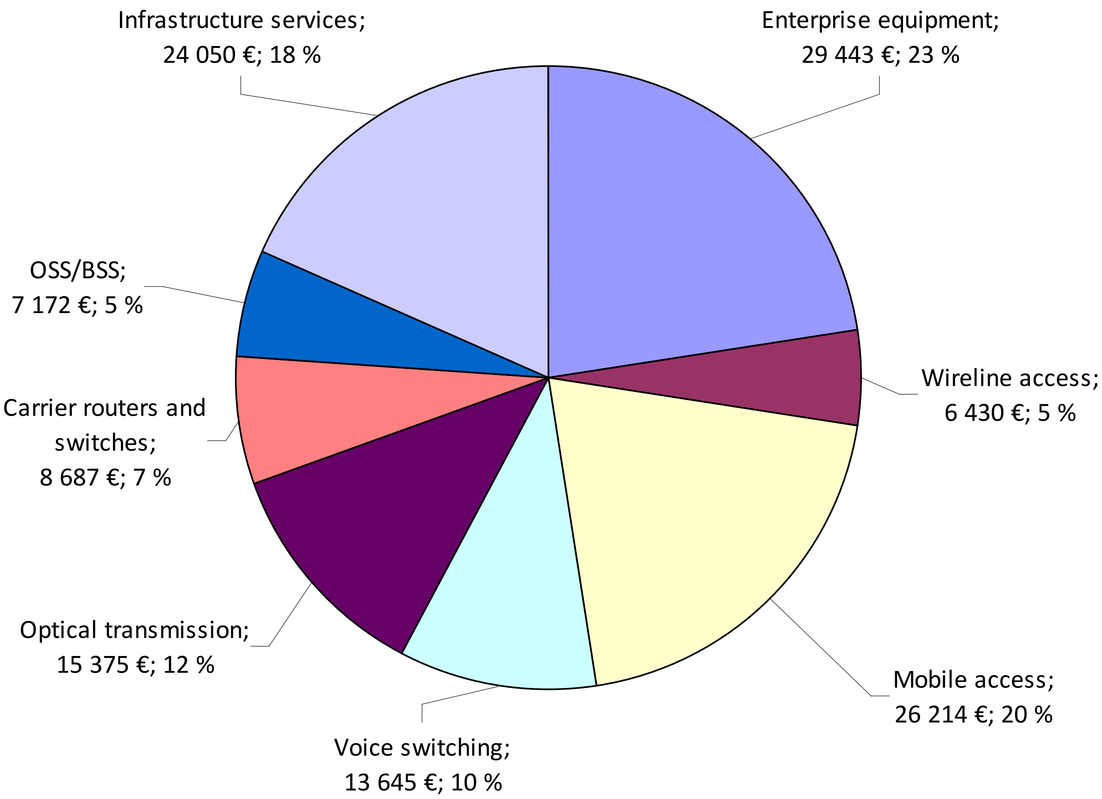



**Figure 11: World telecom infrastructure market value and shares by segments in 2009 excluding mobile handset market (Market value in million Euros), (Idate 2010)**

## 3.2 Software in Telecommunication Systems

In the past telecommunication systems, such as public switched telephone networks (PSTN), consisted of networks elements which were build using specialized and mostly proprietary hardware and deeply embedded software. Hardware and software development for network elements was typically done completely by network equipment providers. The proprietary and embedded nature of telecommunication systems has long held the market for telecommunication software very closed. A major change in the complexity of telecommunication networks happened when first digital mobile telephone networks were introduced in late 1980's. Later, soon after the Internet was introduced to consumers, the data usage in fixed and mobile telecommunication networks increased dramatically and this trend is still continuing. CSPs have currently varying sized portfolios of diverged value added service offerings, which counter tightening competition from services originating from the Internet domain. In the other hand both fixed and mobile networks are converging towards all-IP networks and at the same time CSPs are adapting more service oriented solutions into their infrastructure, such as customer behavior analysis tools. The development of telecommunication systems described ahead has constantly increased the amount of software in telecommunication systems. Also generally used computer server hardware has developed to the level that it meets the tight fault tolerance and performance requirements of telecommunication solutions. Nowadays even the most central elements of telecommunication networks, such as mobile network's home location registers, are running on generic server platforms. Moreover, the use of generic server hardware platforms has accelerated the use of common server operating systems, such as Linux. Overall it could be said that major part of telecommunication systems' functionality is currently implemented with software solutions and thus major part of telecommunication products sold to CSPs are software products.

The range of solutions CSPs have used to build and are using to operate telecommunication networks is very broad. Telecommunication system solutions could be seen to divide into two distinguished categories. All network elements, which are needed to construct a working communication network, form the first category. The second category holds the systems supporting the operation of the networks and these systems are commonly referred as operation support systems (OSS) and business support systems (BSS).

CSPs' networks could be seen to commonly have a high level structure as illustrated in Figure 12. In the Figure 12's example situation a CSP is having a common core network to support both mobile and fixed networks and they both include multiple access technologies. Moreover the different parts of the network are divided into three network layers, which are named in Figure 12 as “core”, “mediation” and “access”. Typically network elements used on access network level are more specialized than the elements used on a core network level. Generally network elements are build using vendor specific hardware and embedded software and thus 3rd party software component use is not widely visible on the markets. This could be due to the fact that very specialized software is needed in the network elements and consequently generic COTS-type of software components cannot be widely applied. Because of very differing nature of mobile and fixed network environments, the network element markets are also divided between groups of vendors concentrating on producing either type of network elements. Software developed for network element products is not widely applicable in other industries and that might be the main reason why software component markets in the network element category are not visible.

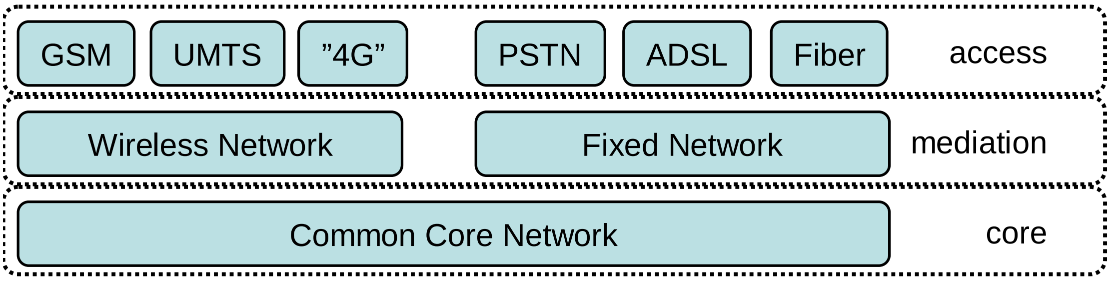



**Figure 12: High level structure of communication network**

The OSS and BSS solutions are software systems, which CSPs use to run their every-day operations. These operations or processes can be divided into three main groups: fulfillment, assurance and billing. OSS/BSS systems include multiple sub-systems, which can be divided into several horizontal functional layers based on what kind of functions they support (Tyrväinen et al. 2007). The process groups and horizontal functional layers together with 14 identified product segment of OSS/BSS software are shown in Figure 13. OSS/BSS industry is still, due to its software nature, characterized by large switching costs to customer, huge R&D investments and high system integration costs (Nieminen 2008). In fact, the integration costs are about three to four times bigger than software costs and their share of the OSS/BSS industry's total revenue has constantly been increasing (Leinonen 2007).

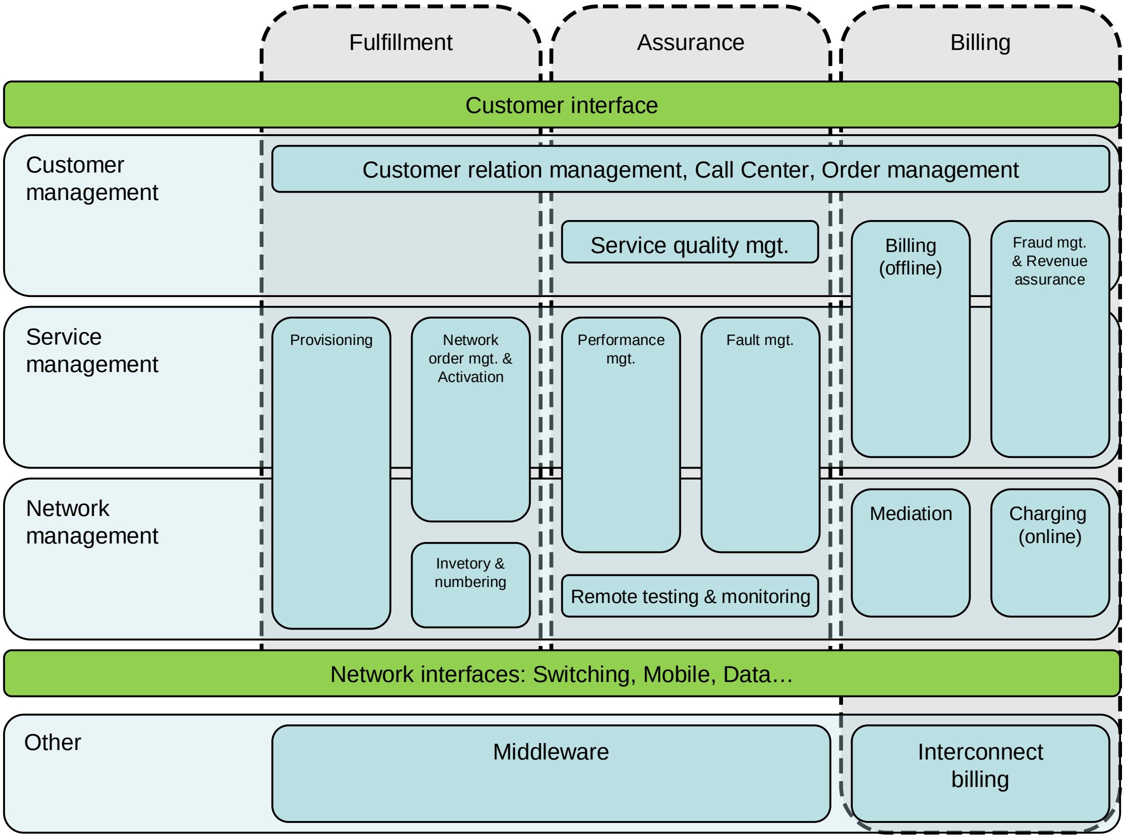



**Figure 13: Vertical processes and horizontal layers of CSP's processes (Tyrväinen et al. 2007)**

The bottom layer in Figure 13 is including solutions used to integrate the different sub-systems and provides the generic services used by other systems in OSS/BSS solution, such as database management, component integration and application servers. Leinonen (2007) mentions that there are big repetitive efforts in the area of the middleware software and NEPs and ISVs are both implementing middleware functionalities into their OSS products. There can, although, be seen an shift towards using standardized middleware software (Nokia 2004) and the findings of Nieminen (2008) are supporting this view as well as his studies suggests that middleware segment is one of the most consolidated product segment in OSS/BSS industry dominated by the two major software companies, Oracle and IBM.

Nieminen (2008) also found out that different types of vendors are dominating in closely related product segments that is, vendors' product portfolios include products, which are complementing each others. Network management layer is still mainly a business of NEPs, although evolution towards all-IP networks in this segment is bringing new entrants to this segment. According to Nieminen product segments on network management and middleware layers are found to be different from other segments. Product segments on service and customer management layers are in turn more closely related to each other when consolidation of different product categories was studied. Tyrväinen et al. (2007) found out that new OSS/BSS solutions and thus new market entries are seen more the closer to the customer interface the new solutions are directed. Customer and service management solutions used in OSS/BSS systems are not completely industry specific and therefore generic cross-industry solutions are seen on these layers.

OSS/BSS area can be seen very attractive market place for software components resulting from increasing modularity in OSS/BSS solutions, cross-industry applicability of software solutions and CSPs' and NEPs' constant need for cost savings.

## 3.3 Consolidation of Telecommunications Software Market

Telecommunication network equipment industry has recently gone through major consolidation through mergers and dominating companies have even more strengthened their positions in the market by acquiring low performing competitors during the recent economic downturn started in the end of 2008. Top five telecommunication NEPs hold currently about 70% of total equipment markets in 2009 including services and OSS/BSS solutions (Idate 2010). High level of consolidation implicates that telecommunications network equipment industry is already very mature.

OSS/BSS industry is still consolidating and has seen large number of acquisitions in recent years and these acquisitions have been mainly done by IT industry heavy-weights, such as Oracle, IBM and HP. through acquisitions these major software suppliers are leveraging their product portfolio from middleware layers to more telecommunication specific solutions (Tyrväinen et al. 2007). Although OSS/BSS industry is consolidating at a high pace, there is no single company providing comprehensive OSS/BSS solution. Several strategic partnerships have been seen between biggest NEPs, ISVs and MSHPs[1][2][3], which aim to gain advantage with exploiting each other's customer base and develop joint go-to-market actions with complete end-to-end offerings.

---

[1] Press release: Nokia Siemens Networks Enters Into Significant Partnership With IBM That Includes Transfer of Research and Development Center Activities in Munich and Berlin, Germany, http://www-03.ibm.com/press/us/en/pressrelease/22401.wss (visited 30.3.2010)
[2] Company web-page: HP & Oracle partnership, http://h71028.www7.hp.com/enterprise/cache/6606-0-0-0-121.html (visited 30.3.2010)
[3] Company web-page: Alcatel-Lucent, http://www.amdocs.com/About/Partners/Profiles/Pages/Alcatel-Lucent.aspx (visited 30.3.2010)

## 3.4 Software Component Sourcing Practices

Software component buyer has multiple options to acquire the needed piece of software from outside companies. Decision of not to make some parts of software in-house must naturally precede the choosing of component sourcing mode. The buyer's procurement processes themselves would be very interesting research topic in order to understand the decision making logics of component buyer. However, we have bound the scope in this study to only identify the different software component sourcing options. Options listed below are based on the findings of Niemelä et al. (2000) and they are as follows.

**COTS – component sourcing**

This option gives buyer a possibility to buy a readymade component, which is usually already tested on various implementations and can be included into the final product immediately. The most usual way to gain a right to use a commercial software component is to buy a license from component supplier. The implementation of commercial software components functionalities has typically needed significant development efforts from the component supplier and therefore it would not be in any way sensible for component buyer to develop the same functionalities in-house.

Niemelä et al. found out in their study of software component use in Finnish electronics, automation and telecommunication industry that the use of COTS-components is restricted to well standardized pieces of software systems, such as communication protocols and user interface components. They also find out that companies participating in the study were not using COTS-components at all in their productions or if components were used their share of the complete amount of software was only 1-5%.

**Component Sub-contracting**

Sub-contracting is a service model where sub-contractor offer software development work for customer company. The customer is usually in charge of development which is done by sub-contractor. Sub-contracting is usually priced based on amount of work done that is, how much working hours sub-contractor has spend. In the studies of Niemelä et al. the share of sub-contracting work in software development was between 15 and 40%.

**R&D Partnerships**

In this option the roles of both buyer and supplier are different from the other options. R&D partnerships in software component development aim to deeper interaction between both parties and all companies involved in partnership share common goal of developing a larger productized system. Responsibilities and roles of each participant are defined contractually.

**'Turnkey' delivery**

A 'turnkey' delivery refers to software component delivery model where component supplier is implementing a component in form of functionality, sub-system or partial system delivery defined by buyer. The price is typically fixed for the delivery and sanctions are set for delivery delays and missing features. According to Niemelä et al. 'turnkey' deliveries were rarely used in component based development although there seemed to be demand for it.

## 3.5 Summary

Five different strategic groups of companies are present on the current telecommunications software market. Besides simple software component supplier-buyer relationships, all strategic groups of companies are also direct suppliers of software and other solution to CSPs. Besides supplying software solutions to CSPs, NEPs, SIs and MSHPs have entered into outsourcing business by management pieces of CSPs' infrastructure. NEPs are also seeking for cost saving by outsourcing their development assets. This has resulted in several strategic alliances between companies in different strategic groups. Overall value network for software component market in telecommunication industry is rather complex and clear structure is hard to reveal.

Two main types of infrastructure products are merchandized to CSPs; communication network equipment and operation support systems. Network equipments are developed and sold by NEPs and their market is already quite concentrated. OSS/BSS solution market is still quite fragmented, although there has been large amount of acquisitions and the market is consolidating with high pace. Largest MSHPs are diversifying their software product portfolios to complement their middleware environment with telecommunication specific solutions. Current general purpose server hardware and operating systems already fulfill the reliability and real time requirements of telecommunication applications. This mobilization of multi purpose runtime environments in telecommunication products have also excited the use of standard 3rd party software components. Highly specialized functions of network equipment products derive more needs for customized software components, whereas common cross-industry software solutions could be used to build OSS/BSS products.

Four types of software component sourcing options were identified in the current market. Options for component buyers are COTS-component sourcing, the use of sub-contracting services, engagement to R&D partnership with component supplier or buying a ‘turnkey’ delivery.

# 4 Analysis of Software Component Business Dynamics

Till now we have discussed the basics of software component development and covered relevant parts of component business literature as well as reviewed the market of telecommunications software. In this chapter we further deepen our understanding about software component markets by first conducting a Porter’s five forces analysis. Five forces analysis is used to reveal the industry structure and current competitive environment. Based on literature review, five forces analysis and industry expert interviews conducted as part of this thesis a set of plausible future industry scenarios are created by following scenario creation method introduced by Michael Porter. All research methods used in thesis are introduced in the beginning of this chapter followed by the analysis themselves.

## 4.1 Methods Used in Analysis

### 4.1.1 Porter’s Five Forces

Porter (1980) has developed a framework to analyze competitive environment of one industry area to find out attractiveness and threats in the certain industry structure. The main goal of applying Porter’s five forces analysis is to find out the firm’s profitability in selected industry and, by understanding this, to cope with the competitive situation and ultimately to influence to the strengths of the forces’. This framework is best known as *Porter’s five forces*. Porter states that there exist five forces which affect the competition environment in the certain industry. Porter also says that if these five forces are somehow in balance then many competitors can find attractive position in industry to gain profits. But if some force out of the five dominates the industry structure, then it is more likely that just a few firms manage to get profitable returns. The strength of each force is determined by technological and economical characteristics of each industry. Industry structures and competitive environments are not stable, because changes in market economies and technology ruptures affect the five forces. By understanding the industry structure through five forces and then leveraging this information to influence to industry structures by strategic choices a firm can build a competitive advantage. This is the Porter’s view of how industry evolution happens (Porter 1980).

The five forces and their sources are presented in Figure 14. Each force has multiple determinants and feasible ones are considered in this paper to analyze current industry structure and competitive environment of telecommunications software component industry.

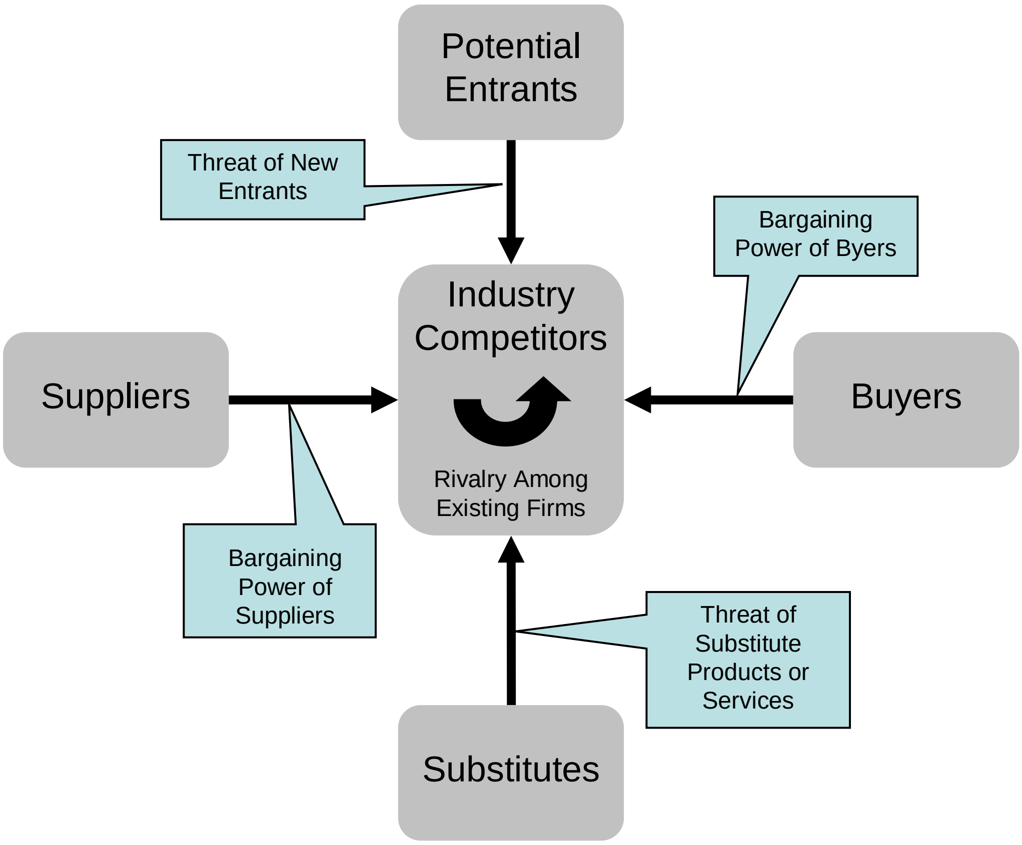



**Figure 14: Porter's five competitive forces (Porter 1980)**

### 4.1.2 Porter's Scenario Analysis

Porter (2004) has also developed a method for creating future industry scenarios based on identified industry uncertainties. The industry scenario analysis is a tool for companies to deal with uncertainties when constructing or redefining competitive strategy. Sources of uncertainties can be found both within analyzed industry and from outside world phenomena and trends. Source of uncertainty could be an upcoming major technological change in industry or an uncertainty upon changing industry regulations. Porter (2004) asserts that commonly uncertainties are not systematically taken into account in strategic planning and that strategic decisions are rather based on managers' own forecasts and represent usually conventional wisdom. Porter's industry scenarios try to tackle this narrowness manager' implicit forecasts may have and addresses uncertainties explicitly. The final goal is then to form multiple scenarios where every scenario is:

"…an internally consistent view of what the future might turn out to be." (Porter 2004)

The set of carefully selected scenarios should reflect the whole range of plausible future industry scenarios and should not include just the scenarios that are considered the most probable ones to realize. This fact points out that industry scenarios are not forecasts. Because industry scenarios are meant to influence to company's strategic decision making, Porter (2004) suggest that a reasonable time-scale should be set for scenarios. This time-scale should be in accordance with most important investments made.

The whole industry scenario analysis process is illustrated in Figure 15 showing also the two loop backs, which reveals the process' iterative nature. Porter's industry scenario analysis process starts with analyzing the industry structure to find out the most important uncertainties. This structural analysis of industry can be done using Porter's five forces framework introduced in chapter 4.1.1. According to Porter (2004) the single most important task in building industry scenarios is the identification of the uncertainties with the greatest impact on industry structure change. Porter suggests that each industry structure element should be examined and then categorized to either be constant, predetermined or uncertain element. This categorization presents the level of certainty each element respectively holds. At this point different data sources can be used to judge the level and importance of each identified element of uncertainty. Both qualitative and quantitative sources can and are used in this study to avoid overlooking sources of uncertainties that are not apparent. In this study a set of industry expert interviews were carried out to gain insights and qualitative data about uncertainties which were identified by structural analysis carried out before interviews (see Chapter 4.3).

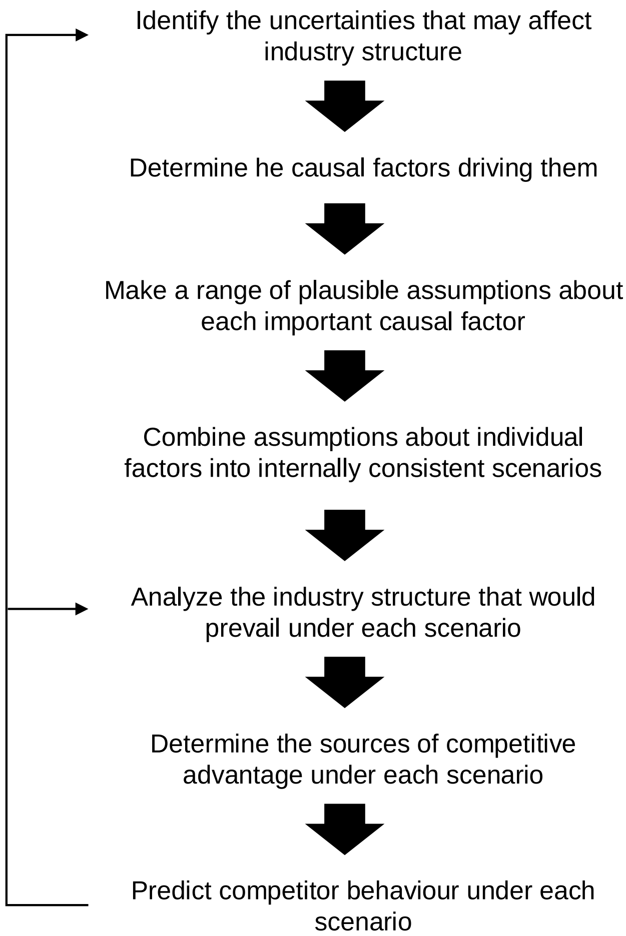



**Figure 15: Porter's scenario analysis process (Porter 2004)**

After the uncertainties have been identified they must be separated to either being dependent or independent. Independent uncertainties are elements that are independent from other industry structure factors. Dependent uncertainties in turn are elements that are dependent of independent uncertainties. The final scenarios are build upon independent uncertainties. The separation to independent and dependent elements is done by seeking and identifying the causal factors affecting each uncertainty. According to Porter (2004) causal factors can be found both inside and outside of the industry.

Once the most important independent scenario variables have been identified from the list of uncertainties a range of plausible assumptions are made about the causal factors of each scenario variable. The range of assumptions will then determine all possible outcomes of each variable. The final scenarios are then constructed from the combinations of selected scenario variables and their range of assumptions. Some combination of two assumptions might have affect on each other and thus this combination may lead to internally inconsistent scenario. These inconsistent scenarios can be ignored from the final set of scenarios. Predetermined and constant elements identified earlier in the scenario construction process are also taken into account when final scenarios are build, because they influence the industry structure in all scenarios.

The final three steps seen in Figure 15 form the analysis part in scenario analysis. First the industry structures in each scenario are determined and then the sources of competitive advantage and industry structural attractiveness are studied. The final step in scenario analysis is to predict the competitor behavior in each scenario.

As said before, the main goal of industry scenario analysis is to formulate a competitive strategy to a firm. Because the future is hard to predict, it is also hard to know which scenario will come true, if any. Porter (2004) warns that is risky for a firm to build a competitive strategy around just one scenario and in other hand a strategy to cover all possible scenarios is expensive. Porter (2004) introduces five basic strategic approaches to deal with strategy selection uncertainty. These strategies are as follows:

**Bet on the most probable scenario**

In this strategy a firm bets what it thinks is the most probable scenario and takes the risk of being wrong.

**Bet on the “best” scenario**

In this option a firm conform its strategy to support the scenario option that is most profitable or other way “best” for it.

**Hedge**

Hedging refers here to a strategy were a firm tries to find a strategy to manage satisfactorily under all scenarios.

**Preserve flexibility**

In this option a firm tries to stay flexible in strategy vice to make strategic changes when some of the scenarios start to look more apparent to realize.

**Influence**

By influencing to causal factors behind industry uncertainties a firm can influence to the odds of some of the scenario to realize more likely than some other scenario.

### 4.1.3 Interview

Interview is a good way to approach a difficult subject. Interviews can be used to collect both quantitative and qualitative research information. Verbal interaction in interviews gives flexibility to interviewee to ask clarifying questions and thus gather profound information. For example a questionnaire, where one has to choose the answer from predefined options, does not give this kind of flexibility. On the other hand interview can easily be directed towards wanted direction by interviewer (Hirsjärvi & Hurme 2008). Especially when people from industry organizations are interviewed there might be a risk that the neutrality of the information gathered is comprised due to the fear of revealing some business secrets.

Interviews can follow strictly a predefined set of questions or in the other end be completely unstructured discussions. In between these two extremes of interview structures lies a semi-structured interview in which interviewee can ask clarifying questions and do not necessarily ask the predefined questions in any fixed order. A theme interview, where only theme and some guiding questions are set, is semi-structured. (Hirsjärvi & Hurme 2008)

### 4.1.4 Industry Expert Interviews in This Study

As discussed in chapter 1.1 the motivation for this study originates from a multinational information technology corporation’s local software division’s need to better understand the software component market in telecommunication industry. Their recent acquisition of small Finnish software company gave a great possibility to get valuable industry insights of software component markets. Although the multinational information technology corporation had experience of software component business, it sell majority of its software products through transactional deals. The newly acquired Finnish company has historically collected a major part of their revenue from software component sales and had managed to establish long lasting supplier relationships with major infrastructure vendors in telecommunication industry. The future of software component business was also unclear for both acquiring and acquired companies.

A set of industry expert interviews were conducted to collect qualitative information. Four interviewees were interviewed. Interviewees were selected to cover the part of software component supply chain that was in scope of this study. All interviews were conducted in person during May and April 2010. A semi-structured interview method was chosen for this study to cover the software component ideas and constrains of current component technologies as well as business issues discussed in this thesis’ literature study. The main goal of interviews was to identify the possible uncertainties telecommunications software component market was facing currently. Interviewees were given opportunity to elaborate further on any topic brought up through questions. The complete structure of the interview is presented in the Appendix A. The list of interviewees and their role in software supply chain is presented in Table 2 below. The company names are not disclosed due to confidential requirements of the participating companies.

**Table 2: List of interviewees**

| Name | Title | Company | Date | Role in supply chain |
|---|---|---|---|---|
| Kyösti Laiho | Technical sales manager | Acquired Finnish company | 25.3.2010 | Technical sales and after sales services |
| Jorma Juvonen | Business development executive | IT corporation | 25.3.2010 | Sales |
| Joni Lehtomäki | Development program director | Acquired Finnish company | 30.3.2010 | Component development |
| Jani-Pekka Virtanen | Product manager | Major NEP | 1.4.2010 | Component buyer |

## 4.2 Five Forces Analysis

Next the five forces analysis is applied to understand the current competitive environment of software component market in telecommunication industry. Five forces framework will also be further used to identify the most important uncertainties that will likely to influence the future market structure of software component markets in telecommunication industry.

### 4.2.1 Rivalry Among Existing Competitors

According to Porter (1980) intensity of rivalry is likely to grow as certain industry matures and growth rate declines. Recent consolidation of OSS/BSS software industry is increasing the rivalry as big corporations, who have recently acquired many smaller OSS/BSS suppliers, go head to head with their competing solutions. As consolidation happen among component buyers, that will probably increase the integration level of solutions they are providing to CSPs that is, the provided solutions include more features or sub-systems packed together than before. The consolidation will also decrease the number of customers for software component suppliers thus increasing the rivalry among component suppliers as well.

Other trend discussed in chapter 3.3 was the strategic partnerships between NEPs, SIs and MSHPs. Smaller software component suppliers may face difficulties to maintain their customer relationship with large CSPs and NEPs when customers in these strategic groups are reducing the number of their suppliers and establishing strategic partnerships with preferred suppliers. As discovered in chapter 3.1, there are two different types of outsourcing seen on telecommunication industry; outsourcing of CSPs' infrastructure management to NEPs and outsourcing of NEPs' own product development to sub-contractors. Increasing level of outsourcing may also increase the vendor lock-in, when outsource partners have control over software component sourcing decisions.

As revealed in chapter 3.1 the industry structure of software component market is characterized by rather complex value network. Companies in different strategic groups are competing on CSPs' direct customer relationship with similar offerings although they might have software component buyer-supplier relationship with each other. This is particularly the situation between relationships between NEPs and MSHPs. Both NEPs and MSHPs are offering competing OSS/BSS solutions to CSPs and at the same time MSHPs are important suppliers of middleware components for NEPs. Similar situation may arise between a NEP and its sub-contractors if a sub-contractor starts to productize and sell some particular solutions, which they have earlier developed for the NEP, directly to a CSP.

### 4.2.2 Bargaining Power of Buyers

Bargaining power of buyers has naturally big influence to industry's competitive environment. Buyers are said to have bargaining power if they can force the product's price down. In addition to bargaining for lower price, the buyers can also bargain for better quality, more services and by playing suppliers against each other's (Porter 1980).

When profit sharing pricing models are used a software component buyer can bargain over lower price per component unit if it can proof supplier that the final product has abilities to sell in comparative large amounts. Buyer could also bargain over lower pricing to pass the economical risks on to supplier e.g. by not doing any advance payments and by binding the payments directly to the turnover gained from the final product's sales. The quality of software component must be proven somehow to the buyer and this is commonly done by showing references of earlier implementations and is also a subject of bargaining. Buyers usually also demand for free evaluation period to test that component they are buying is fulfilling their needs. Even before any component deal is closed a buyer may claim to have technical support and other services in the evaluation phase. The need of global, continuous and adequate support for component is apparent when final product has end-users in global scale, which is usually the case with telecommunication products. Buyer has also great power to bargain over price in the situation where suppliers have similar competing offerings, which happens usually in the case of fairly standard software components, such as database management systems.

Typically telecommunication systems have high switching cost, because the systems are large and installing new sub-systems or replacing legacy systems introduce huge integration efforts. Moreover many telecommunication products has also relatively long life cycle and this implies a long life cycle for software components the telecom product includes. Interchangeability of software components may not be apparent or even possible in all cases and when final product's features are build upon the component's unique features it increases the component's switching costs and thus gives the component supplier more bargaining advantage.

### 4.2.3 Bargaining Power of Suppliers

As software component vendors are the ones who produce the software that is resold as part of other products in the telecommunication equipment industry's value network, the industry have only few suppliers from the software component vendor's point of view. Software components can of course further be assembled from other components and particularly open source sub-components are commonly used in commercial component development. IPR holders of open source sub-component may have significant bargaining power if they suddenly choose to change their license terms. One trend happening currently on the software markets is that the ownership of IPRs of widely used open-source software and development technologies' concentrates into the hands of major software companies through acquisitions. This market phenomenon may introduce concerns about continuum of such technologies among component suppliers using them and increase the bargaining power of IPR holders.

### 4.2.4 Threat of Substitute Product and Services

If different sourcing option of software components are considered from the commercial software component supplier's point of view (see chapter 3.4), the biggest thread of substitutes comes probably from the software sub-contracting industry. The threat of sub-contracting substituting the use of commercial software component could be significant as NEPs have increasing tendency to use outsourced development.

Other completely different emerging software component distribution channel is the so called software as a service (SaaS) model. In SaaS a services, which are implemented with software, are offered over a communication network or the Internet. Infrastructure to run SaaS is usually owned and operated by SaaS provider and service pricing is use based. SaaS could potentially be a substitute for commercial software components, because an OEM company could aggregate needed functionalities from SaaS providers, integrate SaaS services with other pieces of software and offer it as their final product. Although there was no evidence found which would indicate that SaaS services are used as part of telecommunication infrastructure products.

Open-source software components are commonly seen as free substitutes to commercial component, although their use can introduce costs that are not apparent at the time of delivery and there might be no guarantee about future development or support. As discussed in the previous chapter, the risk of open-source component supplier changing the license terms and introducing some charges upon its use might lower the interest towards open-source components among component buyers in telecommunication industry. On the other hand same kind of risks originating from the difficulty to predict the suppliers' tendencies is also present when commercial component suppliers are used, but commercial suppliers have usually contractual liability about their actions, which open-source suppliers do not have, and of course a risk of losing paying customers.

### 4.2.5 Threat of New Entrants

According to Porter (1980) the threat of entry to the industry is dependent on barriers of entry. Porter also points out that new entrants increase the capacity of industry by bringing new resources into the market. New capacity in turn increases the competition which in turn reduces the profitability of the industry as the prices come down and the costs of incumbent companies may increase due to actions they take to counter the competition (Porter 1980). Porter names several barriers of entry and the ones that could be applied to the telecommunications software component market are discussed below.

For component buyer the sources of *switching costs* of component can be both technical and economical in nature. The technical switching costs were already discussed in chapter 4.2.2. Switching costs are evident when component buyer needs to do additional changes to other parts of its system product in order to make a new component from other vendor to fit the target system. Switching costs may also originate from contractual arrangements between component buyer and supplier. In case the buyer is planning to change a certain component the supplier can use every means contractually possible to increase the switching costs of component buyer. When component buyer has already used certain vendor's components in their products the buyer organization has gained intangible knowledge about component vendor's technologies and this introduces also switching costs if for example developers need to be retrained or new type of resources acquired to handle the technology of new component.

Porter (1980) has explained the *economies of scale* to form a barrier of entry if a new market entrant cannot establish such good cost efficiency that the market entry firm would be competitive in certain industry. In software industry the unit costs decline the more copies certain piece of software sells because variable production costs are zero. This would indicate that new entrant would need to make a large scale market entry to reach competitive cost efficiency level. The biggest players in telecommunications software component market from both buyer and supplier side are global companies. Big component supplier are also developing mainly general use component products, which enables them to gain higher volumes of sales for their component and thus push unit cost down. These features of software component business would imply that economies of scale may create significant barriers for entry.

The *product differentiation* as a mean to create a barriers of entry is somehow contradictory in software component context, because the idea of components itself is declaring a need to component standardization. Although the product's qualitative attributes, such as superior performance, can be thought to differentiate the component products from each other. The inner workings of component can be difficult to revise and therefore very hard to imitate. The component's unique attributes can also be protected by IPRs which in part are creating the barriers of entry for new component products.

Increased use of component-based architectures, which are based on open standards and interfaces, is lowering the barriers of entry on the software component market, because for example new component suppliers do not have to pay for accessing proprietary interfaces telecommunication network equipment used to have. Also the use of open standards, which is common among traditional industries, increases the possibility to develop cross-industry components and thus make it possible to take advantage of *economies of scale*.

## 4.3 Identifying Industry Uncertainties

After conducting a structural analysis of software component market in telecommunication equipment industry a few uncertainties were identified to exist. Global telecommunication infrastructure market does not show any significant grow rates in near future and it seems the most vigorous phase of communication network investments have passed. The whole industry has entered into an evolution phase where new technologies are slowly introduced and old ones are fading away. Telecommunication industry can be said to be in a mature state as there exists fewer firms on all strategic groups of industry participants than for example ten years ago. Software component industry has few big supplier companies, which have broaden their software product portfolios in recent years by acquiring smaller companies particularly from the OSS/BSS software industry. Big players among both network equipment vendors and IT infrastructure providers are establishing strategic partnerships with each others, which could potentially create a strategic uncertainty depending on what kind of market actions are resulting from these partnerships.

Telecommunication Software component sourcing is not a new practice for component buyers among telecommunications industry and especially NEP buyers are using recognized and well known software suppliers to deliver component to their network infrastructure products. The continuing product development outsourcing trend among NEPs could have impact on the software component market in the future. Increasing openness in telecommunication systems and new delivery methods of software components could bring new entrants to the market causing uncertainty among incumbent component suppliers. The software component paradigm can also be seen to change to be implemented on higher levels of abstraction in new component-based system development. There is not, however, a clear dominant technical design seen on software component market and new component innovations could bring disruption to the industry.

Porter's (2004) industry scenario analysis process starts with identifying industry uncertainties. The uncertain element of industry structure was mainly collected from the Five Forces analysis. Input from industry expert interviews was also used to identify the uncertainties. All identified uncertainties that could have impact on the business around software components in the telecommunication industry are collected to the list below. The main focus was on uncertainties that may affect to MSHP's business strategies. The list of all identified uncertainties is presented in Table 3.

**Table 3: Uncertain elements of structure in telecommunication industry's software market**

| | |
|---|---|
| *Intensity of rivalry* | |
| | Is the further consolidation in the telecom software market possible or is the antitrust regulations going to stop it? |
| | How ISVs will react to strategic partnerships between big NEPs, MSHPs and SIs? |

Does MSHPs see any major opportunity in software component business in the future or will they start to compete with NEPs with similar larger software based solutions, which integrates MSHP's own components in-house?

Will the outsourcing of NEPs' development increase software vendor lock-in?

What effect the outsourced telecom infrastructure management services will have on the software component market?

Will MSHPs differentiate their software component offerings by providing tailored components?

*Threat of new entrants*

Will some component standards emerge which would lower the switching costs of component buyer?

How will the possible changes in software patent laws affect the IPR management in software component business?

Will some runtime environment technology dominate in telecommunication systems?

Will sub-contractors or SIs productize their telecommunication software knowledge into software components?

*Threat of substitutes*

Will open-source software components pose threat to commercial components in telecom industry?

Will SaaS be applicable sales channel for software components?

Do software sub-contractors or SIs pose a threat to MSHPs' component business?

*Bargaining power buyers*

Will strategic partnering have effect on software component supplier selection?

Will CSPs try to decrease the number of their software suppliers?

*Bargaining power suppliers*

Will industrial IPR holders of widely used open-source components try to commercialize them?

## 4.4 Determining the most important uncertainties

Next, the identified uncertainties need to be divided either being dependent or independent of other uncertainties. Only independent elements of uncertainty can be used as scenario variables (Porter 2004).

From the list of identified uncertainties the most important independent elements were combined to form a list of distilled higher level scenario variables. Considerations about each uncertain elements importance and dependency with others was done by applying the background information collected from techno-economical review of the software component market done in the first part of this thesis as well as insights gained from the industry expert interviews. The list of independent scenario variables is presented on Table 4.

**Table 4: Independent scenario variables in telecommunications software business**

Most important scenario variables:

Level of partnering

Level of openness in component-based telecom system

Less important scenario variables:

Market impact of open source components on commercial component business

Delivery channel of software components

## 4.5 Causal Factors behind Uncertainties

In the Porter's (2004) industry scenario analysis process the next task is to determine the causal factors that drive the identified uncertainties.

The causal factors behind uncertainties can be found both inside and outside of the industry. The causal factors behind the two most important scenario variables identified previously where determined based on the full list of identified uncertainties. The identified causal factors are listed in Table 5.

**Table 5: Causal factors determining the uncertainties in telecommunication industry's software component market**

| Scenario variable | Causal factors |
|---|---|
| Level of partnering | Level of consolidation |
| | Level of supplier partnerships |
| | Type of supplier side partnering |
| | MSHPs interest to offer MOTS |
| | SIs intention to productize their solutions |
| | Competition on who has the direct supplier relationship with CSPs |
| Level of openness in component-based telecom system | Dominating component-based architecture |
| | New IPR laws |
| | CSPs requirements for future telecom systems |
| | Future telecommunication standards |

## 4.6 Dimensioning the Scenario Variables

Now, two scenario variables have been determined and causal factors driving them are identified. Before scenarios can be constructed a range of plausible assumptions about the two scenario variables needs to be made. The range of assumptions should be set so that combining of all assumption about each scenario variable do not result too many scenarios. A large number of scenarios increases the analysis effort and can result in obscuring some important strategic issues (Porter 2004).

First the range of assumptions about the level of partnering has two dimensions: low and high. Secondly the level of openness has set to have two dimensions: medium and standardized. The range of assumptions of each scenario variable is shown in Table 6.

**Table 6: Range of assumptions in telecommunications software industry scenarios**

| Scenario variable | Assumptions | |
|---|---|---|
| Level of partnering | Low | High |
| Level of openness of component-based telecom system | Medium | Standardized |

The telecommunications software market as a whole is already quite converged, but still the software component market has rather complex value networks and various types of supplier relationships can be seen. Therefore we set the two distilled assumptions about future level of partnering among industry players to assess the future outcome of recent consolidation and increasing strategic partnering among industry parties. The result of consolidation is that acquiring companies widen their product portfolios by products of acquired companies thus enabling acquiring company to use these new pieces of software to build more vertically integrated solutions. The most active acquirers are found among MSHPs. MSHPs are also the telecommunication industry's main suppliers of IT infrastructure and the combination of infrastructure and software solutions could increase their market dominance and lower their interest in partnering with other telecom vendor companies. The low level of partnering could mean also that at least the MSHPs would no longer have interest towards separate software component markets. In the other hand the divergence of telecommunication technologies creates demand for highly specialized software development assets and thus creates markets for new software components which industry incumbents are not able to cover with their offerings. Other reason justifying high level of partnering is that IT infrastructure's has become a commodity and the infrastructure products are not outstandingly differentiated nor do they have any proprietary features creating notable vendor lock-in and by partnering both NEPs and infrastructure providers could complement each others' solution offerings.

The technological progress and telecommunication system vendor's willingness to build and support open interfaces between sub-systems and components is defining the future level of openness in component-based telecom system. The two assumptions about the level of openness represent the range of different plausible levels. Medium level of openness refers to a situation where telecom systems are build using open interfaces but they could still be vendor specific making it more complicated to develop generic software components and thus increases the integration costs. The highest level of openness would mean that all parts of telecom systems are based on industry standards and wide use of readymade components would be possible. The low level of openness was ruled out from the range of assumptions, because it was perceived a characteristic of a legacy telecom systems and the industry evolution is not likely to return favoring the vendor specific proprietary solutions.

## 4.7 Constructing Internally Consistent Scenarios

The next step in the scenario building process is to combine the two scenario variables together. According to Porter (2004) the consistency check should be made when scenario variables' assumptions are combined with each other. Some combinations of assumptions may not be internally consistent, because scenario variables could affect one another. Therefore some combinations of assumptions can be left out thus eliminating some clearly impossible scenarios (Porter 2004).

No scenarios were ruled out when the two scenario variables' assumptions were combined and all resulting scenarios were perceived to be internally consistent. This left us with four internally consistent scenarios, which are shown in Table 7. The four scenarios build in this chapter are introduced and further analyzed in the next chapter.

**Table 7: Level of openness of component-based telecom systems and market structure of telecommunications software component market**

| | | Level of openness in component-based telecom system | |
|---|---|---|---|
| | | Medium | Standardized |
| **Level of partnering** | High | *Vertical solutions* | *Component revolution* |
| | Low | *Niche market* | *Fragmented market* |

## 4.8 Summary

The market of software components in telecommunication industry could be said to be fairly matured according to the level of consolidation in both component buyer and supplier groups. It was also evident that the use of software components is widely applied practice in telecom system development. Component market faces disruption mainly from substitute products and services, namely component sub-contracting and other development outsourcing. Component technologies are constantly evolving and no dominating technology was found.

The software component market in telecommunication industry was found to depend mainly on two variables, which are the level of openness the component-based telecom systems will have in future and what will be the level of partnering between all industry parties in telecommunication industry. By combining these two variables four consistent market scenarios were build to reflect the plausible future market structures in the industry.

# 5 Industry Scenarios of Future Software Component Business

The four plausible scenarios constructed in the previous chapter need further dissection, which is done in this chapter by introducing the final scenarios and conducting a structural analysis on each scenario. The structural analysis tries to discover the sources of competitive advantage each scenario could have. Based on the structural analysis different strategies of how a MSHP could respond to structural changes introduced by different scenarios are evaluated.

## 5.1 Final Scenarios

As part of Porter's scenario analysis process the constructed scenarios needs to be further analyzed to see what consequences they will pose for different competitors. This is done by reviewing the future industry structure under each scenario. Competitor behavior is also attached into the structural analysis. The competition is considered from the MSHP's point-of-view.

### 5.1.1 Niche Market

The niche market scenario is the closest one to describe the current market situation. In this scenario, the level of partnering between component suppliers and component buyers is low. The supplier and buyer companies do not see that closer relationship in software component business would benefit them in any way, which could be a result of not having any encouraging experience from such business engagements. Particularly the MSHPs who act as component suppliers are a little suspicion towards component business and consider risks higher than potential revenues. The lack of risk-taking can be seen in over protective pricing decisions and other contractual conditions set by both sides. The market structure in this scenario implies a very complex value chains and no market leader can be identified. The missing leadership in the component market means that nobody is setting the direction for others to follow.

In the niche market the variety of software components does not fulfill the needs of telecommunication industry and therefore other means, like sub-contracting, is widely used to fill this cap, although there are market segments where available software components are seen as best solutions. This is due to telecommunication industry's need for special software solutions which cannot be applied in other industries thus requiring specialized development resources and weakening the possible advantage that could be gained from the economies of scale when the same component solutions are used across different industries.

MSHPs and ISVs sell complementing software solutions, which provide functionalities that SIs and NEPs solutions are not delivering. The trade of complement solution cannot however be considered to be component business, because these complementing products are sold directly to CSPs bypassing the SIs and NEPs in the value chain. This bypassing introduces competition between component suppliers and SIs and NEPs, which in turn could hinder their willingness to establish buyer-supplier relationship in other software component market segments. Possible value-chains in this scenario are illustrated in Figure 16.

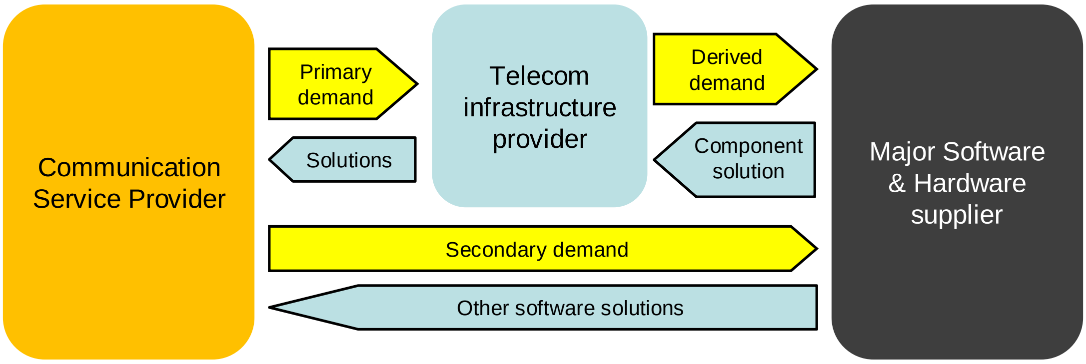



**Figure 16: Simplified value-chains in scenario: Niche market**

The main question concerning competitors is which of them chooses to be present in the software component market and which players choose not to enter or leaves completely this quite unattractive market. The other progress to follow in the niche market situation is that could some company show success in component business that others could follow by learning the best practices of successful market leader. These are probable the main questions which outcome would either initiate an evolution towards other market scenarios and structures or make the market stay in prevailing structure. The structural analysis of "niche market" is summarized in Table 8 below.

**Table 8: Analysis of scenario: Niche Market**

| | Niche Market |
|---|---|
| Future industry structure | • same as currently<br>• no clear market leader |
| Structural attractiveness | • low, no significant revenues from component sales to MSHPs |

| | |
|---|---|
| Sources of competitive advantage | • Telecom specific development assets |
| Competitor behavior | Will other MSHPs offer software components?<br>Yes / No |

#### 5.1.2 Fragmented Market

A fragmented market situation is characterized by large number of component suppliers, new market entries fuelled by standardized destination systems for software components and high quality of components. Competition for the best of the breed component solution in each software component segment has resulted in a horizontal market structure in this scenario. Component innovations are eagerly protected with IPRs, which creates the main source for competitive advantage to small component suppliers. Comprehensive standardization efforts have made component-based development the prevailing development practice among telecom infrastructure providers and MSHPs. Moreover, the standardization and introduction of common component-based architectures has encroached the competitive advantage which NEPs have had earlier with their differentiated telecom products and MSHPs have developed competing solutions in many product segments. The advancement of the component-based development has also reduced the amount of the integration work.

The high number of component supplier and complexity of value-chains leaves the market structure in this scenario fragmented and without a clear market leader. As can be seen from combination of possible value-chains shown in Figure 17 the CSPs have also a direct customership with component suppliers, as in the current market structure, in order to complement the larger solutions or build their own. What is different from current market structure and also from “Niche market” scenario is that MSHPs are not necessarily component supplier themselves at all.

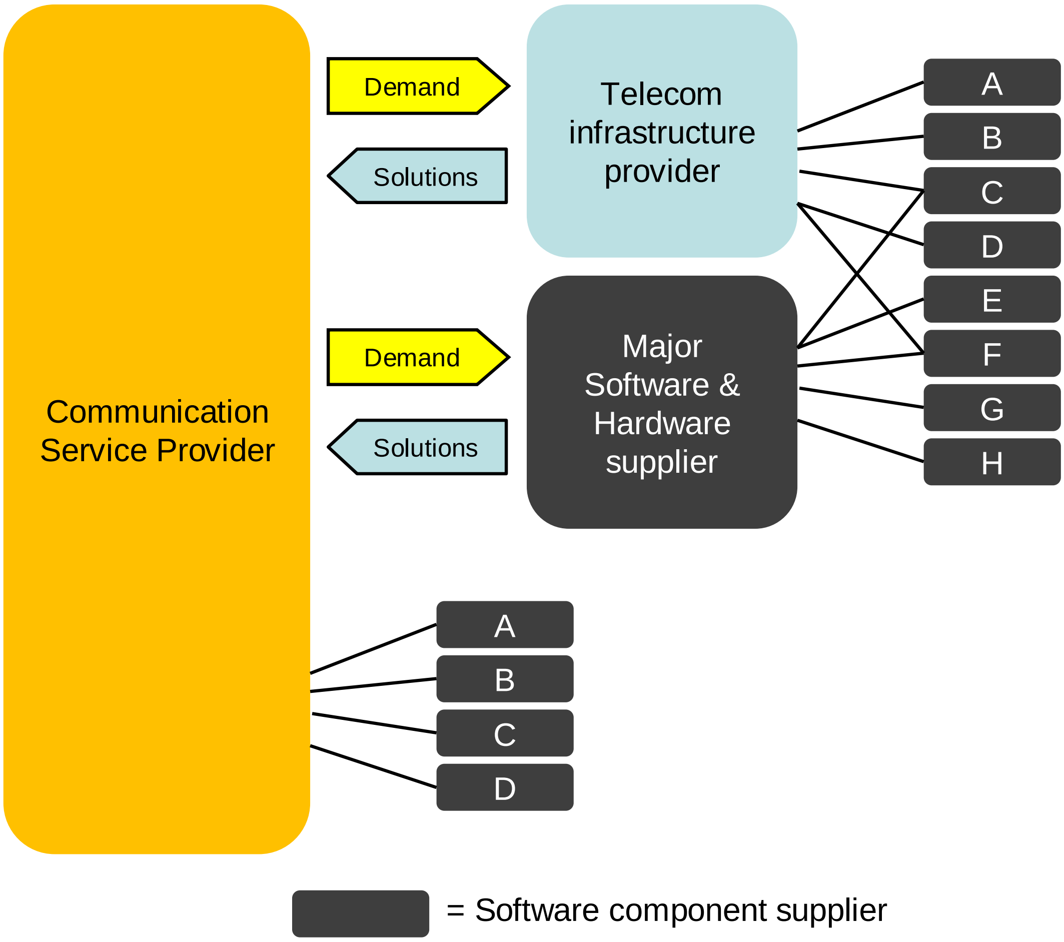
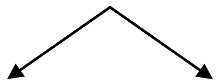



**Figure 17: Simplified value-chains in scenario: Fragmented market**

The main competitor action considered to have a great effect on the market structure changes is the competitors' eagerness to acquire smaller component suppliers. Acquisitions naturally result more consolidated market and that would imply a shift towards scenario "Component revolution". The structural analysis of "Fragmented market" is summarized in Table 9 below.

**Table 9: Analysis of scenario: Fragmented Market**

| | Fragmented Market |
|---|---|
| Future industry structure | • low barriers of entry, because of comprehensive standardization in telecom software<br>• horizontal, highly competitive market<br>• high number of component vendors<br>• high pace of consolidation<br>• no clear market leader |
| Structural attractiveness | • moderate for MSHPs, and high for ISVs and other small software vendors |
| Sources of competitive advantage | • product differentiation<br>• IPRs<br>• influence on development of dominating system architecture designs through industry consortiums |
| Competitor behavior | Will the competitors start to acquire small component vendors?<br><br>Yes No |

### 5.1.3 Vertical Solutions

The "Vertical solution" scenario could be seen to continue the recently announced partnering activities among telecommunication industry parties. In this scenario the biggest industry players have engaged into close strategic partnerships with each other. The partnering companies see the partnership as the best and only way to broaden their scope of product offerings mainly because further consolidation through mergers would be estopped by antitrust regulations. By partnering closely with each other the companies could gain competitive advantage by facing the CSP together with pre-integrated offerings. The integrated offering would create a market of vertical solutions. The partnership coalitions would be very few in number and that would greatly lower the bargaining power of CSPs' and give huge market power to solution vendors. The barriers of entry for small component suppliers would also be high due to enormous economies of scale introduced by partnerships. Thus the level of openness of component-based telecom system in this scenario would be medium, the partnering companies are left with a choice of developing sub-systems which are not compatible with other vendors' solutions outside of the partnership. This leads to lack of openness and would probably hinder the efforts of all industry parties to continue the development of open standards for software components.

In the partnerships the partnering companies would however try to maximize their own profits and that would result in conflicting interests between partners. The same issues as the software component business faces today would probably be the main concern in partnerships as well. The simplified value-chain of this scenario is illustrated in the Figure 18 and a summary of structural analysis is presented in the Table 10 below.

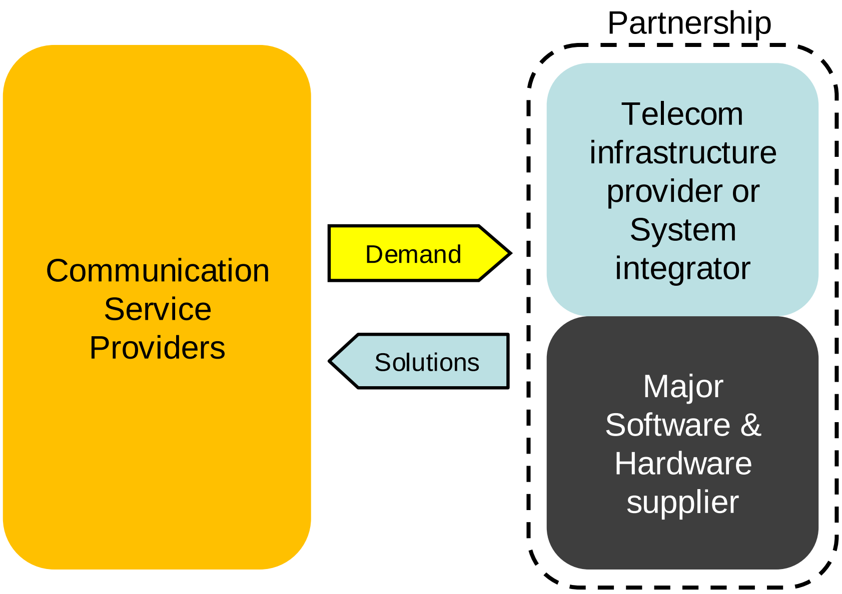



**Figure 18: Simplified value-chain in scenario: Vertical solutions**

**Table 10: Analysis of scenario: Vertical Solutions**

| | Vertical Solutions |
|---|---|
| Future industry structure | • big players in each vendor groups partnering together to complement each other's offerings<br>• shared customer territory to reduce mutual competition between partners<br>• high barriers of entry due to partially vendor specific solutions |
| Structural attractiveness | • high, optimized resource allocations due to synergy gained from partnering |
| Sources of competitive advantage | • economies of scale<br>• geographical presence of vendor<br>• large development resources |
| Competitor behavior | Who will lead the supplier partnerships?<br>MSHPs NEPs |

### 5.1.4 Component Revolution

The final scenario "Component revolution" reflects best the original ideas of software components and component business and is the closest one to the current market structure of component business seen in the traditional manufacturing industries. In this scenario there would be only one type of infrastructure system vendors in the telecommunication industry, which would all have their own networks of component suppliers. The infrastructure system vendors would have substantial bargaining power over their suppliers. This scenario setting would require that CSPs would not be able to develop any infrastructure systems in-house and they would be completely dependent on ready-made offerings on the market. The value chain in this scenario is shown in Figure 19.

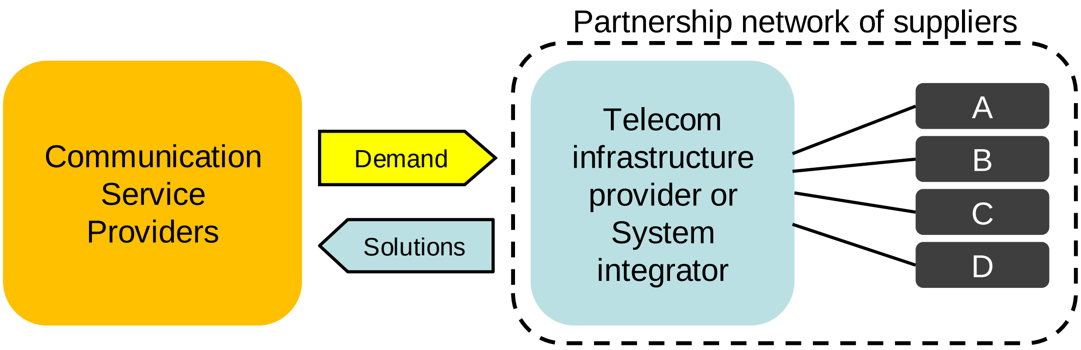



**Figure 19: Simplified value-chain in scenario: Component revolution**

The industry structure prevailing in this scenario would be attractive to independent component suppliers as they could gain substantially large sales volumes being strategically important supplier for a big infrastructure provider. The role of MSHPs in this scenario is somehow unclear. The MSHPs would most likely not collect high revenues in component supplier role in this scenario because of relatively higher cost structure than smaller component suppliers. MSHPs would also be keen to take advantage of their established customerships with CSPs rather than stepping one step farther from the end-customer interface. The only feasible role for MSHP would be to try to establish itself as an infrastructure provider.

The high level of openness would mean low switching costs for the infrastructure provider when replacing a component supplier with another one who would provide the same functionalities to the end product. The amount of repetitive work done in software development would be minimal in this ideal scenario which would eventually result in low cost solutions for CSPs.

The most important competitor behavior in this scenario is considered to be the means taken by the competitor to differentiate its products. New product features would be easy to imitate in the highly open environment of telecom systems, but complementary services instead could push the industry towards new structural changes. If we think about the next steps of industry development, we could state that the service innovations which increase the flexibility of the software component use and lowers the costs of the overall system solutions are the ones that could initiate a major transformation in software component business. A summary of structural analysis of scenario "Component revolution" is presented in Table 11.

**Table 11: Analysis of scenario: Component Revolution**

| | Component Revolution |
|---|---|
| Future industry structure | • OEM companies orchestrate a networks of component suppliers<br>• OEM companies having the best supplier networks dominate the telecom infrastructure market<br>• value networks of component suppliers |
| Structural attractiveness | • moderate to MSHPs and high to ISVs and SIs (sub-contractors) |
| Sources of competitive advantage | • price leadership<br>• economies of scale<br>• differentiation strategies<br>• component innovations<br>• low cost structure of component supplier |
| Competitor behavior | What will be the mean of differentiation?<br>→ Product features<br>→ Services |

## 5.2 Strategies for Component Supplier to Confront the Scenarios

Now that all the four scenarios has been introduced and structurally analyzed there remains the need to turn the scenarios into competitive strategies. Each scenario has its own implications and strategic decisions must follow these implications to best confront the scenario. Optimally a different strategy could be used in each scenario, but that would stipulate that a company knew which scenario would occur (Porter 2004). As the telecommunication industry has quite wide range of different telecom infrastructure products available it is highly possible that various market structures will exist concurrently. Furthermore, as the software component business is already an established market among some telecom infrastructure product segments, the timeframe for constructed scenarios is hard to set. The best guess about timeline for any of the four scenarios to come true would be 5-10 years. This estimate is partially based on the past progress of software component business. Overall the software component market among the telecommunication industry is manifold and therefore a high level of abstraction was needed both in scenario construction and in structural analysis phases. We will not make any probability estimation on which scenarios would likely come true, because we discovered that it would have needed deeper research of the causal factors behind the scenario variables. After these considerations we success that all strategic implications got from this thesis should be treated with some caution.

As discussed earlier, Porter (2004) introduces five strategic approaches under different scenarios. We now continue to take a MSHP's point-of-view also in this part of our analysis. The five different strategic options are further discussed as follows.

**Bet on the most probable scenario**

Betting on the most probable scenario is the most common strategic approach seen in practice. The betting should however be based on explicit considerations rather than implicit ones, because if the betting goes wrong the constructed strategy ends up being worthless. If company perceives some scenario highly probable and company's current resources and position on the market favors the most probable scenario, then it would be beneficial for the company to design a strategy to only support the most probable scenario.

As the software component business and the telecommunication industry are not the only areas of business MSHPs are involved, this strategy approach could be worthwhile to consider. MSHPs would most probable have enough resources to collect sufficient knowledge to make a decision about the most probable scenario and then consider what kind of investment they would make to approach the scenario.

**Bet on the "best" scenario**

In this approach the company seeks for scenario that would bring the best profits and would enable the company to best leverage its resources and its current position in the market to gain a competitive advantage in the long-run. The risk in this approach is the same as when betting on most probable scenario. If the most favored scenario does not occur the company is obviously left with an inappropriate strategy.

For MSHP the betting on the "best" scenario would mean to adjust its software component strategy so that it would take the MSHP to the direction where a component supplier would earn best profits in the value chain. This would probable mean that MSHPs would design their strategies to better support partnering. An MSHP whose current aim is to stay in the vertical software business would not bet attracted on scenarios based on open solutions as in vice versa a MSHP who is already seeing benefits in developing open solutions would bet the opposite. Either way, the strategy designed to support the “best” scenario should be in accordance with company’s strategies in other fields of the software business.

**Hedge**

Hedge strategy aims at secure sustainable profits for a company which ever scenario occurs. In hedging the formulated strategy must deal with all possible competitive environments introduced by scenarios. This would obviously need some compromises to be made thus resulting in a non-optimal strategy for all scenarios. The risk is far lower in hedging than in betting, but so are the conceivable profits.

When concerning the previously introduced scenarios of the software component business the hedging would imply that a company would have to prepare oneself to counter very contradictory competitive environments with a single strategy. On the other hand the strategy should support close partnering and independence as well as a capability to develop vendor specific and open standardized software solutions. In order to preserve the possibility to partner with some other player in the market, would mean that any competition with the same player should probable be avoided. Adoption of this strategic approach is quite unlikely as big incumbents in the overall software market all along fiercely compete head-to-head.

**Preserve flexibility**

In this approach, a company detains any resource investment in preserve flexibility to deal with which ever uncertain scenario comes true. When identified uncertainties are seen to solve themselves and some scenario starts to look probable a company can choose its strategy to fit the occurring scenario. From the MSHP’s point-of-view the flexibility could be preserved using sub-contractors or sub-component suppliers to complement MSHP’s own development resources and after the strategic choice to confront the occurring scenario is made the development resources could then be adjusted accordingly.

Preserving flexibility in the already established market and especially in the area of software product business is problematic. Software development takes time and resources and cannot be suddenly boosted after a change in strategy. Described situation would probably leave the flexibility preserving company behind its competitor who has betted on right scenario and has directed its development activities earlier towards the right scenario. This is the price that first-movers gain at the expense of companies preserving flexibility. In the software business however this price could be too high restraining companies from choosing this strategic approach. The other issue in this strategic approach is that after some scenario looks apparent to come true the selected strategic actions needs to be implemented quickly, which might be difficult for big industry incumbents because of their high level of organizational inertia.

**Influence**

The influence approach describes interactive means to influence the causal factor so that the scenario variables would have the wanted outcome. By influencing causal factors a company increases the probability that the scenario it favors will come true. The scenario which the company is hoping to occur is naturally the one that the company can pose competitive advantage.

In the software component market a company can influence the development of software component standards and technologies by participating actively in industry consortiums. Other mean, commonly used in software business to increase the adoption of new software technologies, is to release some parts of it as open source software and leave the public development community to complement and innovate it further. Whatever the means of influence are used they always introduce costs or need resources.

Apart from strictly selecting just one strategic approach to deal with uncertain scenarios there is a possibility to combine different strategies or use different strategies sequentially to find a best fitting approach (Porter 2004).

## 5.3 Choosing the Right Strategy

In our strategic assessment on industry scenarios only the last task of selecting the right strategy needs to be completed. Porter (2004) gives five factors of how to choose the right strategic approach from five options introduces in the previous chapter. These five factors are next discussed to give insights for a MSHP when creating a more detailed strategy for the software component business.

**First-mover advantage**

The software component market seems to be driven by de-facto standards and technological changes have been comparable slow thus indicating that technologically first-movers cannot gain any significant advantage of betting on the most probable scenario. On the other hand, as discussed in the previous chapter, in software component development some technological choices have to be fixed during the development, which means that some technological commitments needs to be made. The commitments should although be made so that the software component product's adaptability to the different component environments remains thus preserving the flexibility.

If the level of partnering in the telecommunication software market is considered, the first mover advantage could be crucial to gain a market leadership position. The first-mover among MSHPs to engage into large scale alliances with NEPs or SIs could potentially broaden its market access and create great barriers of entry for competing suppliers. Although it is unclear how dominating market position the first-mover could gain in the software component market characterized by partnerships and alliances.

### Initial Competitive Position

According to Porter (2004) a strategy that is designed to fit the company's initial competitive position may produce better outcome than a strategy that fits the most probable scenario. For an MSHP it would be beneficial to follow a same kind of strategy in software component business as with the other business activities of the company thus making the most of the economies of scale. Though it must be noted that the initial market position and given resources can in many cases make some strategy complete impossible for a company to implement. An MSHP for instance could potentially cannibalize its business of standalone software, which creates customer lock-in with vendor specific environment, by entering a component business with versioned product including open application programming interfaces (API).

### Cost of Resources Required

A company choosing to adjust its strategy to only comfort a single scenario requires fewer resources and has lower costs than a company that prepares itself for multiple scenarios. For example developing software components which would be compatible with multiple component technologies is naturally more expensive than supporting only one type of technology. Introducing multiple versions of the same software product can actually have huge extra development and support costs compared to added profits collected by selling additional product versions.

### Risks

All different strategic approaches have risks and Porter (2004) lists four risk factors that affect on all of the strategies:

- timing of resource commitment,
- the degree of inconsistency of strategies for alternate scenarios,
- relative probability of the scenarios and

- the cost of changing strategies once uncertainty is resolved.

It is apparent that the need of software in telecommunication systems will increase or at least stay at the current level and the question of software component market's structural changes are largely about the future structure of the supply chains. For an MSHP a "hedging" approach to avoid the risk of inconsistency strategy would not probably be profitable, because the sources of competitive advantage are the same in the scenarios in which the MSHP would have an attractive position as a component supplier. Moreover, the scenarios where an MSHP do not have an attractive position as a supplier would probably mean that the MSHP has some another role in the supply chain. Preserving flexibility both in technological and in partnership vice mixed with an influence strategy would give an MSHP a good way to reduce the risk of making early resource commitments and to influence on a relative probability of the scenarios in its own favor.

**Competitors' expected choices**

As the market of software components is just a part of the overall software market in telecommunication industry, it is quite hard to predict what kind of approach the competitors will take in component business. According to Porter (2004) petting on competitor's choices will hinder a company to choose a certain strategy and favor the selection of some other strategy.

## 5.4 Summary

Four future scenarios of telecommunications industry's software component markets were introduced and analyzed. The prevailing market structures in each scenario varied from structure alike in the current market to the market structure were telecommunication infrastructure providers are orchestrating a network of software component suppliers very much like in other traditional and older industries. MSHPs could take various roles in the software component value chain and the scenario named "Vertical solutions" in which MSHPs do not have just a supplier role was considered structurally the most attractive one for MSHPs.

When constructing a strategy for MSHP to confront the uncertainty in the software component market the company should either bet on the most probable scenario and take a risk of being wrong or preserve flexibility and influence on the causal factors behind the level of partnering among industry parties and on the level of openness the future telecom systems will have.

# 6 Conclusions

This final chapter of this thesis gives a summary of the results and discusses the relevance and validity of them. Finally some areas of a further study around this thesis' topic are discussed. This thesis presented a study of the current and future software component market in telecommunication industry. The study was conducted performing a scenario analysis which resulted in four plausible future market scenarios that can be used as a basis when evaluating the evolution of the software component market.

## 6.1 Summary of the Findings

The final outcome of this thesis is the analysis of the business dynamics of the software component market in the telecommunication industry and the scenarios constructed based on the found factors affecting the plausible changes in the industry structure.

The software component market in the telecommunication industry is already established and partially matured mode of software business. The software component business is driven by communication service providers' constant pressure to introduce more cost savings. The pressure of cost savings have resulted a need for telecom infrastructure products which have lower capital and operating costs for a CSP. This in turn has initiated a de-integration of telecom infrastructure product development creating a market for commercial software components, software sub-contracting services and software development outsourcing services. The use of generic server and operating system environments has enabled the use of cross-industry software components in telecom solutions. The software based telecom infrastructure solutions are offered by various types of industry players either directly to CSPs or through other providers using the component business model.

When the business dynamics of software component market was analyzed using scenario analysis method and the future market structure was found to depend on two independent variables; *level of partnering* and *level of openness of component-based telecom systems*. These two variables were dimensioned to both have two dimensions to reflect the plausible appearance of the future market.

The final scenarios could be seen to represent the different states of the industry in its evolution towards a structure where a larger product can be assembled using nothing but ready-made components. This kind of industry structure is already the dominating one in the other component-based manufacturing industries, such as automotive industry. The software component market is seen to eventually evolve towards the scenario "Component revolution" when the software component solutions offerings homogenize after dominating standards appear and buyers and suppliers deepen their trade relations into more closely partnerships. However, the exact path and time the evolution takes in the telecommunication industry's software component market could not be revealed in this thesis mainly because the probabilities of each scenarios' occurrence was considered hard to predict.

The first scenario "niche market" is mostly what the software component market looks like today and its attractiveness for the major software companies is perceived low. The second scenario "vertical solutions" is seen as the most attractive market structure for a major software companies, because in this scenario the major software companies could create huge demand for their software solutions by engaging in close partnerships with major telecom infrastructure providers thus creating great barriers of entry for other smaller software component suppliers. The two latter scenarios; "fragmented market" and "component revolution" represent a market structure where large scale software component use has come true. In these two scenarios the role or market position of a major software supplier is seen rather unclear and therefore the structural attractiveness is considered only moderate for a major software company.

Finally, the strategies to confront different scenarios where discussed mainly from the major software company's point-of-view. The strategic assessment on constructed future market scenarios were done on a rather high level of abstraction mainly because of the diverseness of the telecommunication industry's software market. However, two different strategic approaches were found to suit the uncertain market of software components. First suggested way for a major software company to construct a strategy to confront the industry uncertainty is to bet on the most probable scenario. Second suggested option would be to preserve flexibility and concurrently influence on causal factors so that they would result the most desirable scenario come true.

Detailed initial settings and resources of the customer company of this study and other suppliers of software components are hard to reveal in the scope of this thesis and consequently any adequate proposition of what could eventually be the best strategy for a major software company to confront the uncertainty or which market position to take could not be unambiguously given.

## 6.2 Discussion

One drawback of this thesis is that any direct market data about software component business could not be found which leaved us unclear what is the current share of the component business in the whole software market of the telecommunication industry. This affected especially the strategic analysis part of this study because we were unable to estimate the initial attractiveness of the software component market and therefore estimate the market size in different future scenarios compared to other software markets. After all, as the main interest was on industry uncertainties and their consequences, the drawback of missing market data is not considered to be significant.

Software and software component markets are very international or even global and therefore the study of the software component market was not bound to any certain geographical area although the original motivation arose from the presence of the major telecom infrastructure providers in the Nordic area. Therefore the results of this study should be widely applicable outside the Nordic and European telecommunication markets too.

As the interviewees in the industry expert interviews represented only three different companies, which two of them had already merged together, the opinions collected from the interviews might have missed some important uncertainties software component market have. The data collected through literature survey was supporting the opinions collected from the interviews and the found important scenario variables should be threaded as valid ones. If a larger group of interviewees representing all industry parties who are currently present in the software component market would have been interviewed a different set of scenario variables would most probably been collected.

## 6.3 Areas for Further Study

As this thesis has revealed several factors that will most probable change the software component market it could be beneficial to further study what implications identified uncertainties, namely partnering and openness, would have for different industry parties.

Other direction for future research would be to carry an explicit iterative scenario analysis for a named company, which would follow the same step as this study, but would use more accurate data about the case company's initial market position and available resources resulting in company specific micro-scenarios.

Also the emerging delivery model of offering software based solutions as hosted services could be studied further to understand what kind of disruption these new service concepts will bring to the software component market.

## References

Chappell D. 2007/1. Introducing Windows Communication Foundation, Microsoft Corporation

Chappell D. 2007/2. Dealing With Diversity: Understanding WCF Communication Options In The .Net Framework 3.5, Microsoft Corporation

Chung P.E, Huang Y, Yajnik S, Liang D, Shih J.C, Wang C-Y & Wang Y-M 1997. DCOM and CORBA Side by Side, Step by Step, and Layer by Layer. Available online: http://www.ece.cmu.edu/~ece749/docs/DCOMvsCORBA.pdf (visited 1.3.2010)

Cox B. J. 1986. Object-oriented programming: an evolutionary approach, Addison-Wesley

Cusumano M.A. 2004. The Business of Software, What Every Manager, Programmer, and Entrepreneuer Must Know to Thrive And Survive in Good Times and Bad, Free Press

Helander N. 2004. Value-creating networks: an analysis of the software component business, University of Oulu

Helander N, Ulkuniemi P. & Seppänen V. 2002. Understanding Software Component Markets: The Value Creation Perspective, Lecture Notes in Computer Science, Springer

Hirsjärvi S. & Hurme H. 2008. Tutkimushaastattelu: teemahaastattelun teoria ja käytäntö, Gaudeamus Helsinki University Press

Hoch D. J. 1999. Secrets of software success: management insights from 100 software firms around the world, Harvard Business School Press

Idate 2010. World Mobile Equipment Market, Idate. Available online: http://www.idate.org/2009/pages/download.php?id=&fic=503_MobileEquipment2010_Eng.pdf&repertoire=news/503_WorldTelecomEquipment (visited 10.4.2010)

IEEE 1998. IEEE Recommended Practice for Software Acquisition, IEEE Std 1062, Institute of Electric and Electronic Engineers. Available online: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=741938&userType=inst (visited 14.2.2010)

Jacobson I. & Griss M. & Johnsson P. 1997. Software reuse: architecture, process and organization for business success, ACM Press

Leinonen T-P 2007. A Case Study of Distruptive Forces in the Operations Support Systems Industry, a master's thesis, University of Tampere

Meyers B.C. & Oberndorf P. 2001. Managing Software Acquisition. Open Systems and COTS Products, Addison-Wesley

Microsoft Corporation 1996. DCOM Technical Overview, Microsoft Corporation. Available online: http://msdn.microsoft.com/en-us/library/ms809340.aspx (visited 1.3.2010)

Microsoft Corporation 2010. What Is Windows Communication Foundation?, Microsoft Corporation. Available online: http://msdn.microsoft.com/en-gb/library/ms731082.aspx (visited 8.3.2010)

Mohagheghi P. & Condradi R. 2008. An Empirical Investigation of Software Reuse Benefits in a Large Telecom Product, ACM Transactions on Software Engineering and Methodology,Vol. 17, No. 3, Article 13

Naur P. & Randell B. 1969. SOFTWARE ENGINEERING, Report on a conference sponsored by the NATO SCIENCE COMMITTEE, Garmisch, Germany, 7th to 11th October 1968, NATO

Niemelä E, Kuikka S, Vilkuina K, Lampola M, Ahonen J, Forsel M, Korhonen R, Seppänen V & Ventä O. 2000. Teolliset komponenttiohjelmistot: kehittämistarpeet ja toimenpide-ehdotukset, TEKES Teknologiakatsaus 89/00.

Nieminen T. 2008. Market Dominance and Consolidation in the Operations Support Systems Industry, a master's thesis, Helsinki University of Technology

Nokia Corporation 2004. A new R&D Paradigm: Building OSS systems with standardised middleware, Nokia Corporation. Available online: http://www.nokia.com/NOKIA_COM_1/About_Nokia/Press/White_Papers/pdf_files/whitepapermiddleware_highres_net.pdf (visited 4.2.2010)

OASIS 2010. Organization for the Advancement of Structured Information Standards, OASIS. Available online: http://www.oasis-open.org/ (visited 12.3.2010)

OMG 2008. Common Object Request Broker Architecture (CORBA) Specification, Version 3.1, Part 1: CORBA Interfaces, Object Management Group. Available online: http://www.omg.org/spec/CORBA/3.1/Interfaces/PDF/ (visited 3.3.2010)

OMG 2009. OMG Unified Modeling Language ™ (OMG UML), Superstructure version 2.2, Object Management Group. Available online: http://www.omg.org/spec/UML/2.2/Superstructure/PDF/ (visited 14.2.2010)

OSI 2010. The Open Source Definition, Open Source Initiative. Available online: http://www.opensource.org/docs/osd (visited 14.2.2010)

OW2 Consortium 2010. What is middleware, OW2 Consortium. Available online: http://middleware.objectweb.org/ (visited 1.3.2010)

Paulsson, S. R. 2005. Patenting Software vs. Free software. What should the European Union do? - Briefing Paper, Foundation for a Free Information Infrastructure e.V. available online: http://people.ffii.org/~jmaebe/epecosci0502/SoftwarePatent.pdf (visited 26.3.2010)

Porter M. E. 1980. Competitive Strategy: techniques for analyzing industries and competitors, Free Press

Porter M. E. 2004. Competitive Advantage: Creating and Sustaining Superior Performance, Free Press

Rajala R. 2001. Software business models: a framework for analyzing software industry, TEKES Teknologiakatsaus 108/2001

Seppänen V. et al. 2001. Toward original software component manufacturing, VTT tiedotteita

Shapiro C. & Varian H. R. 1999. Information rules: A Strategic Guide to Network Economy, Harvard Business School Press

Szyperski C. et al. 2002. Component Software, Beyond Object-Oriented Programming 2nd ed., ACM Press

Tyrväinen P. & Luoma E. & Frank L. 2007. Business Strategies of Telecommunications Software Vendors, University of Jyväskylä

W3C 2007. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), The World Wide Web Consortium. Available online: http://www.w3.org/TR/soap12-part1/ (visited 15.3.2010)

W3C 2010. Web Services Architecture, The World Wide Web Consortium. Available online: http://www.w3.org/TR/ws-arch/ (visited 12.3.2010)

Warsta J. 2001. Contracting in Software Business, University of Oulu

Waters D. 2003. Logistics - An Introduction to Supply Chain Management, Palgarve MacMillan

Veryard R.  2001. Component-based business: plug and play, Springer

Vinoski S. 1997. CORBA: Integrating Diverse Applications within Distributed Heterogeneous Environments, Institute of Electric and Electronic Engineers

WS-I 2010. WS-I Deliverables, WS-I. available online: http://www.ws-i.org/deliverables/Default.aspx (visited 12.3.2010)

# Appendix A: Industry Expert Interview Structure

**Background questions**

- Name, title and current job role?
- Role - considering the software component business?

**General**

- How well do you consider the original ideology and claimed benefits of component software are realized (encapsulation, inter-changeability, platform independency, better quality, shorter development time etc.)?
- Do you think that the component ideology will someday truly come true?
- Which external factors have the largest effect on the use of component software in telecom industry?
  - Next generation mobile networks
  - transformation to all-IP networks
  - cost saving objectives of communication service providers
  - next generation OSS/BSS solutions

**Software Component Technologies in Telecom industry**

- What kind of technological advantages and disadvantages there is with software components/OEM software? (platform mismatch, difficulties in integration, quality problems quality, etc.)
- How well OEM company's product development processes in general are supporting the use of OEM software?
- Will the use of the component software increase in the future and what would be the main drivers behind the trend?
- Are there some standardization efforts which would have effect on the component software use in the telecom industry?

**Software Component Market**

- How mature component software market is in the telecom industry?
- How major software supplier could gain competitive advantage in the current software component software market?
- Do OEM companies see some major risks in OEM software sourcing and use?
  - Contracts
  - IPRs
  - high switching-costs -> vendor lock-in
  - Total cost of using the component software
- Are major software vendors' offerings competitive in telecom component software market compared to for example tailor made software - why?
- Will open source software introduce challenge to commercial component software products currently or in the future?
- Will partnering of software suppliers and OEM companies increase in the future?
- Service orientation vs. product orientation, which orientation software component suppliers will take in the future?
- How the consolidation of telecom software industry will change the market dynamics in component software business?
- Will the consolidation be more vertical or horizontal?
- Which industry parties and in what areas in telecom industry will push for increasing the use of component software?
- What is your assessment on the future of telecom software component market?
- What kind of market structure would then be prevailing?