

**Veikko Seppänen, Nina Helander, Eila Niemelä &
Seija Komi-Sirviö**

Towards original software component manufacturing

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

V T T T i e d o t t e i t a

Towards original software component manufacturing

Veikko Seppänen & Nina Helander

University of Oulu

Eila Niemelä & Seija Komi-Sirviö

VTT Electronics



ISBN 951-38-5822-7 (nid.)
ISSN 1235-0605 (nid.)

ISBN 951-38-5823-5 (URL: <http://www.inf.vtt.fi/pdf/>)
ISSN 1455-0865 (URL: <http://www.inf.vtt.fi/pdf/>)

Copyright © Valtion teknillinen tutkimuskeskus (VTT) 2001

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, P.O.Box 2000, FIN-02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektronikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN-90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Technical editing Maini Manninen

Otamedia Oy, Espoo 2001

Seppänen, Veikko, Helander, Nina, Niemelä, Eila & Komi-Sirviö, Seija. Towards original software component manufacturing. Espoo 2001, Technical Research Centre of Finland, VTT Tiedotteita – Meddelanden – Research Notes 2095. 105 p.

Keywords original equipment manufacturing, software component markets, component-based software engineering, software acquisition, software supply, software booking

Abstract

This report is a deliverable of a self-funded strategic pre-study carried out by VTT Electronics in 2000 on software products developed for and used by the electronics, automation and telecommunication industries. The focus of the study was on software component products, especially on such software components that can be tailored by component suppliers or buyers before their actual use. In the report, this type of business is called original software component manufacturing, referring to the analogy of original equipment manufacturing that is well-known in other industries. Another phrase that could be used is partnership software component production.

However, there is a continuum from tailored software components that are developed by the product manufacturer itself, who is committed to component-based software engineering, to fully commercially supplied off-the-shelf components. Therefore, this report includes certain aspects along the entire continuum. An important issue related to this is to analyze how and why a supplier of tailored components would develop towards off-the-shelf products. A related question on the buyer's side is the change of an independent product manufacturer to a system integrator that puts together products with the help of fully commercial component suppliers.

One of the main goals of the report, which was produced mostly based on information available on the Internet, is to try to structure the present state of software components, businesses and production processes. The information that was gathered includes therefore both business-related and technical subjects, as well as the views of component developers, buyers, sellers, and so called brokers. The latter is emerging as an intermediary role in software component distribution. The role of a broker can also be played by a first-tier supplier, i.e. a company that manages a portion of the supplying network for the system integrator.

Regarding these roles, the information acquired from the Internet was supplemented by interviewing a few companies that mostly represent potential software component sellers. Overall, the information indicates that there is no well-organized information on original software component markets and businesses available yet. To compare, the acronym "OEM" (original equipment manufacturing) produced hundreds of relevant Web links about parts and product manufacturing in mechanical and other industries. In order to pave the way towards the structuring of the field, an integrative view is taken at the end of this report using the roles and processes of different firms involved. Based on this view, a suggestion is made for further research and development activities.

Preface

This report was prepared to make available the results of a pre-study on what we will call original software component manufacturing (OCM). The phrase was invented to describe the development of tailored software components to be acquired and used by electronics, telecommunication and automation (ETA) companies, possibly by using intermediate component brokering services. OCM was expected to locate somewhere between the commercial-off-the-shelf (COTS) software component markets and fully tailor-made, externally or internally developed software.

It appeared that there indeed are clear wishes for and emerging signs of original software component manufacturing, but the key concept of a component becomes somewhat blurred in this context. The “smaller” the developed and purchased piece of software is, the more susceptible the benefits could be. Moreover, customer-specific modifications of a piece of software may lead to losing the benefits of component-based software production in the first place. Therefore, one of the main conclusions that can be drawn from the study is that OCM would require “big” software components that are modified only “little” to be a successful strategy for the different parties involved. Many of such components would actually be independent software products, and OCM would involve partnerships between product developers, so that one party becomes a system integrator – the OCM company - to whom the others – the OCM suppliers - sell their products so that the parts fit together and serve the same end-user needs.

This report is by no means any comprehensive view to the emerging field of component-based software engineering, not even to OCM. It must be considered more as an opening only. It may even be that OCM will not become a considerable alternative for component producers and buyers – not to speak of brokers. However, even to be able to recognize some of the main reasons for such development from this study would be quite useful for both researchers and practitioners.

In closing of this preface, we would like to thank the industrial experts that we interviewed in order to gain a broader view to the information gathered from the Internet and literature. The results of the interviews became much the spices of the survey, and we hope that some flavor can be recognized in this report, too.

In Oulu, May 2001

Veikko Seppänen & Nina Helander, University of Oulu

Eila Niemelä & Seija Komi-Sirviö, VTT Electronics

Contents

Abstract.....	3
Preface.....	4
Contents	5
List of symbols and abbreviations	7
1. INTRODUCTION	8
2. OVERVIEW OF INDUSTRIAL MARKETS	10
2.1 Industrial versus consumer markets.....	10
2.2 Structure of industrial markets.....	10
3. CHARACTERISTICS OF THE OEM BUSINESS	12
3.1 Derived demand.....	12
3.2 Customized versus standard products.....	13
3.3 Just-in-time philosophy.....	14
4. OEM-TYPE BUSINESS RELATIONSHIPS	16
4.1 Nature of the buying process	16
4.2 Customer relationship management.....	19
4.3 Partnerships.....	23
5. MARKETING CHANNELS AND INTERMEDIARIES	27
5.1 Traditional channels.....	27
5.2 Virtual supply chain.....	28
5.3 Internet service providers.....	30
5.4 Application service provisioning.....	32
6. INTELLECTUAL PROPERTY RIGHTS ISSUES	37
6.1 Copyrights, trade secrets and patents.....	37
6.2 Contracting issues	38
7. THE CUSTOMER INDUSTRY	41
8. OCM – ANALYSIS OF THE STATE-OF-THE-ART	44
8.1 Business perspective	44
8.1.1 Component selling	46
8.1.2 Component purchasing	53
8.1.3 Component brokering	57

8.2	Software production viewpoint.....	68
8.2.1	Component development	69
8.2.2	Component acquisition	70
8.2.3	Component utilization.....	73
8.2.4	General component related information	74
9.	CONCLUSIONS – STATE-OF-THE-PRACTICE.....	76
9.1.1	Business drivers for original software component manufacturing	79
9.1.2	Original software component engineering.....	83
10.	PROBLEMS AND OPPORTUNITIES	85
10.1	Issues related to derived demand	86
10.2	Relationship issues.....	87
10.3	Component-related issues	88
10.4	IPR-related issues.....	89
10.5	Summary	92
11.	SUGGESTION FOR FURTHER R&D.....	95
	REFERENCES	98

List of symbols and abbreviations

API	Application Program Interface
ARA	Actors – Resources – Activities
ASP	Application Service Provider
BTC	Build – To – Configure
BTO	Build – To – Order
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial off-the-shelf Software
CRM	Customer Relationship Management
CTI	Computer Telephony Integration
EJB	Enterprise Java Beans
ETA	Electronics, Telecommunication and Automation Industries
EU	European Union
FAQ	Frequently Asked Questions
FIM	Finnish mark
IPR	Intellectual Property Rights
ISP	Internet Service Provider
IT	Information Technology
JIT	Just – In – Time
OCM	Original Software Component Manufacturer
OEM	Original Equipment Manufacturer
OMG	Object Management Group
QA	Quality Assurance
R&D	Research and Development
SAIF	The Software Acquisition Improvement Framework
SME	Small and Medium-Sized Enterprise
VPN	Virtual Private Network
VSC	Virtual Supply Chain
VTT	Technical Research Centre of Finland

1. Introduction

In the first chapters of this report the aim is to analyze the basic characteristics of the original equipment manufacturer (OEM) type of business. The basic idea is to find out what kind of business logic, processes and relationships are typical for the OEM business. The results are then used as a springboard in the subsequent chapters of the report, to cope with what is called original software component manufacturing (OCM).

One major characteristic of the OEM business is that there exists supplier-buyer relationships, i.e. business-to-business relationships, compared to business-to-consumer relationships. Furthermore, to understand the OEM business, more general knowledge of industrial marketing and management is needed. The information of the business logic of industrial markets and the OEM business presented in this report is therefore mostly based on the literature related to industrial marketing and management. However, analysis of the current trends concerning the OEM business is done to large degree based on the information available via the Internet.

When mapping the possible business logic and processes of OCM to the more general aspects of the OEM business, one important question is the way the distribution of software components is handled. Is it done directly between the OCM supplier and the customer alias the OCM company, or are there some intermediaries? This kind of marketing channel question is also discussed, including both more traditional channels and such newer concepts as ISP (Internet service providers) and ASP (application service providers).

Another important aspect related to the OCM business is the question of intellectual property rights (IPRs), in other words the ownership and contractual issues: software production can be considered as copyright industry, where the digital information that is being developed incorporates high sunk costs but almost zero re-production and delivery costs. For this reason, the owner of IPRs that bears the initial development costs may enjoy extraordinary profits. This is obviously one of the basic issues in OCM type business relationships, too.

The report addresses some of the central issues that are emerging from the information related to industrial markets in general. These issues, like partnerships, customization, ASP and customer relationship management (CRM), are interesting themes not only in the industrial markets in general, but more specifically also in the software industry. These topics are at first presented at a quite general level, but at the end of the report they are brought into the context of the OCM business aimed to serve the electronics, telecommunications and automation (ETA) industry, which is the intended customer industry for the OCM type software component business discussed in the report.

The OEM/OCM supplier term is used for component developer or sellers and the OEM/OCM company term to denote component buyers, i.e. the customer companies of the component developers. The term end user is used to describe the customers who are buying the final product from the OEM company.

The focus of the report is on studying the relationship between the OEM supplier and the buying OEM company. The viewpoint of possible intermediaries is included in the report, too. However, companies always operate in a wider network, which includes other customers for the OEM supplier and other component suppliers for the OEM company. Some of these other component suppliers may be regarded as competitors of the focal supplier. Other important actors in the network are the OEM company's end users: they may be business-to-business customers or consumers. Figure 1 illustrates the area covered by this report.

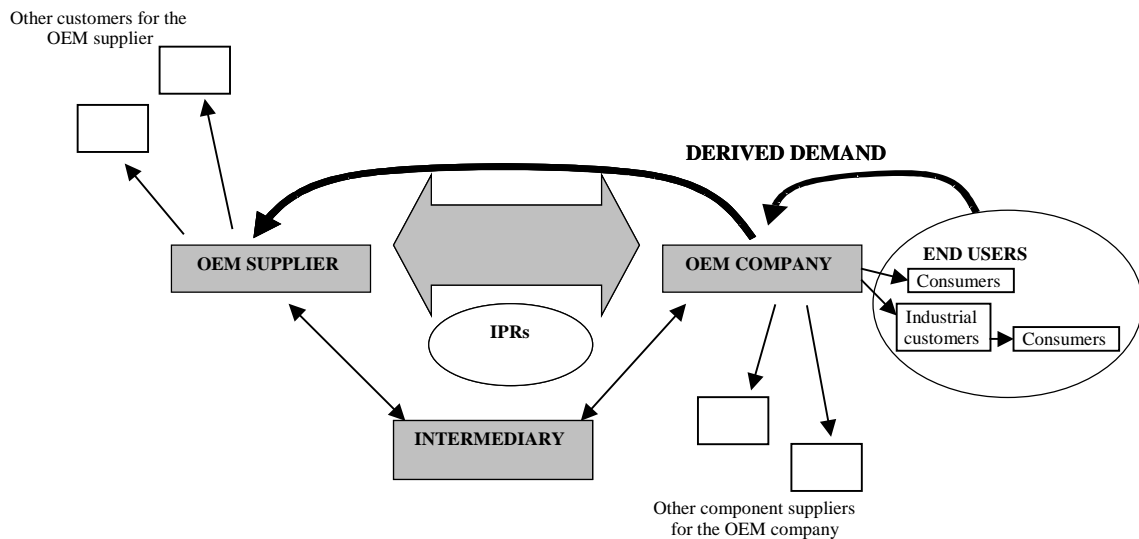


Figure 1. The area covered by the report in terms of OEM relationships.

2. Overview of industrial markets

2.1 Industrial versus consumer markets

One possible way to analyze the basic features of industrial markets is to compare them to consumer markets. These two markets differ from each other considerably (consider, for example, a candy bar vs. a paper machine), but in some cases they become quite near to each other (compare cleaning services produced for a company and for a household) [Alajoutsijärvi 1991]. One major difference between the consumer markets and industrial markets is the nature of demand; in industrial markets the demand is always derived [Hutt and Speh 1998, Reeder et al. 1991]. The consequences of this kind of derived demand are discussed more thoroughly later in the report.

Other important differences are the size of the market and the purchase volume, which are usually larger in the industrial markets than in the consumer markets. Also the products are usually more complex in the industrial markets, which causes more risks and higher prices compared to the consumer markets. When the risks and prices are higher, it is natural that more people are involved in the purchases and the role of personal selling and customer references increase. Due to the importance of personal contacts, distribution channels are more direct and shorter and customer relationships are more intensive and long-lasting in the industrial markets than in the consumer markets. [Alajoutsijärvi 1991, Auvinen et al. 1993] Consequently, the role of inter-organizational relationships is often quite dominant in the industrial markets.

An important difference between the industrial and consumer markets is also the role of marketing as part of management: in industrial markets marketing is usually the responsibility of general management, while in consumer markets it is usually carried out by separate marketing departments [Reeder et al. 1991]. However, this difference can be partly explained by the other differences mentioned above. For example, the more complex the purchased product is and the more people within some period of time the purchase affects, the more important buying is for the customer company.

2.2 Structure of industrial markets

The classic model of the structure of the industrial markets is based on the distinction between commercial organizations and non-commercial organizations, such as government institutes.

Furthermore, commercial organizations can be divided into three subcategories, industrial distributors and dealers, users, and OEMs [Chisnall 1995]. OEMs buy products, raw materials or components for the purpose of integrating them into their own products. The OEM can also be a pure product assembler, which buys all the parts outside the company [Auvinen et al. 1993]. If the OEM decides to buy also the assembly of its products, it becomes a “hollow company” that e.g. merely maintains the brand name of the products. Usually the original manufacturers of the components, in other words the OEM suppliers, and the brands of their parts are not visible for the end users. However, when the OEM company is more a system integrator than an OEM, the brand of the supplier may be visible for the end users, too. It has been argued that if the supplier company is aiming at serving the end users directly, the more adequate channel for it is the system integrator channel, not the OEM channel. [Chisnall 1995]

There are also examples of success stories of OEM partnerships. The Japanese consumer electronics industry is one such example. As a first step towards globalization, several Japanese electronics companies gained scale advantages through an extensive use of OEM agreements with partner firms in Europe and USA. Their aim was not so much to achieve any market share per se, but rather to gain global manufacturing volume and scale advantages. After the scale advantage had been established, these companies moved towards the market, pushing their own brands based on their abilities to offer high quality products at low costs – using the scale advantage in the pursuit of the scope. [Lorange and Roos 1992]

However, it can be argued whether this kind of development from an OEM supplier towards serving the end users is possible in the case of “small” product parts. It can be assumed that this would be more likely in case of “bigger” parts of the OEM product, i.e. when the parts can be regarded as rather independent products in the eyes of the end users. Another problematic issue is the question of vertical markets. Usually the OEM company has control over the whole vertical market, and it may be very hard for the supplier to get a considerable piece of these markets.

One possible way from the OEM supplier’s point of view to handle this kind of problem is to use a horizontal approach to vertical markets. This means that the supplier builds strategic alliances with other component suppliers, not only with OEM companies. With the help of the alliances the supplier may focus on a single kind of customer and product set, while simultaneously leveraging other OEM suppliers’ vertical market experience and business relationships (cf. e.g. [Anon. 2001a]). Together the suppliers may offer a better solution for the end users than the original OEM company.

3. Characteristics of the OEM business

3.1 Derived demand

Demand for industrial products is always derived from the ultimate end user demand [Hutt and Speh 1998]. For example, if the end users do not buy any shoes, there is no need for machines or raw materials used in the process of shoe manufacturing.

Due to the fact that derived demand is the single most important force in the industrial market [Reeder et al. 1991], it has a major influence in the business logic of OEM suppliers. In fact, it forces them to build marketing and management strategies that take into account both the direct customer, which is the OEM company, and the indirect customers, which are the end users. This is not an easy task, because there can be several stages between the supplier and the end users [Chisnall 1995], and the stages make it hard to understand the needs of the users. If the end users are consumers, it is difficult for the OEM supplier to develop a marketing strategy that serves both the industrial market and the consumer market. The supplier should in fact be an expert both in offering its products as components and the use of these components in a larger product – i.e., in the core business of the OEM company [Chisnall 1995].

However, some means can be identified which may help the OEM supplier to struggle against the problems raised by the derived demand. Firstly, for gaining a better understanding of the needs of the end users, the supplier should make the supply chain visible. This can be done, for example, by developing the so-called virtual supply chain (VSC) approach [Bairstow 1999, Anon 2001b]. The principles of this kind of supply chain are discussed later in the report. When a better understanding of the needs of the end users is gained, the supplier can direct some marketing actions towards them. For example, it is usual for industrialists to advertise directly to the end users, even though they do not themselves do the selling [Reeder et al. 1991]. By this kind of advertising the OEM suppliers try to develop a pull-effect in the distribution channel.

If the OEM company serves the consumer market, the supply chain is often visible also for the suppliers. On the other hand, if also the OEM company serves industrial buyers, the supply chain becomes more complex to handle not only for the suppliers, but also for the OEM company itself. In this case, the OEM company faces the same kind of problems of derived demand than its suppliers do.

Figure 2 illustrates the derived demand in the industrial markets [Chisnall 1995] using a simple model of the production flow. As can be seen from the figure, components are usually needed at the beginning of the production flow, just after raw materials.

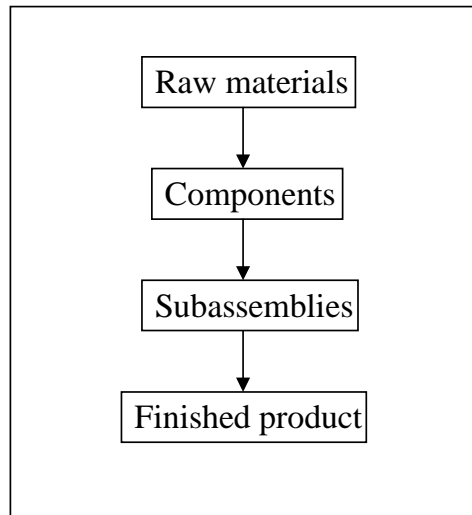


Figure 2. Pattern of derived demand for industrial products [Chisnall 1995].

Besides the derived demand, also joint demand and cross-elasticity of demand characterize industrial markets. Joint demand occurs when one product requires the existence of other products to be useful. Cross-elasticity of the demand stands for the responsiveness of the sales of one product to a price change of another. In other words, the two products are substitutes to each other. [Reeder et al. 1991]

3.2 Customized versus standard products

Customized products are to some extent designed or adjusted to satisfy the needs of specific market segments or even individual customers. Therefore, one of the biggest differences between customized and standard products is the need of communication between the product developer and the customers (cf. [Heikkilä et al. 1991] that discusses this question in the case of software products). Customized products require specific skills from the suppliers [Chisnall 1995], the requirements of the customers have to be viewed much more carefully than in the case of standard products. This also means that the relationship between the product developer and the customer is much closer and interactive. The need for close interaction builds customer loyalty, and usually a high degree of mutual dependency arises [Chisnall 1995].

There is often a need for some kind of customization in the OEM business, and the role of the relationship between the supplier and the OEM company is therefore an important aspect to take into account. For example, if some end users are demanding customized products, the OEM company must identify these needs and analyze together with its suppliers, which kind of product modifications are needed .

The OEM company cannot, however, usually expect that its suppliers would make the necessary part modifications alone, because of their distance from the end users. In other words, the OEM company and the suppliers should have a close enough relationship to design and implement the part modifications together. However, whilst considering the part or component markets, there is usually a need for both standard and customized components. If there is a greater need for customized components than for standard components, requirements for closer co-operation between the engineering departments of the OEM company and the suppliers arise. If standard components play a major role, close interaction between the buying department of the OEM company and the selling departments of the suppliers is emphasized [Chisnall 1995].

Whether the components are standard or customized, it would be useful for the OEM company to categorize the needed components for the purpose of parts sharing. Two useful categories are components with strong influence on product quality and components with only weak influence [Fisher et al. 1999]. This kind of categorizing helps the OEM company to control its production and the quality of the final product more carefully than just the difference between standard and customized components.

From the OEM company's point of view there is also a difference between whether the component is acquired as a part of a product platform or an independent sub-assembly, i.e. a product itself. The difference is due to the fact that a product platform is not a product but rather the core of a product family. In fact, the product platform supports the design of product architectures that spawn one or more product families. For the OEM company to be effective, a product platform should be implemented before any customized products are developed (for strategic issues of software platforms, see e.g. [Sääksjärvi 1998]). The product platform could be regarded as something between the subassemblies and finished products, referring to the pattern of derived demand shown in Figure 2.

3.3 Just-in-time philosophy

One important characteristic of the OEM business is the central role of the just-in-time (JIT) philosophy. Originally, JIT was developed and implemented in the Japanese OEM business, but within a few years it was adapted by European and American companies, more or less successfully. The JIT strategy affects the supplier – OEM company relationship quite significantly, because it demands close cooperation between the relationship parties. JIT can be seen as a strategy of procurement that embodies three closely interrelated issues.

First, the purchase quantity is limited to the exact quantity of components required for the planned manufacturing or assembly. Second, the exact quantity is planned to arrive at a precise time in order to eliminate the need for warehousing. Thirdly, JIT embodies the benefits of carefully planned utilization of transportation vehicles. In some situations it also embodies the use of specialized equipment for the facilitation of efficient materials handling at the delivery destination. [Bowersox and Cooper 1992]

Considering these three central issues of JIT from the viewpoint of software products and components, it is easy to see that the first two do not really matter due to the close to zero reproduction and storage costs – except the need for building the final product on time. The third one may be highly relevant, however, for example to make sure that a software component offered by a supplier fits into the buyer's product platform.

Implementing of the JIT philosophy also brings some other concerns to the supplier – OEM buyer relationship. On the supplier side, one common question is the sudden changes in the customer's demand schedules. It is not unusual that suppliers hold safety stocks in order to compensate poor forecasting on the OEM company's side. Suppliers may therefore feel that OEM buyers have shifted their inventory responsibility to the suppliers, and the suppliers have become liable for any failure in the supply chain.

It has indeed been pointed out that more adequate forecasts for the future parts requirements and delivery schedules should be developed in OEM relationships. [McIvor and McHugh 2000]. This question can actually be traced back to the problem of derived demand discussed in the previous section. Considering it especially from the viewpoint of OCM suppliers serving ETA buyers, it would mean that the need of software components is derived from the needs of the end users of the electronic, telecommunication and automation products manufactured by the OEM companies.

Another concern, from the OEM company's side, is to figure out how to train and educate their suppliers to successfully adopt the JIT strategy and in this way to make sure that the suppliers are correct in terms of time, quantity and quality [Karlsson and Norr 1994]. One way to solve this problem, in the first place, is to reduce the number of suppliers and to extend the time of contracts. This is, however, in contrast both with the need to have second sources for critical components and the aim at pushing some of the effects of changing end user demand to the supplier network.

4. OEM-type business relationships

4.1 Nature of the buying process

In spite of the name “buying process”, which refers only to the OEM company’s side, the buying process should be viewed from the point of view of both parties, the suppliers and the OEM company. In fact, when adapting the relationship point of view, both parties of the relationship have to be taken into account.

The supplier (the seller company) and the OEM (the buying company) may have many uncertainties, when they interact along the buying process. The supplier’s uncertainties are related to the questions whether its capacity and procedures are adequate to serve the OEM company. The OEM company’s uncertainties include, for example, the question whether it can express its needs well enough. This kind of need identification is related to the question of the market and end-user knowledge the OEM company possesses.

On the other hand, both parties have abilities that influence the buying process. These abilities include the problem solving ability and the transfer ability on the supplier side, and the demand ability and the transfer ability on the OEM company’s side. The parties try to influence the uncertainties of the other party with the help of their own abilities, either in terms of trying to strengthen or weaken the uncertainties of the other party. The concepts of uncertainties and abilities are presented in Figure 3. [Ford et al. 1998]

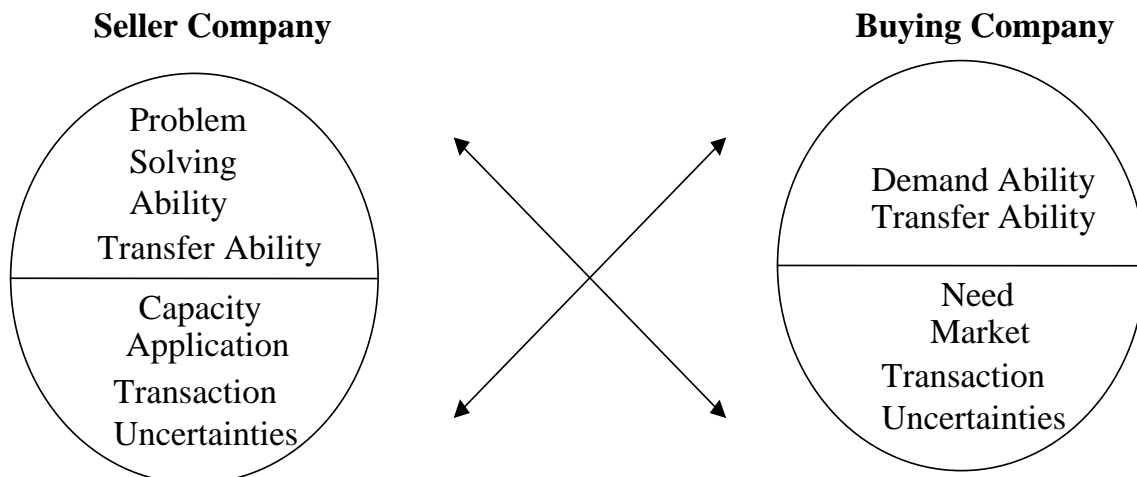


Figure 3. The uncertainties and abilities of buyers and sellers [Ford et al. 1998].

Whether the parties of the relationship try to strengthen or weaken the uncertainties of the other party, depends on the current market situation, on the basic characteristics of the relationship and sometimes even on the phase of the buying process. For example, if the relationship is close and cooperative, the aim is usually to help the other relationship party to solve the problems behind its uncertainties. On the other hand, if the OEM company tends to make its suppliers to compete against each other, it is probable that the suppliers try to influence the situation by strengthening their uncertainties.

Identifying of the activities carried out by the parties during the buying process helps to structure the relationship. Figure 4 divides the buying process for this reason into three phases: preparatory, active and evaluation phase. Each of these phases includes different actions taken by the seller and the buyer. [Kotsalo-Mustonen 1995] The phases therefore represent the purchasing situation itself, rather than the more overall development of the relationship between the supplier and the OEM company.

THE PERSPECTIVE OF THE BUYING PARTY	PHASE	THE PERSPECTIVE OF THE SELLING PARTY
Problem recognition	1 Preparatory	General marketing activities
Determination of characteristics		Target group selection
Description of characteristics		Product development
Search for sources	2 Active	Prospect scanning
Analysis of potential sources		Prospect qualification
Proposal evaluation and selection of suppliers		Proposals and argumentation
Selection of order routine		Modifying details in the contract
Performance feedback and evaluation	3 Evaluation	Implementation and delivery

Figure 4. Phases in the buying and selling relationship [Kotsalo-Mustonen 1995].

The question of whether the purchase is a so called new task buy, modified re-buy or straight re-buy also has an influence on the buying and selling behavior of the relationship parties. For example, the straight re-buy is quite simple, but the new task buy needs a great deal of information exchange and negotiations, because the case is considerably different from the past experiences. [Chisnall 1995]

From the supplier's point of view, one major question is to identify the buying decision group of the OEM company. This group includes all the individuals, who are affecting the buying decision. Usually the group is an informal decision making unit, consisting of individuals from different departments. The group's aim is to gather and analyze the information needed for the buying decision. There are usually different kinds of roles in the group that the supplier should be aware of. These kinds of roles are, for example, gatekeepers, influencers, deciders, and buyers. [Alajoutsijärvi 1991]

The way the buying process proceeds and is organized is influenced by the nature of the mutual relationship, i.e. how close to and dependent on each other the relationship parties are. The interdependence between the suppliers and buyers is furthermore influenced by the market structure, which determines the available choices and sources of power of any prospective partners in the market.

Figure 5 illustrates the influence of the market structure, in other words the number of existing suppliers and buyers. For example, in concentrated markets, where only few suppliers and buyers exist, there is likely to be more partnering cases than in the other three market situations. Less equally balanced supplier-buyer relationships, where there is either a buyer dominance or a supplier dominance, are characterized by one party exercising power over the other. In the last situation, where there are many alternative suppliers and buyers, the mutual commitment between suppliers and buyers is usually low and there is a greater freedom for both parties to change their partners. [Cunningham and Homse 1990] It should be noticed, however, that the number of potential buyers increases, if the suppliers operate in more than just one industry. In other words, the suppliers can decrease the dominance of few big buyers by leveraging their markets also to other industries.

		NUMBER OF SUPPLIERS	
		Few	Many
NUMBER OF BUYERS	Few	Mutual interdependence	Buyer dominated
	Many	Supplier dominated	Relative independence

Figure 5. Market structure vs. relationships [Cunningham and Homse 1990].

The automotive parts industry is a good example of the model shown in Figure 5. As is well known, there are a few big automotive OEMs at the worldwide level. These companies, like General Motors and Toyota, have a major role in the whole industry. Because there is only a few of them, the car manufacturing industry is dominated by the buyers, looking from the parts suppliers' point of view. However, the situation is different in cases where a supplier has a presence only in the automotive industry and where it also operates in other industries. In the latter case it has much more potential customers and is not so tightly dependent on the car manufacturing companies. For example, in the North America the biggest firms in the car parts industry usually also have a major presence in other markets, too, but in Japan, the major parts manufacturing firms are often highly automotive dependent. [Lamming 1993]

The interdependence issue affects the interaction process between the OEM company and its suppliers. In the OEM market there is usually the question of adaptation regarding the quality and functionality of various components. Usually the OEM wants that the supplier would modify, develop and improve its components in different ways. On the other hand, to some extent it is desired that the OEM company accepts standard components which the supplier would make available to its other customers, too. [Axelsson and Håkansson 1990] In most of the cases, the OEM companies are much bigger and more global than their suppliers. The adjusting party is therefore usually the supplier, not the OEM company. However, the situation changes, if the shared value creation thinking is adopted, as will be discussed later in this report.

4.2 Customer relationship management

One emerging trend in industrial management is the growing importance of customer relationship management (CRM). Customer information has traditionally been stored in multiple systems across the company, due to the fact that the old enterprise systems were designed to support specific business processes, not specific customer relationships. Also the fact that customers are served through multiple channels leads to a situation, where customer information is dispersed all over the company. To compare, the basic idea of CRM is to collect, manage and store customer data centrally and in this way to help the company serve its customers better. By defining the CRM processes and implementing a CRM system companies can ensure that the customers are served seamlessly across all channels. [Storbacka et al. 1999] The CRM approach is based on an assumption that every customer has its own processes through which it creates value. When companies have centralized information about their customers, they have better possibilities to understand the customers' needs.

The customers' value creation processes are established to achieve some business goals and mission. Ultimately, this means that customers always measure value in relation to their own goals. If a supplier wants to build a good relationship with the customer, it has to have a thorough understanding of the customer's mission, goals and business drivers. [Storbacka et al. 1999] In other words, to build up a long-lasting and successful customer relationship, the supplier needs to achieve a deep understanding of the activities by which the customer creates value for itself.

However, this kind of customer's problem solving should be done in a profitable way, at least in the long run. As Anderson and Narus [1998] have pointed out, besides recognizing those actions by which the supplier can create value for the customer, it is also crucial that the supplier can perform these actions in an economically profitable way. Solving of the customer's problems should therefore be based on the supplier's own core competencies, so that the supplier can solve the customer's problems in a profitable way without any big sacrifices. This is because core competencies are something that do not deteriorate as they are applied and shared; in fact, they even grow when they are applied [Prahalad and Hamel 1990].

Therefore, the basic idea of CRM is to first identify the customer's value creation process and then to learn to support it in a profitable way so that both the supplier and the customer can benefit from the relationship. In the OEM context, the supplier should at first familiarize itself with the OEM company's value creation process, because the OEM company is its direct customer. However, because of the derived demand, also the end-users' value creation processes play an important role. If either the OEM company or the supplier does not understand the needs of the end-user, the joint business is not going to be successful.

The purpose of the definition of the customer's value creation process is to better understand the concerns related to the customer's business activities. Another advantage of understanding the customer's value creation process is that the supplier can more easily communicate with the customer concerning the value that it can create for the customer [Anderson and Narus 1998]. As MacStravic [1999] has pointed out, it is important to communicate with the customer in a "right" way, so that the customer can realize the value that the supplier is able to create. When the supplier understands that different customers have different kinds of value concerns, it can classify its customers into different customer types based on their value propositions and processes. In practice, different customers need different kind of support in the value creation processes. In every customer relationship at least three types of exchange are included: the exchange of knowledge, emotions and actions [Cross and Smith 1997, Storbacka and Lehtinen 1997].

This means that the supplier can gain a “share of the customer's thoughts” by giving him enough information, a “share of his heart” by shared values and branding, and a “share of his wallet” by right actions. Furthermore, the customer relationship can also be seen through different phases. Different models of the relationship phases have been presented, for example, by Dwyer, Schurr and Oh [1987], Ford [1980] and Halinen [1994]. One possible way for analyzing such phases is the model presented by Storbacka and Lehtinen [1997]. According to the authors, the relationship has the following phases: it is being born, enabled or ended. Relationships create most value for both parties at the enabled phase. New customer relationships are, however, important for the supplier, because new customers will usually bring in new ideas and need for new actions, and in that way force the supplier to develop its business. By analyzing the customer relationships, which are already ended, the supplier can – for example - learn which kinds of mistakes have been done.

Customer relationships should also be analyzed at different levels. According to Storbacka, Kaario and Sivula [1999], three levels can be used to structure customer relationships: the contact level, the relationship level, and the overall interaction level. The contact level refers to the purchasing situation. At the relationship level the customer evaluates how well the chosen product or service supports its own value creation process. At the overall level the customer is interested in understanding how well the whole relationship with the supplier supports the accomplishment of its goals and mission. Unfortunately, it is common that many suppliers only think about their success at the contact level, while in fact they should concentrate more on the relationship level and the overall level. This has been pointed out by many relationship marketing experts, including Grönroos [1997], Gummesson [1997], and Morgan and Hunt [1994].

The supplier can benefit from the customer relationship only when its actions are based on its own core competencies. Another important point is that the supplier should evaluate which customer relationships are the most valuable ones that should therefore be protected and nurtured the most. Storbacka, Blomqvist, Dahl and Haeger [1999] suggest that this kind of evaluation should be based on the so called learning value, reference value and strategic value of the customer, besides the economic return of the customer relationship. After these evaluations, the customer relationships can be classified into those that should be protected, changed, or developed.

Although CRM seems to emphasize the customer's side, its focus is in fact on the relationship, and in this way it includes both the supplier company's interests and the customer company's interests. Figure 6 illustrates the processes that are linked together, when the customer company and supplier company interact.

The customer's processes are characterized by six phases: strategic management, business development, analysis and choice, purchasing, core business activities and follow up. Also the supplier's processes are characterized by different phases. In fact, Figure 4 shown earlier in this report illustrates only one phase of these processes, the purchasing phase, while Figure 6 presents the other phases as well. Every phase includes different kinds of actions and contacts between the interacting companies. For example, the phase of strategic management can include partnership deals and seminars/workshops and the phase of purchasing/selling offerings, correspondingly, contracting and action planning. [Storbacka et al. 1999] The basic idea of this model is to make the relationship more visible and easier to manage, by identifying the actions that will occur between the customer and the supplier.

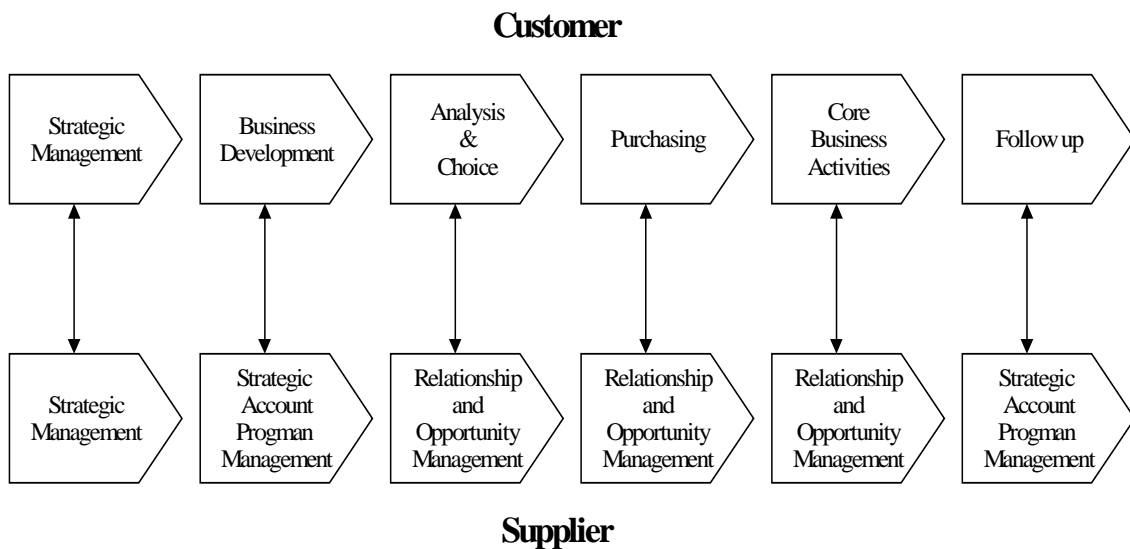


Figure 6. Generic strategic account management zipper [Storbacka et al. 1999].

Besides the activities that take place between the two parties of the relationship, in other words the actors, also the resources that are exchanged during these activities play an important role. The question of resources is taken into consideration in the so called ARA model of industrial networks [Håkansson and Johanson 1992], Figure 7. In short, the model suggests that every network of relationships includes at least two actors, who perform different kinds of activities in which multiple resources are exchanged.

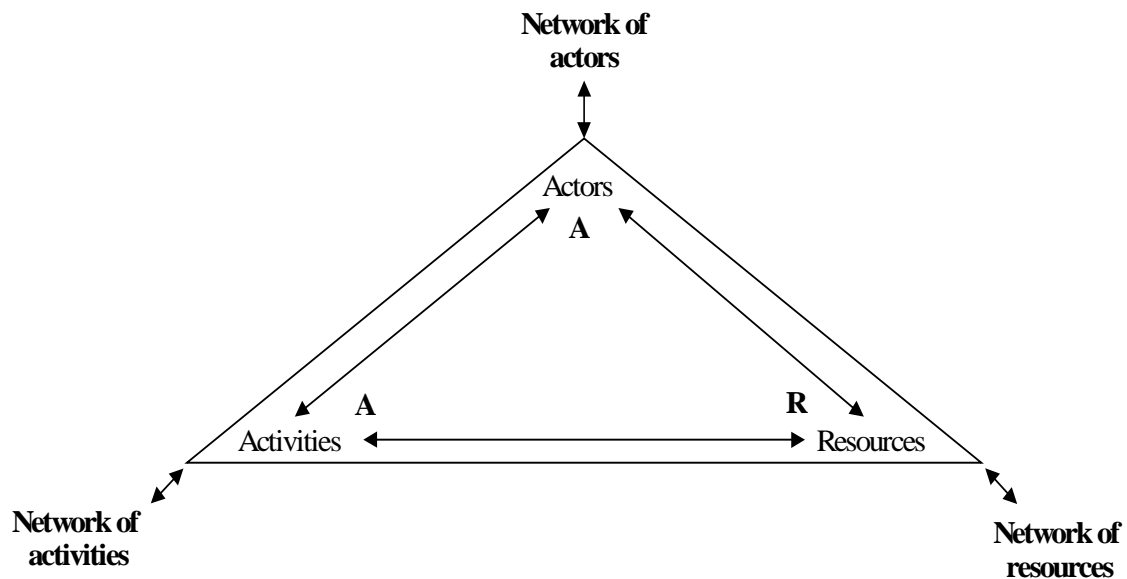


Figure 7. The ARA model of industrial networks [Håkansson and Johanson 1992].

However, this model of industrial networks does not only concentrate on the relationship between two interacting companies; it broadens the viewpoint also to other actors, activities and resources related to the dyad. This kind of network approach is relevant in the OEM business due to the fact that there are usually many suppliers and subcontractors that provide components to the final product assembled by the OEM company. Together all these companies create a value chain, a certain kind of value creation network that serves the end-users' needs. Understanding of the potential actors and the activities and resources is thus crucial when building partnerships.

4.3 Partnerships

Partnerships concern helping companies to get what they want with the help of other organizations. One important part of forming partnerships is therefore that companies understand what they want from their partners. Once this is understood, they can better evaluate the potential partners and their abilities to offer the help the company needs. However, successful partnerships also require commitment to the relationship in the long run. This kind of committing is required from both parties of the partnership. As a summary it can be said that commitment is an important stage of the partnership development along the stages of assessing, exploring and initiating. [Dent 1999]

Long-term business agreements, single or dual sourcing, integrated research and development, the JIT delivery approach, open sharing of technical and commercial information, joint problem solving and continuous quality improvement have been pointed as some of the main aspects of partnerships [Lehtinen 1996].

There has been a great emergence of partnerships also in the field of buyer-supplier relationships, in which category also the supplier - OEM company relationship can be included. It can be seen as a mutual, ongoing relationship, which includes commitment and information sharing. [Ellram 1990] In the Japanese OEM industry, partnering has been in use for a long period of time.

The Japanese model of the OEM business can be described as a hierarchical pyramid in which there are the OEM company, also called the parent company or principal, first-tier suppliers, second-tier suppliers, third-tier suppliers and possibly also fourth and fifth-tier suppliers. The first-tier suppliers have direct links to the OEM company. The first-tier suppliers are specialized suppliers that often provide extensive system deliveries, which are usually based on the JIT-philosophy, quality control, development activities and coordination functions established for the needs of the OEM company.

The first-tier suppliers choose the subcontractors, in other words the second-tier suppliers, who offer special products or expertise and are typically expected to have the ability to offer such services as product maintenance and repair. At the lowest level of the supplier hierarchy there are suppliers without any special areas of competence. They usually operate as capacity buffers or supply standard goods. [Hyötyläinen 2000]

Due to the hierarchical supply chain system the small amount of direct suppliers of an OEM company is one of the basic characteristics of the Japanese OEM – supplier relationships. The Japanese OEM relationship structure is illustrated in Figure 8.

The trend of substantial reduction of the number of suppliers has also been emerging in the western industries [Gadde and Håkansson 1993]. An OEM company can use three means to reduce the number of its suppliers: it can assign whole components to the first-tier supplier, it can reduce the parts count in its products, or it can single-source parts which had previously two or three suppliers. [Lehtinen 1996]

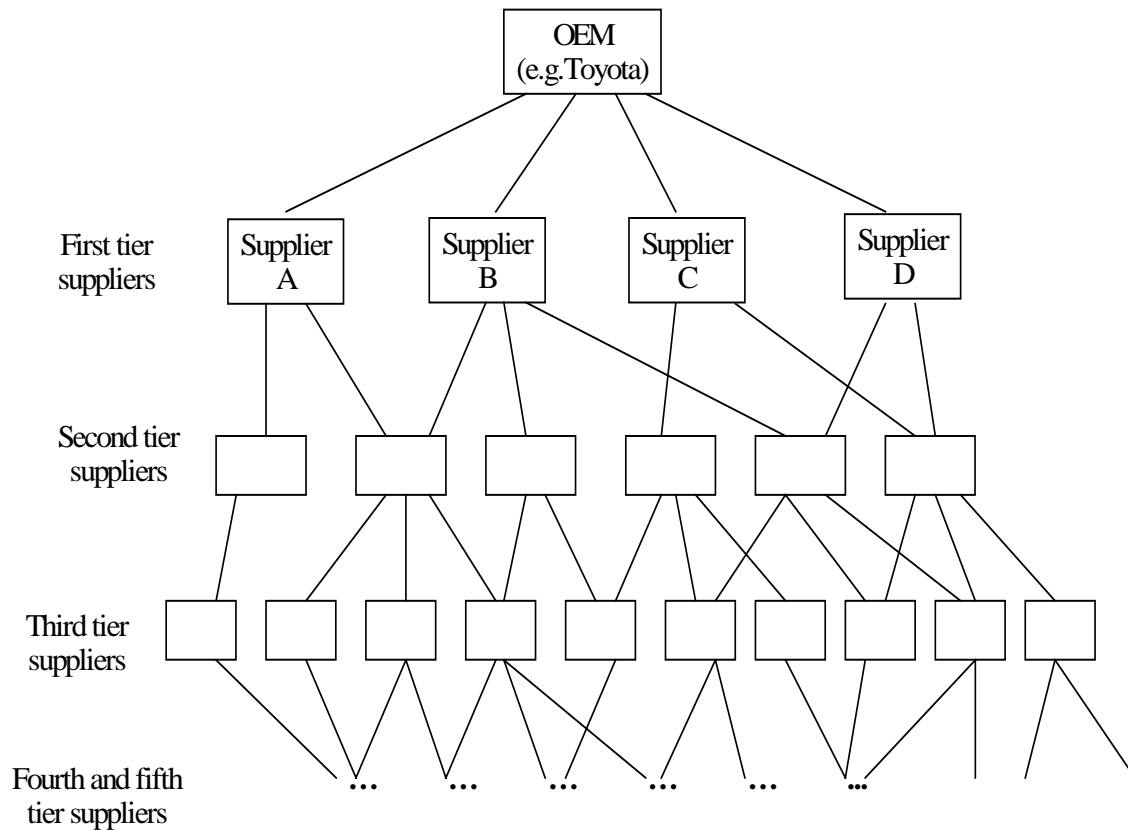


Figure 8. The Japanese hierarchical OEM company – supplier structure.

Besides hierarchicality, also verticality characterizes the Japanese OEM business structure. In Japan, in addition to the OEM company, the OEM suppliers do not usually have other customer companies. In other words this means that the OEM suppliers operate only in one industry. Moreover, horizontal cooperation between suppliers in the same tier is not so common. The Japanese OEM structure is also quite stable, so for example a development of a fourth-tier supplier to an upper level is not that common.

Quite a number of case studies related to the OEM business can be found from the industrial management literature. Common cases that are often presented include, for example, Toyota and General Electric.

The success of Toyota’s production system can be captured in four basic rules: all work shall be strictly specified regarding its content, sequence, timing and outcome; every customer-supplier connection must be direct; the path for every product and service must be simple and direct; and any improvement must be made in accordance with the scientific method, under the guidance of a teacher, at the lowest possible level in the organization [Spear and Bowen 1999].

Another big OEM company, Motorola, needs suppliers, who have the ability to keep pace with attaining perfect quality, are on the leading edge of technology, use JIT manufacturing, and offer cost-competitive service [Gadde and Håkansson 1993]. All of these abilities can be thought to be important also in any other OEM company - supplier relationship. However, Motorola itself as an OEM company invests in the quality of its supplier relationships by providing training and education for the suppliers, while in many cases also the OEM company could learn much from its suppliers. For example, John Deere as an OEM company used one of its supplier's, Hitachi's, know-how in the technology of hydraulic excavators to achieve time and resource savings instead of trying to catch up this kind of know-how by itself [Lorange and Roos 1992].

In the Finnish OEM business the supplier structure can be described as a stable, rather horizontal network, which does not have so much of hierarchical characteristics than the Japanese OEM business. This network of relationships consists of heterogeneous and independent manufacturing companies who make up the value chain for manufacturing the final products. Long-term relationships have always been relatively common among companies. However, contracts for supply have been rather short. Single sourcing has quite often been used to obtain lower tooling and set-up costs.

Relationships end usually because of the changes in the demand of the focal OEM company or the manufacturing strategy of the supplier, not so often because of operational or personal problems. [Lehtinen 1996] It can be argued that the problems related to derived demand rather often impact the relationship. For example, the relationship between the OEM company and its supplier can be ended, because the end-users begin to demand products that do not anymore require the component delivered by some supplier.

The Finnish OEM companies usually have preferred domestic suppliers, because cooperation and problem solving have been easier when both parties speak the same language and also because they are able to visit each other's factories during one day. The favoring of domestic suppliers has an impact on the number of alternative suppliers. Domestic supplier markets are limited. Furthermore, limited supplier markets lead to a situation, where it is costly for the OEM company to change long-term suppliers. [Lehtinen 1996]

5. Marketing channels and intermediaries

5.1 Traditional channels

Marketing channels can be defined as interdependent organizations involved in the process of making an assortment of products and services available for use or consumption [Jensen 1993]. There exist different types of marketing channels from which the companies offering products or services can choose. Usually a distinction between channel types is made by the inherent degree of directness that each channel structure provides. Directness refers to the lack of intermediary stages in the sales or product flow from the point of origin to the consumption point. [Bowersox and Cooper 1992] In other words, the more intermediaries in the marketing channel, the more indirect is the channel.

Direct marketing arrangements have traditionally included vending, telephone solicitation, mail order, television home shopping and door-to-door sales. Some of these channels are relevant only in the consumer markets, despite the fact that business-to-business sales have always been, relatively, more direct than marketing to consumers. The role of multistage distribution has, however, grown along with society that has developed and become more complex. Traditional intermediaries in multistage marketing channels are wholesalers and retailers. [Bowersox and Cooper 1992] However, in industrial markets the channels of distribution are generally quite direct. The selling is mainly done by direct contacts between the supplier and its customers, because the sales usually demand considerable technical knowledge. [Chisnall 1995]

The OEM business is often considered as a marketing channel from the viewpoint of the supplier. Considering the OEM business as a channel refers to a situation where the company's product is sold as an integrated and inseparable component of another company's product. Easy access to specific market segments stands for the benefit gained from this kind of OEM channel. [Kuivalainen et al. 2000] This logic makes the OEM channel attractive particularly for small companies, who are seeking to serve customers with whom they are not so close yet.

However, it is difficult for SMEs to win the attention of big OEM companies. [Moore 1991] Another negative side of this channel alternative is that only the OEM company's brand is visible - the OEM channel is not the best one, if the supplier is dedicated to building its own brand and visibility. [Kuivalainen et al. 2000]

System integrators are seen as a channel that greatly resembles the OEM channel. However, the difference is that when using an integrator as a distribution channel, the brand of the supplier company usually remains visible for the end-user. However, when considering a traditional system integrator, the problem is that it does not often sell the same product twice. It is usually a project-oriented organization that manages large and complex turn-key system deliveries. [Moore 1991] Therefore, although a system integrator would be quite visible as a marketing channel from the supplier's point of view, the problem is that it may not be enduring.

The question of choosing the right distribution channel is important and the channel evaluation should be made properly. The company should pay attention at least to two questions: has the channel already built a relationship with the end-users, and how does the channel fit into the whole product mix and marketing strategy of the focal company? [Moore 1991]

Also, the question of a possible channel conflict should be taken into account. If the OEM business is seen as a distribution channel from the supplier's point of view, there may occur some conflicts between the channel and the other channels used by the same supplier. For example, the product offered via the OEM channel may eat the market share of a product that is sold directly by the supplier. To prevent problems, the supplier should distribute only the core version of its product through the OEM channel and use other channels for selling more full-featured versions of the product. [Rosenberg 2000] However, this policy may not be appropriate from the OEM company's point of view.

5.2 Virtual supply chain

The term supply chain refers to the sequence of processes and activities involved in the complete manufacturing and distribution cycle. As a term it includes everything from product design through materials and component ordering to manufacturing and assembly. It also includes warehousing and distribution until the finished product is in the possession of the end-user. [Franks 2000]

The increasing demand for customized products leads to a situation, where it is not reasonable to build products for the stock. The more reasonable way is to use the build-to-configure (BTC) and build-to-order (BTO) approaches. Many large OEM companies have already realized this. For example, Toyota has announced that it is implementing a system that will allow it to produce a customer-ordered car within five days and Harley-Davidson can build customized motorcycles styled and accessorized according to the purchaser's taste.

However, the BTC and BTO strategies require that the OEM company has a clear and detailed picture of not only each order, but also of the specifically ordered products as they move through the production cycle. Therefore, the company may be much dependent on the information provided by the suppliers to effectively serve its customers.

A good way to ensure this kind of information flow is to adopt and use the JIT philosophy and to make the supply chain more visible. With the help of the Internet, the visibility of the supply chain increases, and the chain thereby becomes a virtual supply chain. [Anon. 2001b]

One important aspect of the virtual supply chain is the integrated flow manufacturing solution that allows companies to eliminate finished goods and work in process inventories, as they synchronize daily production to actual demand. Because the virtual supply chain helps to make the value chain more visible, it offers an opportunity for the interacting companies to create a coordinated community of partners that collaborate in a variety of areas. As a member of this kind of community each company can better concentrate on its core competencies. [Anon. 1999a]

Another basic idea of the virtual supply chain is to provide information concerning, for example, demand projections, production schedules, inventory levels, and quality assurance results for the member organizations of the overall chain. This kind of information sharing can at its best lead to solutions that help to reduce inventories, lower costs of production and also to solutions that support innovative business processes. [Gecowets and Bauer 2000] Indeed, the aim of the virtual supply chain is to move towards a new way to create value [Graham and Hardaker 2000].

The virtual supply chain can, in practice, help the OEM company and its suppliers to share production information in three different ways. The first option is that the OEM company owns the production tracking system and requires its suppliers to input information into that system via the Internet by accessing the OEM company's application via a browser. The second information sharing option for the supplier is to own the production management system and provide appropriate access to that system to each of its own OEM customers. In the third information sharing option, the OEM company and the suppliers both have copies of the unit-level production information, and they synchronize updates to that information. [Anon. 1999b]

5.3 Internet service providers

Internet Service Providers (ISPs) are companies that provide access to Internet for consumers and other companies. Due to this character of a “gate-keeper”, it is common that ISPs are also called Internet Access Providers. The business logic of an ISP is to purchase bandwidth from other companies that have direct links to the Internet. Then the ISP in turn sells that bandwidth to consumers and other companies in smaller chunks. [Anon. 2001a]

Besides providing the access to the Internet ISPs also provide a variety range of different services, like electronic mail and conferencing software, computing support, interactive magazines and online courses [Amuso et al. 1999]. When ISPs are evaluated as an intermediary in the industrial markets, their role as a provider of an electronic market place is central. In particular, the role is essential in the development of virtual supply chains, because ISPs can offer the services needed in building the information system that make up the virtual supply chain.

ISPs can operate either in the consumer markets or in the business markets. It is also possible that they operate in both of these markets. Naturally the business model of an ISP varies depending on which markets it operates in. However, when an ISP is reviewed as a mediator of an OEM supplier-customer relationship, the ISP is operating in the business-to-business context.

An often-used marketing strategy in the ISP business is to provide a free trial period of the services for the customer. This means that for a period of time the new user is not charged for the access service. Besides of the free trial period, ISPs can also offer free software and free technical support. The purpose of these offerings is to familiarize new users with the services the ISP can provide. [Anon. 2001c]

There are several emerging trends, which are going to have an influence on the ISP business, such as the question of Virtual Private Networks (VPN). If companies are going to outsource their VPNs in the future instead of building the VPNs by themselves, they become bound to some ISP companies. On the other hand, if companies decide to build their own VPNs, they remain independent. In other words, they do not commit to use Internet services from some ISP company [Zwicker 2000], but they need to maintain the VPN by themselves. By understanding this trend ISPs can predict how easy it is for business customers to change their ISP strategy. Another important trend for the ISP business is the increase of partnerships. With the help of partnerships ISPs try to answer to the expanding needs of their customers.

Many customers demand that they can get all the services they need from one provider. It is natural that these needs can be answered by strengthening co-operation between supplier companies. [Wilde 2000] Partnerships and mergers will decrease the number of ISP companies. Although the best companies will survive, it is doubtful what kind of a price and quality relationship there will be in ISP partnerships, from the customer point of view, when the number of competitors decreases.

The role of self-service has been emphasized in recent years in the marketing of services. According to Gartner Group, Internet has a big role in developing self-services. It is advantageous for ISPs to support their customers to build self-service packages. Gartner Group also discovers effects on ISPs businesses in the growing need for developing so called comprehensive service environments. Besides the Internet, the comprehensive service environment consists of telephone, fax, and Web and e-mail facilities. As a conclusion: if the service provider wants to fulfill customer needs in transition services, Internet should be considered as a part of this kind of overall service environment. [Zwicker 2000]

The emerging trend of CRM is a major opportunity for ISPs. In short, the idea of CRM is to understand and predict the needs of the potential and current customers. For this kind of understanding and predicting companies need to have databases in which information concerning customers and relationships with them is gathered. Internet can be seen as an effective tool for this kind of information gathering, controlling and utilization. [Zwicker 2000] For example, as a CRM service provider ISPs can help OEM suppliers to take care of their key customer relationships, which usually involve big OEM companies.

In the position of Internet gate keepers ISPs have an important role in preventing the infringements of the Intellectual Property Rights (IPRs). For example, ISPs have a possibility to set warnings concerning inappropriate network traffic or they can deny access for such groups that provide inappropriate information. Also, ISPs are in the position in which they can hire editorial groups to control the network traffic and to enforce standards for the network and its use. Some ISPs have already understood their role with regard to IPRs, whilst especially the small ones have not shown much interest towards it. One possible explanation for this kind of ignorance is that they can achieve competitive advantage compared to larger ISPs by providing access also to such contents that the larger companies do not offer an access to when they try to protect IPRs against any infringements.

In the United States there has been a discussion about whether there should be a law which would force ISPs to take more responsibility in the IPR issues. According to the proposed law, ISPs should remove all the material that causes infringements of IPRs from their systems after they have had a note of this kind of infringement from the owner of the copyrights. The regulation has been demanded, because of the lack of established legal praxis. [Cook 1996] However, instead of fighting against this kind of regulation, the smaller ISPs could turn this question for their own benefit, by consciously building an image which respects the ethics of the Internet.

In the role of Internet gatekeepers many ISPs are going to operate with companies that can be seen as OEM suppliers or OEM buyers. However, the network access provider role is perhaps not the most relevant from the viewpoint of OEM business. The most essential role of ISPs, from the viewpoint of OEM business, is providing such services that support the development of virtual supply chains. As was discussed above, virtual supply chains are helpful in developing OEM buyer-supplier relationships, and ISPs can have much to offer in this development.

On the other hand, as regards product parts and components, the most central role of an ISP is to provide electronic market place services where component suppliers and buyers “meet” each other. However, this way of selling and buying components is most likely not the best one, if the purchased components need tailoring. Thus it can be argued, whether ISPs as electronic market place service providers are much relevant from the OEM business point of view – except in the initial phase of screening potential new component suppliers.

Although the ISP business is continuing its growth, many ISP companies are seeking for new business opportunities. For quite many of them, one possible way to expand their business is to move towards the Application Service Provisioning (ASP) business to be discussed in the next section.

5.4 Application service provisioning

The definition of the concept of ASP is not very clear yet; it has been used in a variety of meanings in different contexts. However, one common interpretation for an ASP is that it is any company, which offers access to application programs on a network basis [Anon. 2001d]. It can also be said that ASPs typically allow businesses to offload their application maintenance needs, including staff and equipment, for a monthly or usage based fee that covers a rental of expensive software applications that, for example, many small businesses could not otherwise afford [Grice 2000].

Therefore, the basic business idea of an ASP model is to rent application programs to other companies using the web as a distribution channel. However, the difference between ISPs and ASPs is not so clear. This is partly due to the fact that ISPs have always been ASPs to the extent that the provision of hosted mail and web services is an application service. Over time, the ISP business has divided between those who provide access and connectivity services, and those who offer application hosting services. The latter ones, particularly as they move into complex web hosting services, are, effectively, ASPs. [Wainwright 2000]

ASPs offer an alternative when companies are deciding if they should outsource applications or maintain – and even build - them in-house. It can be said that acquiring a piece of software for usage from an ASP company becomes an outsourcing situation. The ASP model should allow the customer companies to get their applications installed and running for their own customers faster than by using the more traditional channels. Moreover, the customer organizations should avoid the complexity and cost of establishing an infrastructure on which to base the application and to have an organization to manage it – just by avoiding the staffing costs the customers can save a lot of money. A yet another important business driver is the fact that at the very best an ASP company can provide an "end-to-end solution" for the customer, in other words they can provide a comprehensive, integrated, one-source offering. [Blackwell 2000] As a conclusion, the ASPs create value to the customers by offering a faster and easier way to install and update their software applications [Paul 2000]. For software vendors, ASPs provide a new channel to sell their applications to broader segments of the market on a steady stream revenue basis, compared to the one-time licensing basis. [Anon. 2001d]

In addition to all these benefits, there is also something that ASPs cannot offer. Potential customers of ASPs should be aware of the fact that the ASP model may not offer lower software purchasing costs, at least in all cases. However, the customers may save a lot in the total life-cycle costs, for they need less equipment and own staff. Speaking of the latter, an ASP can not usually offer any customization for the software that it provides; the customers do not have the possibility to change the application. Nor may they be able to use third-party products outside the ASP company's selected group of suppliers, unless they undertake the cost of developing the necessary interfaces by themselves. [Blackwell 2000]

The lack of tailoring possibilities and the need of developing necessary interfaces are issues that may prevent or at least slow down the utilizing of ASPs as an intermediary in the OEM business. This is due to the fact that usually the OEM relationship is characterized by the tailoring of the supplied components and products at least to some extent. In the case of software components the ability to be used by end-users as parts of the overall rental applications of the ASP company, is central.

In more general terms it can be said that ASPs are going to face challenges regarding the number of customers, revenue impacts, licensing complexities, the possibility of channel conflicts and support issues [Anon. 2000a].

Besides the business model, also the question of the technological infrastructure through which the services are provided is important for the success of an ASP company. As an example, a versatile billing system is needed [Anon. 2000b]. It is also clear that technological developments like the ability to use the Internet to move applications from the personal computers onto the “thin” application server environment, and the rise of new types of application servers which can efficiently handle and replace a large number of personal computers, have made the rise of ASPs possible [Anon. 2001d]. Indeed, also the future success of the ASP business in different countries and regions can be predicted by factors like infrastructure and technological culture, but also by market receptiveness and government role, to gauge how open different regions are to the ASP model [Anon. 2000b].

Sound Consulting has listed factors, which are converging to drive the new ASP market forward. These factors are technical, financial and market dynamics. Figure 9 illustrates these dynamics that are fuelling the ASP market. For example, financial factors affect the ASP market because reaching of new customers will become easier and less expensive through the Internet and because the ASP model helps to avoid the costs of application maintenance. On the other hand, the market factors affect the rise of the ASP market mainly because of the difficulty for companies to obtain IT staff. Regarding the technology dynamics, one significant factor is the technologies that allow traditional client/server applications to be accessed in a hosted environment. [Anon. 2000a]

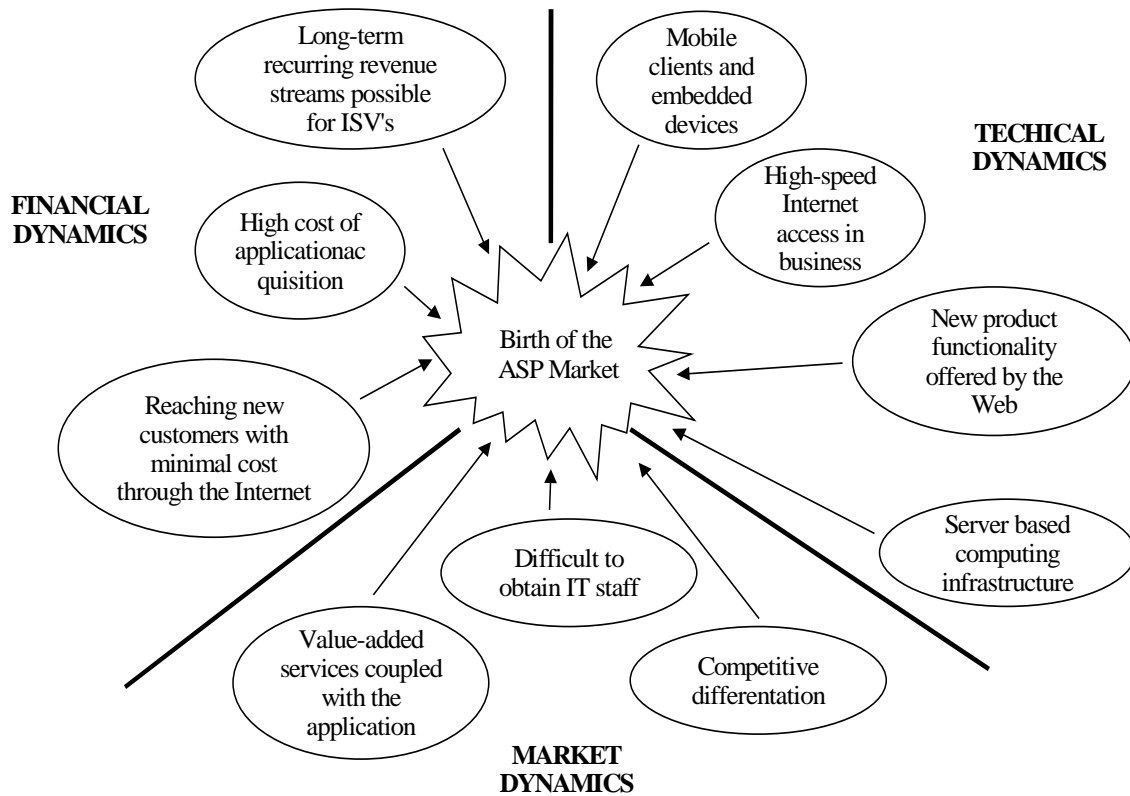


Figure 9. Converging dynamics fueling the ASP market [Anon. 2000a].

However, the success of the ASP business is, in the end, dependent on its customer potential. The most important customers for ASPs will obviously be small and mid-sized companies. Gartner Group has predicted that for them the application hosting will be the dominant software delivery model in the future [Anon. 2000a]. The main reason for this is that for SMEs, application service providers have plenty to offer: an economical, fast and effective way to access a whole set of new applications, to applications that they earlier could not afford. [Anon. 2001d] There are many niches to choose from in serving the SMEs, the most popular being e-commerce and CRM [Torode and Hagendorf-Follet 2000]. Both of these areas are going to be important to develop also for SMEs, but at the same time they are also expensive. The ASP model makes it more cost efficient and straightforward to handle with them in the case of SMEs.

Other possible niche for ASPs in the SME is are, for example, human resource management. It can be argued that the ASP model is not so attractive from the viewpoint of big companies, which usually have the necessary IT resources of their own. As an example, many OEM buyers are quite big companies and it may be possible that they do not get excited about using an ASP company as an intermediary for the simple reason that they have the money to maintain their own systems.

To succeed in the small-business space, ASPs have to be able to offer custom solutions that meet the unique needs of a small business [Torode and Hagendorf-Follet 2000]. To do this, they need to have partners who already are near to the small and mid-sized companies and know their businesses well enough. Some surveys have shown that 75% of SMEs rely on local solution providers as their primary source for IT-related goods and services [Torode and Hagendorf-Follet 2000]. By combining resources with local solution providers and system integrators ASPs might therefore get a better touch of their potential SME customers. With the help of partnering, they have a possibility to also enter markets where they have never been before [Blackwell 2000].

Partnering is also needed because of the fact that many customers demand for a one-contract ASP solution [Grice 2000]. In fact, partnering is so popular in the ASP markets nowadays that most ASPs are actively seeking to chum up with independent software vendors, resellers, telecom carriers, ISPs and hardware vendors. However, the ASPs should carefully consider with whom they are going to partner, because partnering is different depending on what kinds of services are provided. It is also important to understand that the members of a partnership consortium should together build up a functional entirety. For example, when an ASP company partners with a software vendor, there is a danger that the ASP concentrates on building online distribution while the software vendor serves up technical expertise, but nobody specializes in sales. On the other hand, if the ASP partners with a value-added reseller or service-oriented partner, which already has a sales training infrastructure built into its core business mix, there will not be any same kind of problem. [Fusco 2000]

When ASPs choose to sell to customers indirectly, in other words through resellers and agents, they are usually serving the SME market. On the other hand, when they concentrate on large customer companies, it is usually more reasonable to choose the direct sales approach. [Sperling and Gage 2000] Therefore, the need for an ASP to partner with dealers is greater, when the ASP concentrates on doing business with smaller enterprises.

Considering ASPs as possible intermediaries of an OEM relationship, one major problem is the question of whether it is reasonable to rent software components via the Internet. The price of buying rather than renting such components may not be that high, at least in the case of "small" components. In this situation the costs of renting and licensing components may not differ considerably. Furthermore, the question of IPRs can prevent the emergence of ASPs as intermediaries in the OEM software business. One could expect that only if both the OEM company and the component suppliers make use of the ASP model through the same service provider, the question of IPRs becomes irrelevant. In this case, however, the roles of the parties become closer to another type of business web than a supply chain. [Tapscott et al. 2000]

6. Intellectual property rights issues

In the OEM context, some of the most important questions related to IPRs are the questions of licensing, contracting, shared IPRs and royalties. These issues are discussed more thoroughly in the following sections.

6.1 Copyrights, trade secrets and patents

A common requirement of copyright laws is that the work should be original. This means that the work has been created independently and is the personal expression of the author. This factor must be distinguished from the concept of novelty, which usually is not required. In some countries, like in the USA, achieving the protection of copyright laws requires registration, but in some countries, like in Finland, the rights are implemented automatically when the work is done [Haarmann 1994].

The rights protected by copyright laws include exclusive rights and moral rights. The exclusive rights of a copyright holder are the rights to reproduce or copy, adapt, distribute and publicly perform the work. The precise nature of these rights often differs among different countries, but they are usually recognized to some extent in most countries. In some cases the moral rights are recognized as well. For example, the EU Software Directive recognizes these rights. Moral rights include the right to be known as the author of the work (right of paternity), the right to prevent others from distorting the work (right of integrity), the right to control the publication of the work (right of disclosure), and the right to withdraw or modify the work after it has been published (right of withdrawal). This kind of moral rights protection reflects the view that the individual, not only the work, is to be protected.

The basic requirements for patentability are novelty and non-obviousness. Many of the generic and most commercial valuables do not fulfill these requirements. [Chávez et al. 1998] Generally speaking, patent applications should be limited to inventions of significant commercial value.

However, with a help of a proper patent portfolio, the company is allowed to earn revenues from the sales of technology and/or the license of technology rights and to have useful negotiating chips in hand when a third-party patent holder claims infringement [Cerrone and Villeneuve 1996]. Five steps are needed to build a strategic patent portfolio. Firstly, innovations and inventions should be protected by seeking patent protection within the home country and abroad. Secondly, thorough prior-art patent searches need to be conducted to determine where patent coverage is available.

Thirdly, patent searches help to identify and avoid possible infringement issues. Fourthly, monitoring of competitors' patent activities is needed to anticipate areas for future product development, and fifthly, third-party patents can be identified to complement the emerging patent portfolio. [Davis and Weitz 1997] However, the value of a company's patent portfolio can only be maximized by integrating effective IPR protection strategies with normal business practices [Cerrone and Villeneuve 1996]. In other words, the IPR strategy should be built as an integrated part of the overall business strategy. However, because there are different kinds of procedures related to IPR issues in different countries, a company that operates in the global markets should be aware of foreign IPR regulations and patents. This may be a difficult task for example to a small supplier that does not have much resources for pursuing the patent strategy.

If the IPRs of a certain component are shared between the OEM company and its suppliers, proper procedures should be planned regarding which party will act in the case of any IPR infringement. The question of IPR control is therefore important to take into account in OEM relationships.

A useful source that focuses on the European intellectual property law is the Intellectual Property Rights Helpdesk: <http://www.cordis.lu/ipr-helpdesk/>. Useful Finnish sources are, for example, the books of Ritva-Liisa Haarmann (e.g. [1994]) and Timo Kivi-Koskinen [1999], as well as the www-pages of Patentti- and rekisterihallitus (<http://www.prh.fi>).

6.2 Contracting issues

Contracting is an essential part of any business. The role of contracts will be even more significant in the near future, when such trends as globalization [Robinson 1997], outsourcing and alliancing continue to emerge. In the OEM business contracts do play such a central role that sometimes the OEM supplier is called the “contract manufacturer”. The basic problem in many OEM contracts is the question of balancing interests between the relationship parties. Usually the buying OEM companies are much larger than their suppliers and often also possess more legal expertise than their suppliers. This can lead to a situation, where the supplier is more or less forced to abide by the terms the OEM company demands.

One essential feature of contracts is the freedom of contracting. This is a principle that contains the freedom to make a contract, to choose with whom to make the contract, to choose the way to put the contract into effect and to choose which country's law is applied in the case of any disagreements. [Saarnilehto 1996]

On one hand, the freedom of contracting makes the contracting process easier, but on the other hand it also requires much attention from the contracting parties. For example, the terms used in contracts can in a juridical sense and when juridically interpreted, contain issues that the parties have not been aware of. When making global contracts, an important issue is to choose the country the law of which is applied in the case of disagreements. For Finnish companies it is often reasonable to choose Sweden, because its legal system closely resembles ours. Despite the importance of contracts, many small companies seem to ignore to develop their contracting policies. Usually they do not possess enough legal expertise inside their own organization, but still they do not want to use outside services, e.g. attorneys offices or legal consultants. This can partly be due to the fact that many small companies have difficulties in identifying their professional service needs. Also, it is possible that they have, for some reason, negative attitudes towards services. [Lindqvist and Manninen 1998]

Sometimes a contract may not be closed because of disagreements concerning the formal structure of the contract, e.g. the terms, etc. However, it is more common that the parties fail to reach an agreement, because they have different expectations about the future despite their common interests. In this case the parties are so confident in their prediction about the future that they refuse to compromise and no agreement can be reached. This situation could be avoided in some cases by using a straightforward but frequently overlooked type of agreement called the contingent contract. The terms of this type of contract are not finalized until the uncertain event in question – the contingency – takes place. In an increasingly uncertain world, this kind of contracting offers at least the following benefits: it enables a difference of opinions to become the basis of an agreement; it reduces the risk by sharing it among the parties; it motivates the parties to fulfill their promises; and it reduces the impact of asymmetric information.

However, despite all these positive things, the contingent contract requires continuing interaction between the parties, and transparency as well. After all, the final outcome of the contract will not be determined until some time after the initial agreement is signed. Therefore, negotiators need to consider the nature of their future relationship with the other party and the possibilities they both will have to observe and measure the future event they have bet on. [Bazerman and Gillespie 1999]

Problematic areas in OEM contracts may, for example, be the questions of how the payment and maintenance of the supplied parts are handled. One good way to organize the payment question is to use royalties. This approach is good, because it helps the relationship parties to share the risk of any market failure. In more general terms, risk and reward sharing could be seen as a dividing line between a partnership relationship and a contract-based purchaser-supplier relationship.

From the OEM's point of view, risk sharing also encourages the very OEM supplier to do its best. In the case of payments which are not tied to the success of the final product, there may be a danger that the supplier does not care what will happen to the product, because it has already got its own share. On the other hand, one-time payment may be disadvantageous to the OEM supplier side in the case of unexpected market success.

The question of maintenance can be difficult, because the OEM usually wants as long-lasting maintenance and warranty contracts as possible. Especially smaller suppliers may not favor such contracts, because the business logic is different from the development and manufacturing of the original parts. As an example, it is common that only some of the biggest software houses that develop tailored applications to customers offer maintenance services as well, and the latter is often done in separate business units or subsidiaries. In the case of the OEM business the responsibilities regarding product development, manufacturing and maintenance services depend also on the end-users, who may be able to maintain the finished products by themselves or may need services from the OEM company, the parts suppliers or some independent third parties.

One important type of contract in the OEM context is a licensing deal. In a licensing deal the rights to implement the innovation are transferred for value to a third party who will make, use or sell the invention [Chisnall 1995].

7. The customer industry

In this report especially electronics, telecommunications and automation (ETA) companies are considered as potential customers of software component suppliers, i.e. as OEM companies.

The electronics industry is usually discussed together with the electrical industry. In Finland these two industries contributed together as much as 28% of the whole product exports of Finland in 1999. The brut value of the production was 100 billion FIM, investments 4 billion FIM and the R&D expense as much as 9 billion FIM. Although the number of Finnish electronics and electrical companies was only 600 in 1999, they employed as many as 65 000 people. [Anon. 2000c]

The markets of electrical and electronics industry have rapidly grown in Finland: the volume of the industry doubled during the nineties [Lehtinen 1996]. The increase of the production volume of the industry has been estimated to continue. For example, the production volume is expected to grow over 20% during this year compared to the last year. [Anon. 2000d]

As is the case with the Finnish electronics industry, also in the Finnish telecommunications industry the growth has been spectacular. However, much of this recent growth can be attributed to a single company, Nokia, which for example in 1998 accounted for 69% of the turnover of the Finnish telecommunications cluster. Apart from Nokia, the cluster also contains several other important companies: 27 out of the 500 largest Finnish companies belonged to the telecommunications cluster in 1998. [Salo et al. 2000] However, as regards the automation industry, this kind of key figure presenting is not an easy task. This is due to the fact that in official statistics the automation industry is fragmented into being part of many product groups. In fact, the automation industry is a branch of business that includes among other things hydraulics, pneumatics, electronics, and electrical and information technology. [Lindqvist and Manninen 1998]

When comparing ETA industries in their experience of using commercial software components, the automation industry has the widest experience. This can be explained for example by the fact that while the electronics and telecommunications industries prefer subcontractor –based business, the automation industry uses fewer subcontractors. Also the fixed-price deliveries are more common in the automation industry than in the electronics and telecommunications industries. [Niemelä et al. 2000] One possible reason for the difference is the lack of industry-wide standards in automation applications, compared for example to telecommunication, where the growth and vertical integration of companies can base much on standards.

When considering ETA industries as potential software component buyers, an important question is the need for product customization. Such final products as industrial machinery, telecommunication network equipment and industrial automation systems are usually customized, but the components of the products do not necessarily have to be customized; they can also be standard ones, but including a wide range of options and variants. [Lehtinen 1996] Usually the computing infrastructure on which the products are based is built of rather standard hardware and software parts, depending on the need of integration – for example most handheld electronic devices also include customized application-specific components. In telecommunication products also the communication protocol platform may be standardized to a large extent. Such applications as the control of certain types of mechanical machines or industrial processes, are often a mixture of customized and standard solutions. A good example of a standard part of many applications is a database management system needed to store and manage application-related data.

However, in most of the cases some components of the final ETA products have to be customized due to the changing needs of the end-users. Often this is done by following some sort of a product family approach based on software, because it is the most malleable part of the whole product (cf. [Sääksjärvi 1998]). For successful customization, close communication and interaction is needed between the suppliers and their customers. In the case of the software industry, customer involvement may be even more important than in many other industries, because of the malleability of the material of which components are being made [Hoch et al. 1999]. An OCM software component supplier may or may not interact with the end-user of the product, depending on the buyer's purchasing policy and the use of the component in the product.

Moreover, although customization favors direct distribution channels between OCM buyers and their supplier, and even close contacts with the end-users of the product, software component brokerage is a growing trend. The idea of component brokering is that standard or semi-customized components could be marketed to other companies with similar needs [Hoch et al. 1999]. There already exists a few virtual software component marketplaces on the Internet, for example Component Source (<http://www.componentsource.com>). Their purpose is to facilitate component supply and to provide a reliable and branded channel to sell and buy software components, i.e. to facilitate the acquisition and support processes of software components. [Spratt 2000]

Development of information systems for trading software components over the Internet is also going on. There is a growing need of systems that would collect information about software components worldwide over the Internet and provide a set of electronic catalogues of the components. This would help the potential buyers to screen and evaluate components and their suppliers.

Component buyers will need, for example, information about purchase and license conditions, the functionality of the components, component interfacing and such performance issues as the execution time and minimal memory size required by the components, etc. [Aoyama and Yamashita 1998]

At a more general level, brokers can create value both to component producers and buyers through a variety of services. These include, for example, market-based services, requirements-based services and negotiation-based services. In market-based services the intermediary can create value by providing customers with specialized knowledge of the market. Such knowledge includes qualities and quantities of available components and consumers. In requirements-based services the broker can create value by providing clients with feedback on interactions among their requirements and how the market might meet them. Lastly, in the negotiation-based services the intermediary can create value by interacting with customers to create mutually acceptable deals. In some cases the intermediary can, for example, create a packaged deal which matches the number of customers and component producers. [Robinson 1997]

In the OCM context, the relationship between the component supplier and the OCM company is usually so close that the first two services are not necessarily relevant. The third one, “joint deals”, could be important at least in terms of a virtual supply chain which includes a great number of partners. Even the first two types of services can be valuable in the beginning of an OCM partnership, i.e. in the situation where the supplier or the OCM company is searching for potential partners. In fact, there is a strong intention in the electronics industry to increase the use of services that help companies in a component acquiring process e.g. in terms of finding the appropriate component suppliers [Laine et al. 2000].

When considering the possibility of OCM business development in the ETA industries, the specific features and trends of these industries should be taken into account. For example, the electronics industry is characterized by the attractiveness of the business branch, technological entry barriers, unpredictable speed of growth and changes, hierarchy of actors, strong networking, flexibility of production, high level of product tailoring, and clear aim at succeeding in the global markets [Laine et al. 2000].

It can be argued that unpredictable speed of growth and changes, as well as the need of flexibility in the production are trends that favor the use of external software components. In a turbulent environment, the companies need to have several ways to adapt for example their production volume. Using components acquired outside the company enables this kind of adaptation. The aim at succeeding in global markets usually requires partnering and strategic alliances, because only few companies – if any – have enough resources to conquer the world markets by themselves.

8. OCM – analysis of the state-of-the-art

A simple structuring of the information acquired from the Internet on OCM is followed in this chapter. The information is divided into two areas, business related and technical, which will be handled separately. Business-related information is divided further into component selling, buying and brokering topics, and technical information into component development, acquisition, utilization and general topics.

8.1 Business perspective

In general, software business related information is sparse and scattered around the Internet. For example, most American business schools do not seem to have studied the software industry - more business-related publications can be found in software engineering journals and conferences than in scientific business journals (cf. [Messerschmitt and Szyperski 2001]). However, there are a few extremely popular practical software business related magazines, such as the Software Magazine (<http://www.softwaremag.com/>), and plenty of popular textbooks available.

The pieces of information found on the Internet and analyzed in the following are summarized in Table 1. As shown by the table, there already exist two big software component marketplaces, flashline.com and componentsource.com. Other component selling sites found were very small – some of them apparently more a hobby than any serious business. The two big sites illustrate well the kinds of services that will need to be provided when selling software components, especially generic COTS components. The kinds of services associated with COTS selling might, after all, be one of the most important topics to study further from the viewpoint of component developers.

Compared to the selling sites, there is almost nothing on the purchasing of software components – except some general software acquisition frameworks and standards. The buyers' side of the component market would thus deserve more attention, taking into account especially OCM type components that must be integrated into the buyers' products and software production processes. A surprisingly large amount of information – including a few good papers already referred to above – was found on software component brokering. This includes one small company specializing in OEM software licensing consultant work and papers describing ideas of automated software brokering agents. It is likely that brokering will not end up in the ends of the continuum, but somewhere in the middle. ASP type businesses can be seen as a specific issue that is briefly addressed regarding software component brokering; there may be cases where some components are needed only occasionally and their renting – use over the Internet for some specific period of time – becomes a good option.

Table 1. Information related to OCM business.

Source of information	Topic
Component selling	
http://www.wrlldcomp.com/	Seller of embedded COTS user interface software component(s): user interface engine + application
http://www.gsia.cmu.edu/bb26/45-941market/cmarket.html	Software component marketplace (small – hobby?)
http://www.flashline.com/	Software component marketplace (big)
http://www.componentsource.com/	Software (business) component marketplace (big)
http://www.itseeds.com/softwarecomponents/index.htm	Software component marketplace (small)
http://www.votek.fi/ , http://www.votek.fi/te_sw_co2.htm	Finnish OEM software company selling components, products and platforms
[Chavéz et al. 1998]	Software component licensing principles
Component purchasing	
http://www.sei.cmu.edu/publications/documents/98.reports/98tr003/98tr003abstract.html	Description of a computer-assisted method for improving the software acquisition process (e.g., for COTS software components)
http://interactive.sei.cmu.edu/Columns/COTS_Spot/1999/December/COTS_dec99.htm	View to process and product changes required by commercial software products
http://www.esi.es/	GURU software reuse economics method and tool
http://www.roguewave.com/corp/press/articles/sdtimes.cfm	Poll of industrial views to software components and component-based software development
Component brokering	
http://www.everything-linux.com/research/itkc/2000/may/hurwitz_050200_intraware.html	Digital marketplaces, virtual resellers ... monitoring of the usage of software by, for example, component brokers?
http://www.microsoft.com/oem/main.htm	Program for Microsoft OEM system (hardware) developers, meant to help buying and making use of Microsoft products and components.
http://www.stromian.com/guide.htm	OEM software licensing guidelines
http://www.flashline.com/content/Metrix/tm3/tm3.jsp?sid=967783281234-2193382842-151	ASP overview
http://www.softwaremag.com/archive/2000apr/ASPorNot.html	ASP strategies
http://www.adtmag.com/	Application integration
http://www.sdmagazine.com/breakrm/fr_graz.htm	ASP situation in Summer 2000
[Aoyama and Yamashina 1998]	Internet-based software brokering
[Robinson 1997]	Automated software brokering agent

8.1.1 Component selling

The following links are examples of information mainly related to the selling of components, i.e. the viewpoint is that of a component developer, who seeks for and delivers components to external buyers. In summary, not much information concerning the selling of software components was found on the Internet. Two big COTS software component marketplaces have, however, been established in the USA. These aim at making use of the Internet to associate sellers to buyers, including additional services geared especially towards sellers – i.e. component developers.

<http://www.wrlldcomp.com/>

The "World Computers Ltd" is a British company located in London and Brussels. It represents the development of an embedded systems software subcontracting company to a COTS component seller – an idea that is at least indirectly included in the evolution of OEM type software component providers. For the company in case, this change took five years (1995 – 2000). However, there is still basically one component, a user interface engine/application, and the company seems to be almost like a one-man firm.

The Web pages of the company indicate that the company is prepared to tailor and maintain the components, and does not charge its customers based on the volume of usage of a certain component. The main component provided by the company is called the "Front End Module", i.e. a generic configurable piece of user interface software that can function as a stand-alone component or be executed on the top of an operating system. The costs of using the component are claimed to be, at maximum, U.S.\$ 7,500, whereas the development of a corresponding piece of software would cost U.S.\$ 45,000. Savings in time are claimed to be over three months.

The user interface component is described as "a services layer module and as such ... designed to occupy a services slot within the Application. The concept is to move all the common software out of the application layer, to generalize this software, and to move it into the services layer". In other words, the component would separate the presentation layer from the application layer, in a layered software architecture for embedded applications. The user interface component with its associated components can be bought for a single or unlimited number of execution platforms used by the buying company, and also for the software development platform of the purchaser. This distinction is important, as it makes a difference between component-based product developers and exploiters. The components can also be embedded into the customer's own software or hardware solutions, or into some specific operating system. The annual maintenance fee of the component is 760 US dollars. Details of the license agreements for the component can be found on the Web pages of the company.

<http://www.gsia.cmu.edu/bb26/45-941market/cmarket.html>

The site is characterized as "the software components market". Currently it supports the buyers who will be purchasing software components. In the future, it may be expanded to support "the sellers and the market administrators". The list of available components and the current bids can be viewed, i.e. the market functions as a kind of auction site. A bid for components can be submitted. To ensure the validity of the bids, the component buyer needs to provide a team identification and password.

<http://www.flashline.com/>

Flashline is one of the biggest software component market places on the Internet. The basic functions provided by the site are the following: Extend Your Component Development Expertise, Developer Links, See How It Works, Register as a developer, Search through Requests, View your Developer info, and Need a license agreement.

A component design manager is provided for software suppliers. The basic procedure to become a software component supplier is the following:

- "1. Developers register with Components by Design, and create an online profile identifying areas of interest and expertise. Registered developers receive an email notification when a request that matches their qualifications is entered into the system.
2. Anyone can view requests in the system, but only registered developers can ask questions during the Review phase, or participate in bidding. Bids can be edited or withdrawn at any time during the bidding phase.
3. Each developer that bids on a request receives an email notification when a bid is accepted.
4. The developer and requester then communicate directly with one another to bring the project to a close".

For the software component buyers the procedure is as follows:

- "1. Companies complete the "Component Request" form, which includes the component specifications, due dates and other requirements, such as testing and documentation. An email is sent to qualified developers notifying them of a new request.

2. The request then enters a Review period, during which developers have an opportunity to ask questions about the project, and requesters can refine the component specification.
3. After the review period, developers place online bids. The requester can view the bids and developer information. In addition, Requesters can categorize and rank the bids.
4. Once the winning bid is selected, the requester states the reason, and an email notification is sent to all bidders.
5. The requester and developer then communicate directly with one another to bring the project to a close”.

Flashline also provides for software component certification, by ”verifying the availability of critical documentation and by disclosing test results”. The following documents and reviews are submitted:

1. UML diagrams for each component -- provided by the component vendor, posted, and reviewed by Flashline.
2. Java Language Standard compliance and component design review -- Pass 1, executed, evaluated, and returned to component vendor by Flashline.
3. Component efficiency review -- Pass 2, executed and posted by Flashline.
4. EJB scalability -- Pass 3, executed and posted by Flashline on the application server of the vendor's choice.
5. Installation / de-installation instruction accuracy -- reviewed, validated, and posted by Flashline.
6. Application server compatibility -- verified by Flashline on up to two vendor-selected app servers. (more than two application servers can be supported at additional fees).
7. External data dependencies map -- provided by the component vendor; validated and posted by Flashline.
8. API dependencies -- provided by the component vendor; validated and posted by Flashline.

9. Posted customer reviews -- Flashline will provide the repository for these reviews.

The costs of the certification services are as follows: one JavaBeans™ component - \$1,800.00, one Enterprise JavaBean™ component - \$2,400.00, and additional fee for certification on more than two application servers - \$400.00 per additional application server. Flashline also offers component testing services, provided by Flashline's QA Lab. Tests can be executed during all stages of development. Three separate sets of tests are available: Pass 1: Code Structure and Design Analysis; Pass 2: Component Performance and Efficiency; and Pass 3: EJB Load and Performance. Flashline markets testing and certification outsourcing as one of its key services associated with the component market.

<http://www.componentsource.com/>

Componentsource is another software component marketplace, but with a focus on business-related application components. The site especially markets the opportunity for software developers to encapsulate "existing knowledge" into components that can be sold in the marketplace: "Do you have expertise in a specific area or vertical market?, Do you have existing applications with "locked in" functionality?, Do you have plans to build a new product - that will be built using components?, Do you know Visual Basic, Java, Visual C++ or Delphi? If you can answer 'yes' to any of the above questions you can start building business components now". The procedure for component developers to follow would be: "Identify the components that you can build, To get more ideas read our case studies or visit our Component Request Center to see a list of components in demand, Read our White Papers (These will tell you about how to create and price your components), Send your components to us, and We can review the components, test them and give you feedback prior to you offering them for sale".

The present white papers on software components included in the site are the following: Open Market Components a CBDi Forum; Creating Commercial Components (Microsoft® COM and Microsoft® .Net Framework); Creating Commercial Components (EJB); and Creating Commercial Components

(Borland® VCL). The white papers to appear are the following: Creating Commercial Components – Product Naming, Pricing and Licensing; Creating Reusable Components – Separating functionality into self-contained, reusable components; and Creating Components (CORBA).

The components given as examples in the site include the following: (AFD Software) Addressing components, (BEA Systems) eCommerce components, (Business Architects International) Customer relationship components, (Dedupe Technologies) Address deduplication components, (EDS) Retail banking and financial components, (Inabyte) Email validation component, and (Synergy Holdings – AppSoft) Accounts package interface components.

The component marketplace is characterized as a means for "Open Market Component Based Development" that offers buying companies the ability to choose COTS software components for any platform from multiple component authors and combine them with their own in-house components to build applications. Components for the open market "should ideally be built for deployment on one of the following platforms": Microsoft Component Object Model (COM), Sun Enterprise JavaBeans (EJB) and Object Management Group (OMG) Common Object Request Broker Architecture (CORBA).

However, there are other aspects to building open market components that are necessary regardless of the platform, such as utilizing security certificates, providing adequate documentation such as help files, and installing the components. Security certificates allow developers to digitally sign their components, insuring their authenticity and preventing them from being altered.

This service is provided by the company Verisign (<http://www.verisign.com/>). Documentation – especially help files - are "essential to building successful open market components". Other services that are important include Component Object Model (FAQs), Web resources, and white papers.

In order to protect the component provider's intellectual property, some sort of component licensing is needed. This can be used to create evaluation versions so that the potential buyers can "try before they buy". It also allows the component seller to create different versions of the component (e.g. Express, Standard, Professional) with different functionality that can be determined by the serial number or license key entered at install time. The "final step involved in creating components for the open market" is building an installation package so that developers can properly register the component for use with their own development environment.

<http://www.itseeds.com/softwarecomponents/index.htm>

Itseeds is a French company that was founded in 2000. It describes itself as "the creator and operator of a business-to-business community dedicated to leading edge companies from the information technology industries [that provides for] an online marketplace.

Qualified buyers and sellers of IT products and services are able to negotiate deals, i.e. the marketplace allows companies to choose to trade with a variety of commercial partners. Transactions "are private and secure".

The company "generates revenue through a transaction fee to sellers". Compared to the two previous marketplaces, the additional services are not yet well-developed – they are more related to information sharing than actual support of the component deals. The basic functionality of the site includes the following: Search for component vendors, Publish a software component request, Discover software component resources, and Register your company and products (vendors only). The components are classified as follows: Java products, COM/ActiveX products, Visual Basic / Delphi products, and Others. In May 24, 2000 (the latest update!), there were only five registered buyers, but 97 registered vendors.

<http://www.votek.fi/>, http://www.votek.fi/te_sw_co2.htm

Interestingly, one Finnish company – Votek – was found to sell embedded software components. One class of the components is "Base Components/Utilities", including tool kits for managing, running and developing applications based on Votek's components and products.

The component class "Network Resource Components - Signalling Resources" includes ISDN, ISUP and INAP signaling software components. Correspondingly, the "Network Resource Components - Transport Resources" class switching functions for digital access cards. The "Network Resource Components - Voice & Fax Resources" component class is meant to implement DTMF functions and fax sending and receiving function. The "Database Components" includes SQL-based database services to applications based on an ORB interface. The component classes "Network Resource Components - Text-to-speech & Automatic Speech" and "Service Components" – the most extensive class, are also available. Votek also represents a few third-party components, including Orbix middleware, Orbix developer package, Oracle database servers, and Signalling software based on Deck and HP Signalling stacks.

The company characterizes itself as follows: "Votek Oy, with facilities in Pori and Helsinki, Finland, is a company specializing in automated telephone services for customer service organizations, media companies and teleoperators. Votek Oy produces automatic telephone platforms and applications, combining the most modern solutions of computer and tele-technologies like CTI (Computer Telephony Integration). The packaged applications have been developed in close co-operation with customers and they are integrated into the service processes of the customer".

And its product strategy is described as follows: "Votek Oy is a system provider specialized in developing value added services who also designs application software for teleoperators and global system integrators by partnership contracts. The Votek systems are based on a generic TElement system application, open architecture and platform independence, though committing - if needed - to develop partner/end user specific versions of the system- and application software".

This characterization matches very well indeed with the idea of an OEM software developer, which offers components, products and platforms to its customers. The fact that the offerings are based on open standards and technologies, as opposed to proprietary customer-specific solutions, is explained as follows: "Traditional companies offering value added systems have tailor-made solutions containing both HW and SW. These systems are extremely closed and un-standardized plus very expensive. Possibilities for customizing and modifying of the services are very limited as well as the flexibility of the systems. Standardised IN environments will serve different kinds of HW configurations and service volumes. Votek SN/IP with a standardised IN interface will allow several services simultaneously (either generic or segmented services). If services are IN-integrated it will be possible to run the basic functions in IN, which ensures reliability, usability, flexibility and capacity of services."

Chavéz, A., Tornabene, C. & Wiederhold, G. 1998. Software component licensing: A primer.

The paper was the only one found on the Internet to focus on licensing software components, not only software products. The main differences that the authors consider between licensing of software components vs. software products include the following: components are not stand-alone applications, but form considerable portions of some other application; components should be of high quality, ready to be integrated into the product of the licensee; and most licensees assume that the components can be adapted and modified somehow.

From the viewpoint of this report, the last item is most important, because it covers the OCM case, as opposed to the COTS case. According to the authors, this would result in that the licensor of a component makes either the source code or some application programming interfaces available to the licensees: "Component licensors face the disturbing prospect of enabling licensees to integrate components into unknown third-party software, possibly even allowing those licensees to view and modify the component's source code". To keep some control over the modifications, the authors suggest that component developers should limit – by naming them in the contract - the products into which the component can be integrated, as well as the parts of the code that can be modified.

Before making any modifications, the licensee should get a permission from the licensor. The same may hold in more general terms for confidentiality, i.e. access to the component should be limited, especially if the source code or some proprietary application programming interface is available. The authors propose relative small up-front license fees for software components, associated with royalties based on the volume of the actual use of the component. The licensor should "retain all the right, title and interest in the component".

However, things become much more complicated, if the licensee is allowed to modify the component – the interests of the two parties should be shared. The problem of liability, warranty and support regarding the components and their possible modifications should also be addressed. A component licensor should usually consider "offering component licensees a strong but short-term warranty, and making technical support available at standard commercial rates to licensees after the warranty expires". Typical maintenance and support fees would be 12 to 15 percent of the up-from license fee of the component.

8.1.2 Component purchasing

The following pieces of information mainly involve the buyer's viewpoint, i.e. they deal with purchasing of software components.

<http://www.sei.cmu.edu/publications/documents/98.reports/98tr003/98tr003abstract.htm>

The Software Acquisition Improvement Framework (SAIF) of the Software Engineering Institute is based on SEI's reference model (Software Acquisition Capability Maturity ModelSM (SA-CMM®) for buying commercial software products and components.

SAIF is a computer-assisted "tool" or organizer to help in improving the acquisition process. The document in case (CMU/SEI-98-TR-003) "discusses rationale behind the need for the SAIF, the elements constituting the SAIF, and the intended operational usage of the SAIF".

http://interactive.sei.cmu.edu/Columns/COTS_Spot/1999/December/COTS.dec99.htm

This article "The Elusive Search for Categories" by David Carney deals with one of the basic problems in using COTS software components: is this product really COTS?

The answer should indicate in what case a piece of software can be regarded as a commercially available off-the-shelf product on which the buyer's own products can be based. Although the article is somehow naïve, it illustrates what the component buyers should consider regarding the supplier ability as a professional software sales organization, especially with regard to the need of the component buyers to go "from simple installation through more complex parameterization, through tailoring, through customization". The Software Engineering Institute (SEI) defines a COTS product as one that is: "sold, leased, or licensed to the general public offered by a vendor trying to profit from it supported and evolved by the vendor, who retains the intellectual property rights available in multiple, identical copies used without modification of the internals".

Carney concludes that although this definition is better than many others, "the attempt to pin down whether a product fits the perfect definition of COTS is the wrong question: Even if you can determine absolutely that a given package is or isn't COTS, that won't tell you what you really need to know", but some more useful characterizations of commercial software would be needed that "could aid us both in making choices between competing alternatives, as well as in making the more basic build vs. buy decision". This holds very well for OEM type software components, too.

According to Carney, one of the basic aspects of using commercial software is that "commercial software implies process change: we can place the notion of business process--more precisely, the degree to which a business process must change--on a spectrum whose scale is bounded by a custom development at one end and a totally as-is solution on the other".

He calls this the Change-to-Process Spectrum and argues that "rather than worrying about whether some solution is or isn't really a COTS product, it may be of more use to map its potential value (and its potential savings) against the cost that is implied by how much reengineering it will demand".

Although Carney does not speak about OEM software products, the change of the process is an interesting viewpoint from the buyer's viewpoint. Moreover, he points out that the use of more or less customizable pieces of software would also require one to consider the Change-to-Product Spectrum, which is bounded by the degree of change that is possible by the buyer. A more complex end would involve "installation, parameterization, tailoring, customization, ... [and] finally ...gross internal modification".

The product change spectrum is, according to Carney, more ambiguous than the process change spectrum: " up to a point, vendors of some software packages expect and encourage changes: ... there is a certain amount of work that the vendor expects you to do", although it is "equally true that there are certain things the vendor doesn't want you to do". All in all, Carney concludes that "knowing how much work you have to do to use" commercially available software is more valuable than trying to fix the definition of COTS software. From the perspective of OCM buyers, this viewpoint may very well be the most crucial one: what the supplier does and what the buyer needs to do by himself.

This means that "a commercial software decision is massively context dependent, and simply giving some product a label, even if one can, doesn't address the problem. What the decision-maker needs is greater insight into the details of his or her particular context and more information about how candidate products fit in that context". Moreover, process and product changes could be contrasted to each other, "it might be interesting, at some point in the future, to superimpose these two axes into an X-Y graph, and populate it with other actual examples". This kind of product-process change matrices with regard to some software components could be a useful tool for the buyer.

<http://www.esi.es/>

The European Software Institute has dealt with COTS software components in several different projects. From the viewpoint of component buying, the Guide To Reuse Economics (GURU) is interesting. It aims at "helping organizations to estimate the economic benefits of reuse". Moreover, it "provides a framework to correctly quantify RoI by evaluating reuse-associated costs and savings", which "is essential strategic information to decide on a reuse investment and plan reuse adoption to maximise RoI".

Although GURU focuses on in-house reuse, its ideas can be easily accommodated to the reuse of COTS and OEM type external components. For example, GURU helps people in different roles to consider the make-or-reuse (buy) decisions. Software engineers can evaluate whether it is convenient to reuse a component or develop it from scratch.

Component producers can evaluate the benefits gained from developing a reusable component, a project manager can plan which components will be reused and which will be created within a project, and development managers can understand what is the benefit of investing in reuse for a specific domain.

The results of GURU include a reuse economic model framework to help organizations estimate reuse benefits, a tool to support the estimation of the benefits of software reuse, and guidelines for using the reuse economic model framework and the supporting tool.

GURU performs estimations on reuse benefits, gives different alternative scenarios and keeps economic-related data for the reusable components of the organization. NPV, ROI and PI can be presented in graphical form. The tool requires a standard PC platform.

<http://www.roguewave.com/corp/press/articles/sdtimes.cfm>

This is a short article by Rebecca Rohan in a popular journal SD Times on the role of components in software development. However, because it is based on a poll of "a number of industry executives to get a sense of where component development is headed", we will summarize its main findings below.

According to the interviewed executives, one of the main trends is that "components are going to address higher-level business questions. ... because things like EJB and XML give us common ground to build on. Standards, broadly used, provide infrastructure to build higher-level components". Another significant long-term trend would be that of "interconnectivity among myriad types of devices - phones, Pads, non-computer appliances, cars - [because] the increasing demand for software in everything is what's driving the industry toward a components-based developer's model".

Moreover, the "almost frantic adoption of the application server architecture" will create need for components", as "there's a proliferation of non-personal computing devices, and client/server computing doesn't work in that climate because you can't have fat clients. Then there's scalability, because the Internet creates potential for almost infinite demand. One day you could have 10 people on your site, and the next day 100,000. We're seeing an extremely rapid adoption of n-tier computing and application servers- and what runs in application servers are software components".

From the business viewpoint, one of the interviewed executives considers that it is necessary to view the software development "battleground" from radically new perspectives, "as opposed to just fine-tuning what's out there now. It's apparent that the historic application development cycle just won't work - it takes too long".

However, component reusers will "need full development expertise. You won't be able to put together a system using EJB components unless you really understand EJB", although component buying will become very tempting especially for shortening the product's time to the market: "Components are typically priced in such a way that when you look at buy vs. build, you do better by buying".

One of the problems will be how to integrate existing solutions not based on components with new software components. The interviewed executives recommend to "look at wrappery - wrapping current functions. Part of [IBM's] component architecture and tooling strategy will support the ability to wrap current applications to interoperate with EJB components. It's not a new concept, but they should be looking for it".

Regarding extensive reuse of commercial components, "there are various expectations around bugs - a lot of issues around components to consider. There's a huge advantage to choosing components that are pre-built, but you have to shop carefully and know what you're looking for. Be very clear upfront - find out the long-term costs. There are maintenance costs on all software - will you have to do it yourself, or will the vendor provide you with bug fixes? If the EJB standard changes, will the vendor provide you with the latest? Will there be deployment fees, and, if so, when my system scales up, will there be long-term costs associated with that? ... All those things should be addressed in a license or contract with the vendor you're working with".

8.1.3 Component brokering

The following links, as well as the few articles that will also be analyzed, are related to bringing software component providers and purchasers together, i.e. they involve the brokering perspective.

http://www.everything-linux.com/research/itkc/2000/may/hurwitz_050200_intraware.html

The short article "Feeling the Pain and Doing Something About It: When a Digital Marketplace Is More Than a Reseller. News from Hurwitz TrendWatch" by Carol Baroudi from May 2000 discusses the role of digital marketplaces as virtual resellers – and what can be done "beyond product offering and price" to help the buyer. Some insights can be gained regarding the role of component brokers, based on the company called Intraware, which is used as an example in the paper.

This company "identified various pain points not only in purchasing software, but also in installing software, maintaining software, and managing software licenses" for its customers. Tools are provided to seek for the best buy, e.g. to optimize discounts. Other tools monitor the customer's present versions of software to help in taking new versions into use, as well as the use of the software to ensure an optimal level of licenses.

While these services are geared towards helping the buyers of desktop software in large organizations to optimize the cost of office computing, the idea of helping to make the best deals for certain COTS components and assisting in monitoring the actual need of the purchased components on behalf of the buyer, could suit component brokers.

<http://www.microsoft.com/oem/main.htm>

The site provides a very good example of procedures available for OEM hardware vendors, "new complete or bare-bones PCs or distributing hard disks or motherboards". The overall set of procedures is called the Microsoft OEM System Builder program. Information and tools are provided by Microsoft to "successfully build, sell, and support new systems with genuine Microsoft OEM System Builder products", including promotions, technical support, system compatibility tools, software downloads, and online training. The service is delivered through Authorized Microsoft OEM Products Distributors.

What can be ordered by OEM vendors are, for example, Microsoft operating systems, pointing devices, keyboards, selected applications, and system components. Credit, systematic order processing, and fixed delivery times are being offered by the Microsoft distributors.

There are no minimum volume commitments for licenses, the authorized distributors deliver genuine Microsoft products, which include anti-piracy features to help prove that the acquired product is genuine. The System Builder Preinstallation Kit is delivered with operating system products. The kit "makes it easy for you to ship branded systems with software and hardware peripherals fully installed, tested, and ready to use by your customers". In order to become a System Builder Program member for one calendar year, the OEM company needs to buy at least one Microsoft OEM operating system multipack per month from an authorized Microsoft OEM Distributor.

<http://www.stromian.com/guide.htm>

The site is titled to the "OEM Software Licensing Site: Stromian's OEM Software Licensing Guide" by Donald K. Rosenberg of the company Stromian Technologies ®.

The site is highly relevant from the viewpoint of original software component manufacturing. As Rosenberg characterizes, "OEM sales are widely known in the hardware world, but many companies, including software companies, are still unclear about software OEM licensing".

According to Rosenberg, the OEM software business model is similar to that of hardware OEM, "but is driven by the special pressures of the software industry". One of the basic concerns is how to ensure that the customers will adopt software by a licensee to increase functionality that "will add more than enough customers to pay for the license fees".

His guide is based on the company's experience in OEM licensing at Q+E Software and INTERSOLV, in "third-party work licensing software to many vendors of databases, development tools, and applications". For OEM software vendors Stromian Technologies offers market analysis for the product, partner identification and contact, and negotiations.

According to Rosenberg, "the OEM channel delivers higher returns for investment than any other channel. Resellers, even if they pay you to join your partners program, require constant visits and shepherding to keep your product out there. The major distribution channels require payments from you and take a heavy percentage of the sales price, and a roll-out for a full marketing program for an end-user box product is a multi-million dollar undertaking. Even direct sales, in which you get to keep all the revenue, require mailings that typically cost \$50 per order. Finally, all the sales made through resellers, distribution, and direct mail require individual tech support for every end-user".

OEM software sales require, on the other hand, "identification and targeting of the limited number of software applications and vendors most likely to benefit from your software, the making of mutually beneficial arrangements with those firms, and technical support limited to dealing with their software engineers in support of your product. This work is done largely on the telephone and with some visiting. The result is a steady and--we hope--growing revenue stream". This justification of OEM software is valid for components, too. It is worth pointing out that Rosenberg emphasizes the need of technical support, i.e. when an agreement has been reached, the main contact with the customer is its technical experts.

According to Rosenberg, one of the main goals of an OEM software developer is "to become a standard, so that your product goes from being the odd choice to becoming the obvious choice". Rosenberg argues for OEM customers' ability to "hammer" the software so that its robustness and quality will increase considerably, compared to "ordinary" buyers. Moreover, OEM customers help the software producer to introduce new, generic product features by first developing it for these customers.

Rosenberg claims that "the best deal is to get the customer list -- the people on it are known to be buyers of a certain type of software". In other words, "if your OEM licensee is large and won't give you all the customer's names, then offer to trade your list against his list, name for name". In other words, the OEM company and its buyer may end up in fighting for gaining the same end-users as "full customers". OEM software sales is, according to Rosenberg, somewhat slower in terms of sales cycles than end-user software sales: "Six to eight months is typical, and a large company may take over a year" to become a buyer. Therefore, one of the most important time-determining factors for an OEM software company is the customer's product's release cycle, since it determines the time window for make-or-buy decisions and for evaluating alternative purchases. The OEM software guide provided by Stromian Technologies includes the items shown in Frame 1.

I. Why Should You License?
High Benefits/Cost Ratio
Increase Number of Channels
Prestige and Publicity
Product Improvement
Industrial Quality QA
New Features
Increase Market Presence
Increase and Identify Your Future Market
II. Who Should Try it?
III. What Should You Offer?
IV. When is OEM Licensing Appropriate?
V. Where?
VI. How is it Done?
Determine the Product
Determine the Market
Don't Forget Your Competitors
Early Approaches are Market Orientation
Approach the Prospects
Evaluation
Negotiation/Pricing
The Contract
What if They Want to Buy Me?
Don't Forget Support
What About Source Code?
OEM Licensing Consulting

Frame 1. Table of contents of the Stromian OEM software licensing guide.

An OEM software company should "be offering a core version of your product, with limited functionality to prevent channel conflict with your full-featured product. Ideally, this core functionality will work as a feature inside the licensee's product, and the end-user will not be able to access all the functions and features of your full product as sold in the box".

According to Rosenberg, "OEM licensing is appropriate at any stage of the product life cycle". For example, in the case of some well-established dominant technology, there may be some system-level functions that would be useful in many products. On the other hand, if some technology is rapidly becoming hot, it is likely that many product developers would want to use it in their products. Technology in its early stages is therefore a candidate for OEM licensing: "it is possible to interest a larger company in a product and have them help fund the new product and company through large advances and license fees". However, Rosenberg points out the fact that depending on the overlap of OEM products, the OEM company may very well try to license technology to people who might think of themselves as their competitors.

Regarding the pricing of OEM software, "a complicated, modular product will likely require a correspondingly complicated royalty arrangement". The ideal may be a flat price or a percentage, based on boxes sold or users, but on the other hand, there may be several other ways of counting and calculating, including seats, servers, sites, and one-time or annual fees. As the licensee's business changes over time, it is necessary to make changes in the OEM agreement, too. Rosenberg points out that while the OEM contract is confidential, "the relationship is an important point to publicize, so put something in the contract about joint publicity. ... [and] marketing efforts (such as mailing lists, coupons, etc.), and make sure your rights in your software and trademarks are fully protected (the licensee should display your copyright and patent information in every place that his information appears)".

One of the most important things in maintaining the OEM relationship is to "support the engineering efforts of your licensee as he integrates your product. You will need to specify clearly the standards of support and time limits, as well as the cost for support beyond those limits". It is likely that the "source code is bound to come up, particularly in your earlier transactions". According to Rosenberg, "the best way generally to handle requests for source code is to agree to an escrow arrangement, and then charge the licensee suitable initiation and annual fees for the service". The escrow arrangement will ensure the buyer that he will not risk his own business by the use of the present OEM software deal. It is important that the conditions specify that royalties will still be paid even if the licensee obtains the source code from escrow, and "that once the conditions that triggered the delivery of source code from escrow have been rectified, -- the code is to be returned to escrow".

<http://www.flashline.com/content/Metrix/tm3/tm3.jsp?sid=967783281234-2193382842-151>

The article "11 Steps to CBD. The BeanQueen" by Paul Harmon is included here as an example of writings of the rapidly evolving ASP markets. From the viewpoint of this survey, ASPs can be regarded as one form of software component brokering, where the component is rented for the use of the buyer, rather than licensed. This kind of a setting would, of course, make the ASP company become a part of the operating system offered to the users. In practice, the component "buyer" company would ensure that certain components will be available at run time for the larger system that it is offering to the customers. Mobile applications are a good example of such a situation, and indicate that the ASP software component scenario is not merely a theoretical possibility.

According to Harmon, "J2EE has become a standard that has been adopted by most application server vendors. As a result, the factors that once differentiated the various application servers are fading away. In a way, J2EE has raised the stakes. Competition is no longer fierce at a low technical level, but at a high solution level". Harmon considers application servers to consist of core and extended services.

The former include, for example, Runtime and execution processes, DBMS access and connectivity, Dynamic Web pages, and Context/session management. The latter include Load balancing, Failover, Management tools (administration and deployment), Database connection pooling, and Caching techniques. Application servers would also "need development tools in order to boost productivity".

In addition to J2EE, such scripting techniques as Microsoft's Application Server Page, CFML, PHP, and Perl are available although Microsoft provides, according to Harmon, "little more than a scaled-down version of the Microsoft platform". Moreover, "If you don't have a problem adopting Windows NT for middle-tier development, then the Microsoft platform (COM, DNA, ADS and so on) is a logical choice. CFML, Allaire ColdFusion's markup language has also become a part of its parent company's Java strategy".

There are also a few Open Source scripting languages like PHP or PERL. For example, PHP has according to Harmon "been extremely successful and appears to be a real contender". Scripting tools are often "considered entry-level application servers because they lack built-in load-balancing and failover features; nonetheless, they often get the job done". J2EE-based solutions aim at enterprise-level applications, because of J2EE's distributed and object-oriented capabilities.

Because application servers are becoming standard at their core level, the notion of application server is becoming "much like the operating system". Harmon claims that "J2EE plays an important role in this standardization process", but Open Source Software, which is strong in the Java application server field is another force fuelling the evolution. "A handful of important Java OSS projects such as Enhydra, Apache JServ & Tomcat, and Jonas make it easy for every company to have a J2EE application server".

Harmon points out that "one big issue still needs to be addressed: application server productivity. If we compare the productivity of today's Web application development tools with the rich client/server tools we had (and still have), there is still a lot of room for improvement".

<http://www.softwaremag.com/archive/2000apr/ASPorNot.html>

David Linthicum addresses the question "To ASP or Not to ASP?", mostly from the viewpoint of enterprise applications, although "new ASPs ... are looking to deliver all types of applications using the ASP model. These companies offer core enterprise applications, such as ERP (e.g., SAP, PeopleSoft, Baan, Vantive, etc.), down to the run-of-the-mill office applications such as scheduling systems and even word processors". Linthicum points out that "in addition to ASP start-ups, traditional network companies and ISPs are looking to get into the mix. For instance, AT&T just launched a new campaign looking to attack the emerging ASP market. The integrators are in the game as well. Andersen Consulting and Exodus Communications are teaming to provide ASP and e-commerce services. Moreover, Qwest and KPMG have formed an ASP joint venture to address business-to-business e-commerce. Finally, Microsoft is working on Microsoft Office Online".

According to Linthicum, Microsoft is hoping to be the first vendor to extend the ASP model from enterprise applications to office automation, although yahoo.com and excite.com "already provide some rudimentary hosted office automation applications, including e-mail and shared scheduling".

The architecture of ASPs is "nothing more than a hosted Web-enabled application, like ones that are running on many sites today. ASPs typically run Web-enabled versions of applications from application vendors". They also employ some security software to validate users, and some virtual private network software, to ensure that others do not see sensitive data on the Internet. Linthicum views ASPs through a four-tier Web-enabled solution consisting of a data layer, application layer, Web server layer, and user interface layer or browser.

To host many applications and serve a large number of users, ASPs maintain data centers: "While some ASPs provide only one centrally located data centre, many ASPs are hosting applications at many locations around the world, so that the application users can connect to the application servers over a short distance" 24 hours 7 days a week. Reliable infrastructure is therefore important when selecting an ASP. However, "there is really nothing technically innovative about the ASP ... what is innovative is the business model they employ, and the volume of application information they deliver to users".

According to Linthicum especially start-up companies "are finding that the ASP model makes sense for them. They're able to get up and running quickly, and they don't need to hassle with purchasing enterprise servers, nor perform application installation or customization". Issues that existing businesses have to deal with when considering the ASP model include integration of ASP-hosted applications and current enterprise systems, security and privacy, and perceived versus real costs of the ASP model. ASP integration issues involve three dimensions: ASP-to-enterprise integration, ASP-to-ASP integration, and intra-ASP integration.

Linthicum considers that "most organizations that use ASPs typically host only one or two of their enterprise applications with the ASPs. This is known as ASP-to-enterprise integration and is a huge challenge to the new ASP model". To solve this problem, "ASPs are going to have to provide middleware pipes from their hosted applications over the Internet to existing enterprise systems". ASP-to-ASP integration refers to the fact that many companies may have more than one ASP under contract, and may have to share information with all applications. "Most ASPs do not provide a solution today to share information between their hosted applications and applications hosted by another ASP". Finally, the notion of intra-ASP integration "is not as challenging, and is something that ASPs are currently addressing. Simply put, it's the integration of applications, or application domains, hosted within the ASP data center".

Besides integration, security and privacy are issues that may be important to many organizations. Moreover, cost is clearly the "reason many existing companies may migrate to the ASP model". In part for this reason, some ASPs will be focusing on offering "a very low-cost solution, furnishing simple applications where little, if any, integration is required (e.g., payroll)". This will provide the early ASP opportunity for most companies. Others are aiming at the high-value enterprise application market where integration is a key requirement. There are, however, only two basic pricing models: rental and per-use. The former is based on accessing an application over the Internet on a monthly or annual basis. Usually there are limits of the number of users who can access the application at any given time, capacity of database storage, and the amount of changes that can be made to the hosted application.

The pay-per-use model could be more attractive for many smaller companies. When using this model the ASP customers pay only for the access time. For instance, a charge-per-minute rate may be used.

Another issue to consider when thinking about ASP costs is configuration and customization. "Most organizations are not willing to take the application as delivered out-of-the-box, and most spend a good deal of time and money configuring and customizing the application to meet their exact requirements". This is an expensive process and may not entirely change when considering a hosted application. "Typically, the more customization required, the more the ASP is going to cost, and thus the less value the ASP model has within your organization".

<http://www.adtmag.com/>

One of the big issues facing OEM software developers, be their offerings available as licensed pieces of software or as ASP services, is application integration. To solve this issue, some kind of middleware is required – either generic, commercial and well-packaged solutions or specific and custom-made connectivity programs developed for some particular application integration need. Jack Vaughan's article "Application Integration. The state of application integration" from August 1998 discusses some of the basic questions related to application integration and middleware.

According to Vaughan, the driving forces behind application integration include, for example, "linking back-end data stores to the World Wide Web and connecting generalized 'off-the-shelf' enterprise resource planning (ERP) packages to other, perhaps more specialized, application packages. ... There is hope in the industry that packaged applications may come to be connected via application integration software that is, in turn, mostly a packaged solution". Vaughan emphasizes that application integration is "essentially a build-buy issue". A technical problem to address is that in many cases application servers are "very narrow in the functionality they support," so that "we end up with application server farms that are like a Noah's Ark, one of this and one of that. One from Netscape, one from Active [Software], one from Oracle." This can cause software developers to use middleware technology from several separate vendors to connect business applications, i.e. to require the developers to write custom code to link the middleware and application software together.

From the end users' point of view the question is, if the application integration package vendor becomes a service provider, rather than only a middleware supplier. That will happen, according to Vaughan, since for the end users "the appeal, it would seem, is in not having to write your own connectivity programs ... [because] it could take two years to scope out [a heterogeneous distributed system]".

http://www.sdmagazine.com/breakrm/fr_graz.htm

Chris Preimesberger and Alexandra Weber Morales discuss the situation of ASPs in July 2000, in their article "The Application Service Providers Have Landed". Why would an organization favor the "rent-an-app" mode? According to the authors the reason is that "management hopes to save money, time and talent in moving to a ready-made, fully supported e-business or resource planning suite", freeing the in-house software development team to concentrate on coding more important pieces of software. However, they feel that "despite the hyped-up market potential, ASPs today are primarily aiming for the low-hanging fruit of dot-coms—those with big ideas, deep pockets and few or no developers—and providing conventional e-commerce or business software for vertical markets". Yet, "the software rental model isn't new; in fact, it goes back to the mid-1960s, when it was called mainframe time-sharing. When PCs and then networks came to people's desktops in the early 1980s, the pay-as-you-go concept went fallow. But now it's come full circle."

What is important is that "ASPs serve as the new delivery vehicle for software companies. ... Within a couple of years, we will see the majority of client/server software available only through the ASP delivery model." The ASP field began to grow about three years ago. In 1999, the total market was worth \$296 million, will possibly surpass \$1 billion in 2000 and reach close to \$10 billion by 2004. At this point, according to Preimesberger and Weber Morales, "just how many ASPs are out there is unclear, however. Information Week Online listed 86 providers of application-hosting services (mostly start-ups) doing business in the U.S. at the end of 1999, using either existing packages such as SAP R/3 and PeopleSoft or offering their own custom-developed software. In May, WebHarbor.com, a Web portal offering a free listing to companies describing themselves as ASPs, listed 564 firms".

However, "of the hundreds claiming the moniker, four merit attention as true market leaders [of ASPs]: they are USinternetworking, Breakaway Solutions, Oracle and Corio. Ultimately, the situation from the end users point of view can come to what is envisioned by Scott McNealy, CEO of Sun Microsystems: "Five years from now, if you're a CIO with a head for business, you won't be buying computers anymore. You won't buy software either. You'll rent all your resources from a service provider". Typically "customers sign three-year contracts with a 10 percent down and monthly fee. About 80 percent of the applications that for example USinternetworking provides its customers are packaged products. According to the authors "Oracle is in a strong position to dominate the emerging ASP industry because it has more system vendor, service provider and independent service vendor partners than any other information management company". It plans to expand its own ASP program by partnering with telecommunications companies.

Aoyama, M. & Yamashita, T. 1998. Software commerce broker over the internet.

Aoyama and Yamashita describe an approach to "web-based commerce of software over the Internet", a system that would collect information about software components available world wide, and make use of digital catalogues of the components so that they would be described using a specific language. This language would make explicit both technical and business related aspects of the components. The authors claim that they have already (in 1998) collected information about more than 16 000 components. However, no newer publications by the authors could be found to describe any implementation or experiences from the use of the system. Most likely, the system is not in any remarkable use, if even finished from the research prototype stage.

Robinson, W.N. 1997. Electronic brokering for assisted contracting of software applets.

The kinds of services that a brokering organization can offer, according to Robinson, include: market-based services (possessing detailed knowledge of the markets, including cumulative information on the development of the markets); requirements-based services (assisting software sellers to match their offerings with the needs of certain markets and customers); and negotiation-based services (matching of sellers and buyers, creating packaged deals as opposed to isolated single transactions – possibly becoming a kind of system integrator helping to put together larger offerings, and helping to organize associated legal, financial etc. services).

Robinson's idea of electronically assisted contracting is in his own opinion useful especially if the market is volatile and the contract to be negotiated is complex (for example, simple contracts in stable markets can be managed by electronic data interchange systems). This is close to the COTS case especially if the components that the buyer is seeking for are generic technology-driven solutions, as opposed to application-oriented components.

8.2 Software production viewpoint

Table 2 shows the sources of information analyzed on the development of OEM type software components. Referring to the beginning of this chapter, the sources of information in the table are divided into software component development, acquisition (from the technical point of view), and utilization. There are also some general pieces of information related to the development of software components, such as bibliographies, which will be handled separately in the following.

Table 2. Information related to OEM software development.

Source of information	Topic
<i>Component development</i>	
http://www.forsoft.de/teilprojekte/a4/ccotssw/index-e.html	COTS requirements specification
<i>Component acquisition</i>	
http://www.sei.cmu.edu/publications/featured/technical.html#99tn015	Lessons learned from COTS acquisition
http://www.sei.cmu.edu/publications/featured/technical.html#99tn003	-"-
http://www.sei.cmu.edu/publications/featured/technical.html#99tn010	Custom-made vs. COTS software architectures
http://www.sei.cmu.edu/cbs/papers/eval_bib.html	Bibliography on COTS evaluation
http://interactive.sei.cmu.edu/Columns/COTS_Spot/1998/December/COTS.dec98.htm	COTS evaluation experiences
<i>Component utilization</i>	
http://www.sei.cmu.edu/cbs/cbs_description.html	A COTS initiative
http://www.asset.com/Information/home.html	The asset project
http://www.bredemeyer.com/links.htm	Software architecture links
http://www.infosys.tuwien.ac.at/Research/projects.html	Active components project proposal
<i>General</i>	
http://wwwsel.iit.nrc.ca/projects/cots/bibliography.html	COTS bibliography
http://cs.ua.edu/components/research/index.htm	Software component research links
http://www.componentsoftware.org/home.html , http://www.cbhdq.com/links.asp , http://cetuss-links.org/	Software component related link collections (interest groups)
http://www.cbdedge.com/resources/index.html	-"- (private)
http://wwwsel.iit.nrc.ca/projects/cots/bibliography.html	COTS bibliography
http://cs.ua.edu/components/research/index.htm	Component-related Web site

8.2.1 Component development

Software component development is here considered mainly as the problem of component sellers, although it should be emphasized that usually the buyers must follow a component-based approach, too.

<http://www.forsoft.de/teilprojekte/a4/ccotssw/index-e.html>

The link describes a plan to address requirements engineering of complex COTS in the area of automation software. A motivation for the research topic is that COTS software is "developed for a whole market instead of individual customers. This influences strongly the requirements engineering process beginning at the elicitation until to the specification of the requirements and also the whole development process of this kind of software".

The following issues would need to be considered regarding COTS requirements engineering:

- the analysis of different sources of requirements,
- the prioritization of requirements,
- the development of complex COTS in versions and variants, and
- the utilization of the bulky information for the requirements engineering process of complex COTS.

The contact person regarding the research is Bernhard Deifel, at Technische Universität München, Institut für Informatik (deifel@in.tum.de). However, it remains unclear from the material included in the pages, if the project has ever been carried out. No publications are mentioned, either.

8.2.2 Component acquisition

<http://www.sei.cmu.edu/publications/featured/technical.html#99tn015>

The report CMU/SEI-99-TN-015 "Lessons Learned Applying Commercial Off-the-Shelf Products Manufacturing Resource Planning II Program" by Lisa Brownsword and Patrick Place "seeks to identify important acquisition, business, and engineering issues surrounding the use of COTS-based systems and thus derive available solutions, where possible".

The authors claim that although easy system construction from pre-existing COTS-based building blocks that "snap into place" is appealing, "current reality reveals a less than ideal picture, particularly for commercial off-the-shelf (COTS) software components". They examine the similarities and differences of organizations that have applied COTS and the successes and failures of those organizations, to identify a number of significant capabilities that an organization must have to succeed with a COTS-based approach to software development.

<http://www.sei.cmu.edu/publications/featured/technical.html#99tn003>

Scott Hissam and Daniel Plakosh address risks involved in COTS-based software development in their report "COTS in the Real World: A Case Study in Risk Discovery and Repair" (CMU/SEI-99-TN-003). The pressure to adopt COTS components can come into conflict with security constraints. The major elements of this conflict may be the buying organization's overall approach to system security on one hand and the economic forces that drive the software component industry on the other hand. In the report the authors describe an actual product evaluation where such a conflict occurred, examine why that conflict exists, and outline the corrective steps that were taken.

<http://www.sei.cmu.edu/publications/featured/technical.html#99tn010>

The report CMU/SEI-99-TN-006 "Custom vs. Off-the-Shelf Architecture" by Robert C. Seacord, Kurt Wallnau, John Robert, Santiago Comella-Dorda and Scott A. Hissam addresses the Generic Enterprise Ensemble (GEE) developed by the members of the COTS initiative of SEI. GEE is a generic approach to building distributed, transaction-based, secure enterprise information systems, i.e. it can be used as a tool to help in the selection of technologies and architectural choices when building enterprise information systems. A comparison is made in the report between GEE-based solutions and off-the-shelf solutions based on the Enterprise Java Beans specification.

http://www.sei.cmu.edu/cbs/papers/eval_bib.html

The link includes an annotated bibliography of COTS software evaluation approaches, produced in 1998-2000. According to the report, "the topic 'evaluation' itself is vast, ranging from rigorous statistical analysis to the sociology of group consensus". The bibliography covers papers that have explicitly been devoted to the evaluation of COTS software as well as a selection of papers that describe the important techniques used for COTS evaluation.

When evaluating COTS software, it is important to define what to look for. According to the authors of the bibliography, two kinds of attributes are generally considered, functional and "non-functional." The former ones describe the operation of the software, i.e. what it does. The authors "found little literature on this subject; apparently it is presumed that you know what you want, though this is often questionable in practice". On the other hand, many papers list sets of non-functional attributes, which are also sometimes called the "-ilities." A typical example is the ISO 9126 set of functionality, reliability, usability, efficiency, maintainability, and portability.

Once the desired attributes are defined, the various candidate products can be examined to see to what extent they exhibit these or other useful attributes. The authors "found few papers explicitly on this topic; most papers seem to assume that one just looks at the manuals". Typically, examination of the candidate products yields a checklist or spreadsheet describing each, with no one product clearly superior on all criteria. Before these values can be numerically combined, they must be transformed into scales for which numerical calculation is appropriate.

According to the authors, the bulk of the heavily statistical literature on decision support "involves methods to aggregate a number of disparate attributes to choose the 'best' among the candidate products". This is often called the multi-attribute utility theory.

Some papers are concerned with the process: what sequence of steps will lead to the "best" choice, or at least a defensible choice. Some of these are concerned with processes for incorporating COTS in larger systems, others address the sub-task of evaluating a collection of COTS products to select one. Most of the process-related papers also list a few categories of "non-functional" attributes which should be considered.

Many of the individual papers included in the bibliography are annotated, which provides for a good overview of the methods proposed for software evaluation from the COTS point of view. The papers will, however, not be analyzed in any further details in the following, because the information is best available from the above mentioned link.

David Carney addresses in this article the problem of "COTS Evaluation in the Real World", as he describes it. He defines COTS evaluation as "a process to decide whether to select one product for use in a given context". Moreover, evaluation is not restricted simply to a single moment of selection, but includes for example the tasks of defining the selection criteria, perform vendor surveys or market research, and so forth. Carney suggests evaluation to consist of three main tasks: plan the evaluation; design the evaluation instrument – as he calls it; and apply the evaluation instrument. The tasks include the following activities according to Carney:

Plan the evaluation:

- Define the problem.
- Define the outcomes of the evaluation.
- Assess the decision risk.
- Identify the decision maker.
- Identify resources.
- Identify the stakeholders.
- Identify the alternatives.
- Assess the nature of the evaluation context.

Design the evaluation instrument:

- Specify the evaluation criteria.
- Build a priority structure.
- Define the assessment approach.
- Select an aggregation technique.
- Select assessment techniques.

Apply the evaluation instrument:

- Obtain products.
- Build a measurement infrastructure.
- Perform assessment.
- Aggregate data.
- Form recommendations.

Moreover, certain roles of people involved in the evaluation process must be defined. The "three essential roles" in the COTS evaluation process are those of a decision maker, analyst and stakeholder. One of the central ideas of Carney's is the importance of context, i.e. evaluation always takes place within some factors and constraints (for example, functional, technical, platform related, business related) that exist before a COTS product is chosen. Context "includes everything with which and against which the COTS product must harmonise, conform, and operate; it is the basis on which we develop evaluation criteria to assess the product". Moreover, instead of merely hard "must haves" by which to judge a product, the evaluators should have a more flexible collection of features, some mandatory, some desirable, some nice to have.

This collection of features provides, according to Carney, the context, the source of the criteria by which to decide whether a product is sufficient for the needs. Furthermore, "since this evaluation context will in some ways replace a complete set of requirements, it is very possible that novel or unexpected features of COTS products will actually change our idea of the system as it is being built". This would be "a radical situation for people who consider that an architecture ... must be established at the beginning and that COTS products must be found to conform to it". Depending on the business strategy and situation – the context - of the buying organizations, the ways that they perform COTS evaluation may be quite different. For example, one organization's evaluation plans can be rather opportunistic and another one's follow a strict method. Evaluation criteria can be business-related or weighted towards technical issues.

8.2.3 Component utilization

http://www.sei.cmu.edu/cbs/cbs_description.html

SEI's COTS-Based Systems Initiative is certainly one of the biggest efforts to try to boost the component-based approach not only to software development, but to maintenance as well: "The shift from custom development to COTS-based systems is occurring in both new development and maintenance activities. If done properly, this shift can help establish a sustainable modernization practice".

<http://www.asset.com/Information/home.html>

The Asset project was originally constituted by the Defense Advanced Research Projects Agency, as a sub-task under the Software Technology for Adaptable Reliable Systems program to provide an on-line repository for reusable software. In 1995, Science Applications International Corporation began transitioning Asset to a commercial site on the World Wide Web. The company's core capabilities and experience, therefore, are "centered around software reuse and Web technology: digital libraries, database management, object-oriented systems development, software configuration management, distributed information systems and Internet/Web-based telecommunications". The company offers products and services in digital library support, electronic commerce, World Wide Web solutions, and software engineering with an emphasis on reengineering and reuse.

<http://www.bredemeyer.com/links.htm>

Although Carney warns in his COTS evaluation article about the fallacy of fixing the product architecture or design too early, so that novel COTS solutions cannot necessarily be utilized, there must be some architecture into which to integrate COTS software components. In the case of OCM type situation the role of architecture is essential, because the component supplier is prepared to make modifications for its offerings based on the customer's special needs.

The web site of Bredemeyer Consulting company includes "Top Software Architecture Sites", including a definition of architectures, architecture design resources and communities, and seminars, workshops and literature about software architectures. Links to methods, languages and tools used in software architecture design are also included, such as the Unified Modeling Language, as well as standards organizations.

<http://www.infosys.tuwien.ac.at/Research/projects.html>

The site describes a project proposal for EU's fifth framework program, called "EasyComp". The idea is that such web entities as XML data and pieces of software would be active components ("shippable places") that "will replace documents, desktops, browsers, 3-tier architectures and complete applications: everything will be an active component". Both end-users and professional software engineers would need to compose active components." The composition technology should enable to buy components in web-based supermarkets and compose them with powerful composition operators". The project's goal would be to develop the foundation of such a composition technology and to demonstrate it in a web-based composition machine. However, there is no further information available, the proposal has most likely not been funded.

8.2.4 General component related information

<http://www.componentsoftware.org/home.html>, <http://www.cbdhq.com/links.asp>,
<http://cetus-links.org/>

All these are link collections to software component related sites. For example, Cetus claims to include structured access to "18,875 Links on Objects & Components". It is "non-commercial, free of charge, free of advertising, quickly accessible, mirrored all over the world, independent of any organization, developed and maintained by private people and protected by copyright". Another way to put this would be that Cetus is a software component expert community.

From the viewpoint of component development and utilization, these kinds of communities are useful as structured information sources. They may also provide an alternative to such commercial Web-based software component warehouses.

The Cetus software component links are divided into the following: General Information, Distributed Objects & Components, Internet & Intranets, Architecture & Design, Languages & Development Environments, Databases & Repositories, and Related Topics. Each main area is further divided into different sections, and each section into paragraphs. The standard paragraphs are: Start here, Examples, Central Sites, Related Sites, Link Collections, Tutorials, FAQs, Newsgroups, Mailing Lists, General Articles, Special Articles, Books, Magazines, Organizations, Conferences etc. This structure can be compared to the structure of the commercial software component sites. In a way, it can reflect the needs of component users, whereas the structure of the commercial sites should reflect the needs of component acquirers.

<http://www.cbdedge.com/resources/index.html>

This is a yet another public non-commercial, but private, component information site by Alan Brown: "Components will power the next generation of software solutions for the Internet age. Yet many people are uncertain where to turn to get the latest information on components technology, tools, and development approaches. Let CBDEdge be your resource guide! CBDEdge.com is a personal collection of ideas and information brought to you by Alan Brown. It is an independent set of resources, and is not affiliated with any organization or company".

<http://wwwsel.iit.nrc.ca/projects/cots/bibliography.html>

This site is a rather limited bibliography of COTS related literature by the National Research Council, Canada. It has been updated only in June 1999.

<http://cs.ua.edu/components/research/index.htm>

This page is maintained by the Software Component Engineering Research Group of the Department of Computer Science, University of Alabama.

9. Conclusions – state-of-the-practice

Initially, the acquired information was intended to be structured based on a view to component-based software development and businesses borrowed in part from [Niemelä et al. 2000]. There are numerous ways to view software businesses (cf. [Hoch et al. 1999, Moore 1991, Nukari and Forsell 1998, Autere et al. 1999]) and even more concerning software production. For the former the models shown in Figures 10 and 11 were intended to be used, and for the latter correspondingly, the component-based software production processes shown in Figure 3 that are based on [Niemelä et al. 2000] and reprinted from [Niemelä and Seppänen 2000]. These models do not include much regarding the businesses and software processes of component brokers, which was thought to be the main aspect to be revised in the models, on the basis of the gathered information.

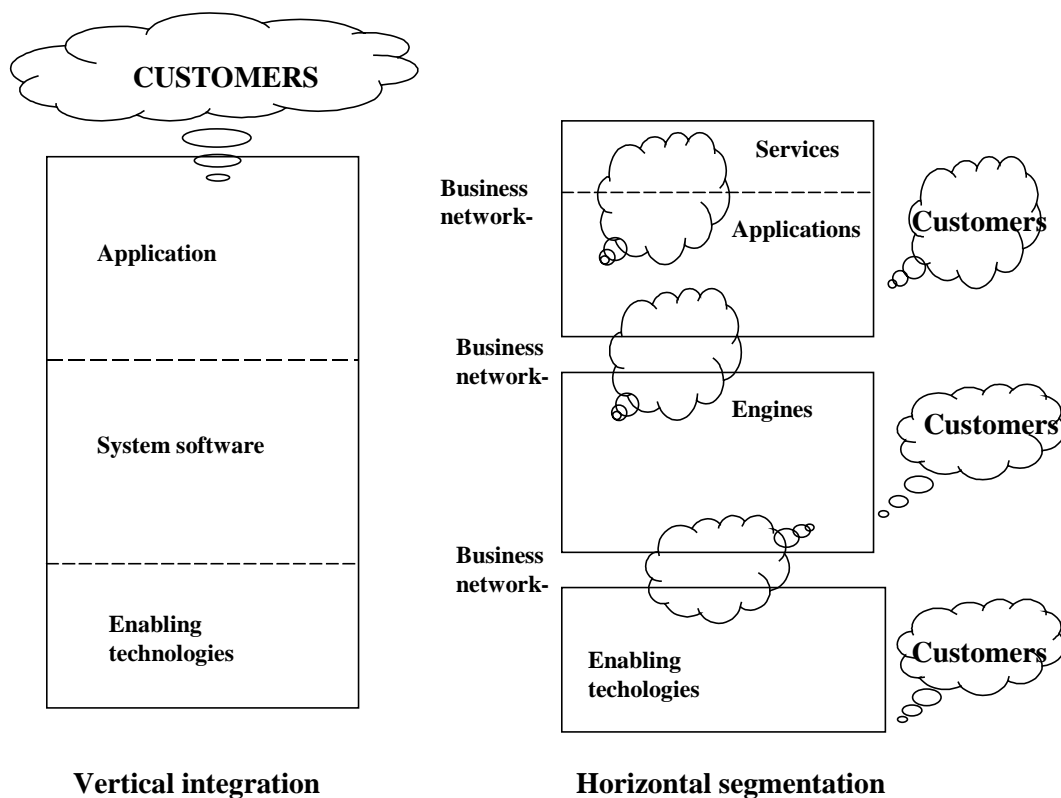


Figure 10. Horizontal segmentation: view to software markets.

Figures 10 and 11 indicate that the initial concepts intended to be used in the structuring and analysis of the information from the software business point of view were as follows:

- type of deliverable (software production services; the entire, or part of the, products of the customer or the supplier's own branded products),
- the number of expected or present customers of the deliverables,
- six different types of businesses that follows from the above, depending on the closeness of customer relationships and tailorability of the deliverables,
- horizontal product segments vs. vertically integrated products,
- business networks: relationships to vertically integrate some horizontal segments, and customer relationships, to deliver the products.

This initial analysis model appeared to be far too fine-grained in the light of the acquired information. On one hand, there is apparently much more information available in publications not accessible by searching the Web that could have helped to make use of the intended analysis framework. On the other hand, the information that was found indicates what was pointed out in [Niemelä et al. 2000] about the immaturity of the software component markets and businesses. This view became even more obvious in the interviews that were conducted. In other words, components have become an interesting approach to organize software development in-house, but business wise open component markets are still immature.

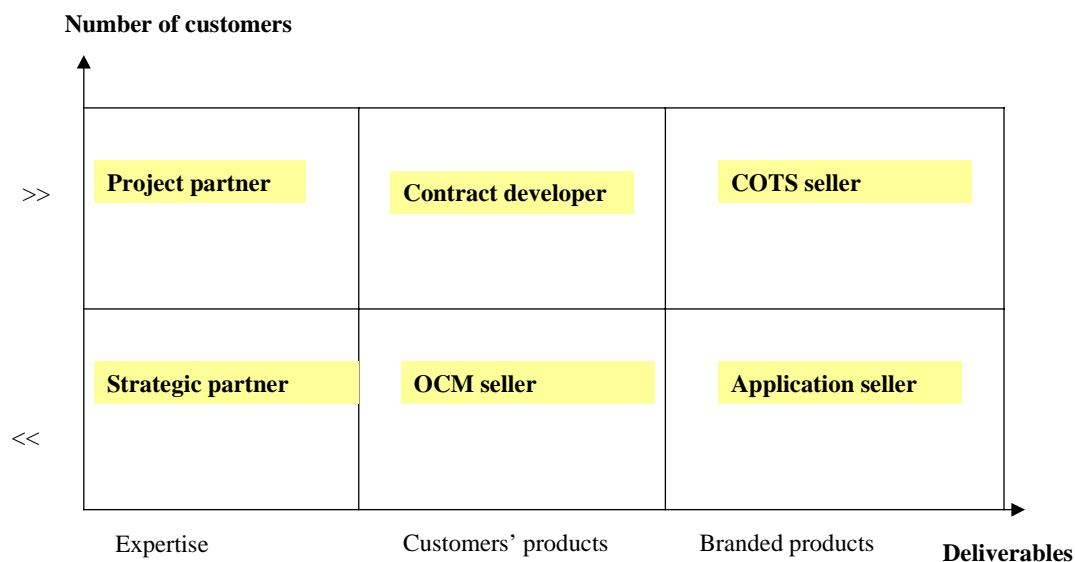


Figure 11. Types of software businesses.

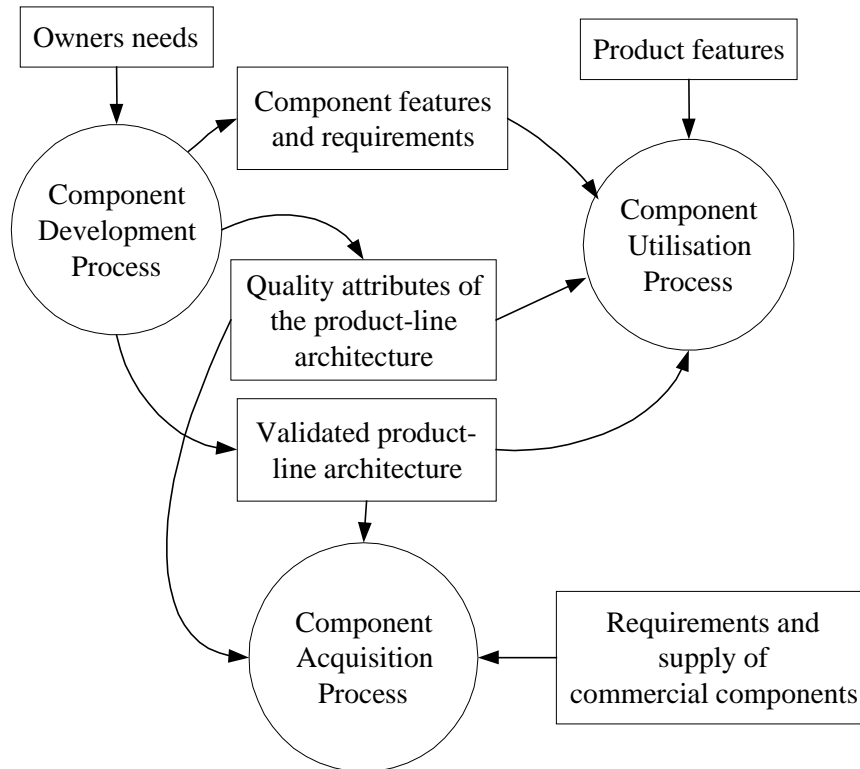


Figure 12. Process context for component-based software production.

Figure 12 shows that three key process areas were considered to be involved in component-based software production: the component development process, the component acquisition process, and the component utilization process.

Although only one stakeholder (“owner” of components in the development process) is shown in the figure in relation to these processes, there are more of them. The roles of a component developer, supplier and user can be seen from the material acquired in this survey. Another key concepts in component-based software production shown in the figure are the following: component development needs, component supply requirements, component features and requirements (or, specifications), requirements (“quality attributes”) of the software architecture in which the components will be incorporated, features (“validated”) of the implemented software architecture, and features of the delivered products.

Figure 12 has been drawn from the viewpoint of a component buyer, who is an active component developer itself. In other words, it assumes an ETA company developing and integrating software components in the context of a product-line type of setting. This viewpoint can be contrasted with Figure 11 that illustrates the suppliers’ perspective to software components.

Also the initial process model appeared to be too specific in the view of the information that was gathered. The information was much too generic, but on the other hand no detailed analyses of information included in certain comprehensive software component sites were carried out. What the results indicate is that the model shown in Figure 12 is not apparent in the information included in the web, i.e. no such key process areas could be made visible.

The attempt to structure the information in advance was shown above to illustrate that what was considered relevant by researchers to characterize the problem in hand cannot yet be seen in the information put into the Web. In the following, the analysis will therefore be carried out by starting from the simple structuring of the information in Tables 1 and 2: component selling, component purchasing, component brokering (the business view); and component development, component acquisition, component utilization and general component-related matters (the technical view).

9.1.1 Business drivers for original software component manufacturing

One of the main conclusions that can be drawn from Table 1 and to some extent from Table 2 is that the idea of software components has thus far resulted only in a very few considerable COTS business endeavors, i.e. software component markets do not yet exist even in the USA. The two big component marketplaces, Flashline and ComponentSource, are after very small considering the total volume of software businesses. In Finland only one small company was found to actively market software components – although there may, of course, be others as well.

The services that have been built in this phase focus on quality assurance, testing and component certification – the most important technical services related to components. The marketplaces are not organized according to any application domains or the buyers' industries, but based on the technological foundations of the components to be sold (Java, COM, Visual Basic, CORBA). In addition to the two big sites, a few small component selling companies were found, but these cannot be considered as any serious businesses. No companies that would openly sell OCM software components could be found.

Table 2 indicates that there exist, however, rather many interest group types, non-commercial sites to leverage information related to components, and even actual components. General interest in software components is thus quite high. Although this interest is not geared towards business, it may affect the markets (cf. Linux), if there is enough convergence in some component domain managed by some community.

The interest groups that were found in this survey do not indicate such convergence, though. The same holds for the commercial COTS marketplaces. The idea that is most visible is more like a COTS software component supermarket than any specialty store. The basic difference between the firms selling COTS or OCM type software components, in addition to their size, is in the business idea: World Computer and Votek types of companies, small OCM suppliers, have apparently relatively few customers for which the components are being marketed, delivered and installed using the traditional software product sales and customer service channels.

This is possible even for COTS components because of the small size of the firms – the seller’s costs to be covered are limited, after all. It is interesting that both World Computer and Votek have clearly chosen the combined OCM and COTS mode, i.e. they try to cover two rather different types of businesses. The most likely reason for this is, again, the small size of the companies. One could still say that this is the idea of producing highly generic (horizontal) components for very narrow domains – cf. diversified business-to-business product sellers. The user interface is one of the most obvious domains, but at least World Computer does not seem to have any big business based on the idea of diversified COTS software components provided for this domain.

The article of Chavéz et al. [1998] on software component licensing is well tuned with the COTS orientation of the information found on software component markets. It is likely, however, that the ownership of OCM type components regarding the basic component solution and its customer-specific features, as well as the price, are difficult questions in practice. As the article indicates indirectly, it may not be worthwhile to address this question, if the component is “small”.

If the component is a fully generic COTS solution, the licensing and pricing problem is much the same as with any software product used as part of another product – i.e. the proposition that “most licensees assume that the components can be adapted and modified somehow” does not hold.

The few items found on component purchasing are actually more engineering oriented than business related, and some of them take a rather mechanistic view to the purchasing process. It seems that this area of software business is most immature yet. However, some good pieces of information were found on software component brokering, dealing especially with OCM components.

The component purchasing problem, in the view of Table 1, includes the main questions of choosing what components to purchase, how to find candidate suppliers, how to evaluate and select components, and how to change the software process for the needs of the externally acquired commercial pieces of software.

What is not shown, but is still most likely of utmost importance (cf. Figure 12), is how to design the buyer's product architecture – not only the process – to support incorporation of software components. Moreover, it might be worthwhile to buy some parts of the architecture, too, not only components – which is actually not indicated in Figure 12. Moreover, if the buyer's product is a large system, it may include several sub-domains with different software development and purchasing strategies. This became very clear in an interview of a system integrator type of a company carried out in the project. One conclusion from the few pieces of information on component purchasing in Table 1 is thus that as OCM type buyers' products become "big", there is no uniform view in software platform and component development and purchasing anymore – different sub-systems of the same product may be based on quite different strategies and solutions.

The information that was found indicates that software acquisition reference models can be used, for example, by COTS purchasers to evaluate and improve their component buying processes. In practice, these models may still be rarely used, because the volume of purchasing must be at a certain level to make it cost-effective to establish and keep improving the purchasing process. The cases where systematic software acquisition processes need to be established may be either large one-time investments or continuous purchasing of commercial software for the company' products (for example, just to mention a figure - over 30 percent of the purchased software).

Regarding OCM, the Microsoft services to OEM companies (i.e. firms that use authentic Microsoft products and components in their own products) illustrate the kinds of arrangements that would be needed between original software manufacturers that develop software products based on branded COTS component or software or hardware products of other companies. Especially the idea of an OEM program provided only through authorized distributors (or, authorized component brokers) would be relevant. This kind of program can be established at least when selling platforms – or interrelated parts of platforms, not only individual components, to OEM buyers.

Maybe this could be one of the basic differences between COTS and OCM suppliers – in the former the services are those of a marketplace (searching, trying, choosing and buying of components), in the latter the choices are much more limited, and the services focus on supplying prefabricated subassemblies to the buyers (first-tier supplier).

Rosenberg, an OEM software consultant, implies that OEM software business would be of the type of product feature business, so that the OEM company can sell full products of its own based on the OEM and other features. This is important from the viewpoint of the expected growth of the software sales, and was pointed out in one interview as the basic fallacy of component purchasing.

The OEM company's customers may not be interested in "tiny" optional features of large systems, and the salesmen are not eager to sell them. The "feature component" providers' earnings may thus become very modest. This can also be understood from the fact that a company may aim not at creating new product features for the market, but whole products, businesses, business categories or industries. On this scale, OEM software selling – especially if it means providing optional features for the buyer's product - would be only a start for something much bigger.

Aoyama and Yamashita [1998] view the software industry to become divided into component vendors and component integrators, and the software commerce broker industry to serve as a middleman. This would be comparable to parts manufacturers, system integrators and intermediate wholesale type of distributors in other industries. Although the paper is apparently somewhat optimistic especially with regard to the mechanizing of the global software component market – including the kind of language that they propose to describe the components - it is an interesting scenario because of this view to the change of the software industry. Flashline and ComponentSource are clearly examples of this kind of change, and represent the emerging responsibilities of COTS software component brokers.

Compared to the interview of a system integrator (or OCM) type of a large company, the need for, and role of, a software commerce broker would, however, depend on the granularity of the purchased components or platforms. The bigger the purchased items are, the less obvious would the role of the broker become, because the relationship would be reduced to the "ordinary" software product selling and buying problem. In this kind of situation, an ASP type broker could, however, have a role not during the product development, but in the operational use of the OCM. Robinson [1997] is yet another mechanized approach to software component brokering, this time focusing on the contracting process of applets. Robinson argues for the need of intermediators in software production and product selling value chains, but seeks to automate the operations of such intermediators. In particular, he proposes that software products – or components – could benefit from negotiation agents that seek for best deals.

According to Robinson, optimization of software transaction costs would make room for the brokering business, in the first place: brokers must be able to improve the deals, to make it cost-effective for the buyers and sellers to use their services. This is most likely a valid remark: the brokering businesses must indeed bring along some added value, compared to direct relationships between sellers and buyers.

Referring to the discussion on Aoyama and Yamashita [1998] above, in the case of OCM type software components this question is even more important: the seller (component developer) and the buyer (component-based software developer) are likely to have rather close relationships, and both parties need to be rather technically oriented. In COTS component developer – component exploiter type cases the relationship is likely to be more distant, and the exploiter type of buyer may not be technically as competent, with regard to software development, as the component seller.

9.1.2 Original software component engineering

Rather many references were found on COTS acquisition and evaluation, which is clearly one of the topics on which researchers have focused. This stems, most likely, from the fact that there are no open OCM markets – for example, customer-specific components are being developed as part of subcontracting type relationships – and the buyers, who wish to make use of components without any close software development contracts with suppliers, need to consider COTS solutions. However, as was pointed out earlier in this report, also the COTS markets are immature (“primitive” may be the right word to describe their current status), and the information that was discussed above deals mostly with research topics and results. Therefore, the actual present and near-future needs for component acquisition and evaluation methods is not yet known.

Speaking of component evaluation, the overall conclusion from the papers included in the bibliography of evaluation approaches is that evaluation has been considered as a major problem by several researchers, but there are still no superior or dominant approaches to solve it. The reason is most likely that evaluation is a context-sensitive task (cf. Carney’s article discussed above), and that there is also an unknown balance between mechanized evaluation techniques and expert judgement.

The more mature the buyer’s own software process and products are, the more likely is his use of rigorous component acquisition and evaluation methods. However, at least one of the interviewed buyers, a well-established company, did not consider evaluation of “small” components as any problem at all, compared to the problem of finding partners that would provide “big” components, sub-products and platforms for long-term use. In our opinion this viewpoint is opposed to many of the papers on COTS evaluation, which emphasize the needs of an occasional buyer to select a piece of software from an open market place, without any close relationships with the potential suppliers.

The same kind of flavor is also incorporated in the references to general software component related information shown in Table 2. Component-based software development is, on one hand, considered as a strategic choice for the buyer organization – including the design of e.g. product architecture that provides for the utilization of components. The components available from suppliers are, on the other hand, seen much as generic pieces that must be sought, evaluated and selected without many possibilities to affect the suppliers' solutions. This view became apparent also in the interview of a large system integrator type buyer, who is following the architecture-driven component-based software development approach.

10. Problems and opportunities

The information presented at the beginning of this report concerning the characteristics of the OEM business is mostly based on studies related to more traditional industries like the automotive industry. This is a consequence of the fact that there are hardly any studies made that would deal with the OEM business specifically in the software development context. Due to this, one of the basic questions to be made at the end of this report is how well does the OEM related information concerning more traditional industries apply to the software industry.

One major difference between the software industry and more traditional industries is that the software industry is much younger. The industry is not as ready for structuring in OEM style channels as the more traditional industries are. For example, the development of western automotive businesses from competitive supplier relationships to more stable and closer OEM buyer-supplier relationships has taken several decades. It can be argued whether the software industry is yet ready for the OEM type business, in the very essence of the term. The emerging component-based software engineering approaches will certainly pave the way to this direction, but in the end the question of flourishing OCM buyer - supplier relationships is business-related, not merely technical.

Another important question is if the software industry will ever be ready for close OCM type relationships, due to the strong role of competence in the buying and selling industries and the malleability of software. When there are continuous and rapid changes in the industry, predictions of the future markets are hard to do. This can lead to a situation where the buying companies are not ready to give away any parts of their business, because they do not know which part of their business is going to be successful in the future. They may decide to hire more software engineers themselves rather than invest in software supplier relationships.

This kind of fear of losing future possibilities can prevent the development of close supplier-buyer relationships: it has been argued that in technologically turbulent industries, which the ETA and software industries are, neither the suppliers nor the buyers want to get deeply engaged with any specific partner. However, the high turbulence often also means resource scarcity in the times of heavy demand for end products, and in this kind of case, it might be worthwhile to take the risk of trying to develop more cooperative relationships [Möller and Wilson 1989].

Also, the questions related to product architectures differ between more traditional industries and the software-intensive industries: the architecture of physic products is simpler and less abstract than the architecture of software-based products [Sääksjärvi 1998].

It can be argued that the complexity of product architectures both in the ETA and software industries can delay the full utilizing of commercial software components. However, in spite of this kind of complexity, the entire software industry will move progressively to utilize component architectures over the next five years, according to many predictions (cf. e.g. [Sprott 2000]). The ETA industry has already largely adopted the product family based approach during the nineties, and it will include software architectures and components as part of that approach during the coming few years.

Despite all these differences between the software industry and more traditional industries, there are some basic features of the OEM type business which are relevant also in the software business. Firstly, the central issue in the OEM type business is the concept of component, whether it is a physical product part or a piece of software. Secondly, the demand for components that become parts of an end-product, is always derived. This causes major impacts in the business, as presented earlier in this report. Thirdly, the relationship between a supplier and an OEM company can be analyzed, for example, by the closeness of the relationship parties, dominance versus balance between the parties, and by the relationship phase, despite the industrial context.

10.1 Issues related to derived demand

Demand for industrial products is always derived from the ultimate consumer demand, as was discussed earlier in this report. According to this definition, the derived demand also concerns those industrial products that are bought by another company to be used in their production process. However, the impacts of derived demand are stronger in such industrial products, i.e. in components that are bought to be assembled as parts to final products. This is seen, for example, in the question of maintenance, which becomes more complex when the product is sold to a third party.

Usually the supplier has no direct relationship with the end-user, and the supplier and the OCM company have to make an agreement in which the responsibilities of warranties and maintenance are defined. An interesting question in the OCM context is whether a third party could take some responsibility for the maintaining and repairing of software components. For example, in the automotive industry the evolution of the component business has led to a situation, where the third parties can do the repairing of the product and can replace non-functional components with new ones.

Derived demand has also other impacts on the OCM business. Firstly, the industry where the OCM company operates can have major impacts on the buying behavior and on the product development processes of the company.

For example, ETA industries have different kinds of policies, structures and dynamics, which will influence the relationship between the supplier and the OCM company. Secondly, the question whether the OCM company operates in domestic markets or in global markets has its own impacts, too. Even if the OCM company itself would be domestic, it can have major markets that are global. Because of the derived demand, global end-users and their cultural differences may affect the supplier and its software component development processes.

The third influence of derived demand is the question of whether the OCM company itself operates in the business-to-business markets or in the consumer markets; in other words, whether it serves other companies or individual consumers. If the OCM company operates in the business-to-business markets, the company also finds itself in the situation of derived demand. This can cause many effects to the component supplier.

Closely related to the question of derived demand is the difference between a system integrator and a pure OCM company as a distribution channel for the supplier. If the software component developed by the supplier is delivered to the end-user through a system integrator, the component is usually more visible. On the other hand, if the component is a part of a pure OCM product, the producer of the component is most often not known by the end-users. Some suppliers may prefer the pure OCM style, because they do not have the capabilities or even willingness to build strong brands of their own. However, other may aim at serving the end-users directly some day or want to develop their software components into a direction of COTS sold to other companies, too. In both of these cases, it can be argued that the brand visibility provided right from the start could be beneficial for the supplier.

10.2 Relationship issues

Relationships play an important role in the OCM business. Usually the relationship between supplier and the OCM company is rather close, due to the need of component tailoring and shared product development process. The closeness of the relationship can bring many advantages that include, for example, reduction of overlapped functions and new product innovations. However, the close relationship also requires a lot from the relationship parties; building a successful relationship needs many investments in the relationship and most of all, it requires commitment. This kind of commitment can be a risk for both parties because by committing to a certain supplier or customer, cooperation with other potential partners may be blocked, as a result. Especially in such a competence-driven and dynamic industry as the software industry is, this kind of blocking is essential to take into account when considering commitment to certain actors of the network.

The question of commitment is even more important to the less powerful party of the relationship than it is to the dominant parties. Both the supplier and the OCM company can be the dominant party, however, depending on in which markets they operate. For example, if the OCM company is a domestic company buying from a global supplier, it can be possible that the OCM supplier is larger and more dominant. However, usually the OCM companies are larger than their suppliers are. This dominance balancing question was discussed earlier in this report.

Due to the close relationships, the question of the possibility of using intermediaries is quite complex in the OCM situation. Usually the relationship does not allow any third parties, because these parties can prevent the development of mutuality in the relationship. However, because the close relationships do require a lot from both parties and usually concern very long-term decisions, the choosing of a right partner is an essential task. Many small software companies may not have enough resources e.g. to find suitable partners, to choose among them and to get them to believe how profitable a relationship would be for the buyer. A third party could be useful in these kinds of problems. Thus, a need for relationship intermediaries may occur especially in the development and early phases of the OCM relationship.

To succeed, the OCM relationships also require open information sharing. This may be a difficult question just between the two relationships parties, the OCM and one supplier. It is not a surprise that the question of information sharing becomes even more complex if there is an intermediary in the relationship.

10.3 Component-related issues

There are different kinds of software components needed in producing and assembling products of the ETA industries. The components can be classified by different criteria. One relevant criterion is the importance of the component in the whole product. It can be argued that the more important part of the whole product the component is, the more important, equally, is the supplier for the OCM company. This importance of the specific component can increase the dominance of the supplier compared to the OCM company.

The size of the component is another criterion for categorizing components, but usually it is quite closely connected to the importance of the component. Both of these questions can have a major impact especially on what kind of license agreements the supplier and the OCM company should make.

In addition to these two criteria, also the price of the software component has an impact on the relationship parties. If the price of the component is “low”, the OCM company should buy it in such large volumes that it is profitable for the supplier to tailor it for the buyer. If there is no need for any tailoring of the component and the supplier can sell the same component to many other customers as well, the question of buying volume is not that important. However, the basic question is whether there is any point in even considering the tailoring of cheap components. It is possible that tailoring expenses are greater than profits per component. The price issue affects license agreements, too. It can be questioned if it is reasonable to license “small” components with “low” prices, at least if the selling volume is “low”.

10.4 IPR-related issues

Intellectual property rights are a big concern in the OCM business for many reasons. IPR policies are not well structured even in the software industry in general, not to speak of the software component business, which is still in the beginning of its development. For example, licensing may be more complex in the case of software components than in the whole software products field merely because of the “small” size and “low” price of the components.

Also the need for a component to be as flexible as possible easily results in problems in making appropriate license agreements. Because the actual use of the component may be quite hard to define beforehand from the OCM company's viewpoint, the control over IPRs can become a serious problem to the supplier company. Depending on who does the possible tailoring of the component, there arises the question of how the IPRs of the modified parts of the component are divided. One possible solution is shared IPRs, but they do include some major risks, too.

Because the software component is going to be assembled and used as a part of another product – which may be a piece of software – the OCM company usually needs to have access to the source code of the component. The consequences of such access are extremely difficult to supervise, the buyer may be able to change any part of the source code any time.

In more general terms, intellectual property is a problem that should be carefully designed and managed in software companies. In managing IPRs, one of the main challenges lies in navigating a minefield of existing or pending patents, both to produce goods that do not infringe on other companies' rights and to vouchsafe rights that will secure the companies' own market share. [Davis and Weitz 1997]

There are many issues related to intellectual property rights that should be taken into account in a software firm, or in any technology-driven company. Firstly, they should have written confidentiality policies and require every employee to sign a confidentiality agreement. Also, at least the technical employees and management should be required to execute assignment-of-invention and non-competition agreements. However, the need for non-competition agreements is dependent on the state law, for example in Finland there is no need for such agreements. Secondly, technology-driven companies should require consultants, contractors, joint developers and suppliers also to execute these kind of agreements. Thirdly, they should make a written agreement that specifies the desired ownership arrangement and contains the necessary assignment, license and royalty provisions in the case of joint development projects. Fourthly, it would be reasonable to maintain written and dated invention records because a company with accurate documentation will be better able to support a claim of an earlier date of invention. [Cerrone and Villeneuve 1996]

Computer programs are proprietary works under many countries' copyright laws, for example under the U.S. copyright law. Both source code and object code are protected, although a "rule of doubt" is issued for object code. However, the copyright law only protects the originality of expression, not the underlying idea or function. [Halligan 1995] In other words, copyright laws protect the form of expression, but not the idea itself. With respect to software, this typically means that the computer program, in both human-readable and machine-executable form and the related manuals are valid for copyright protection, but the methods and algorithms within a program are not protected [Anon. 2001e]. Therefore, by definition, copyright protection alone is not adequate to protect the commercial value of software [Halligan 1995].

The demand of originality may prevent achieving copyright protection for some generic software components. However, this is more relevant question in the COTS case, not so much in the OCM context, where there usually occurs some level of tailoring.

Coupled with copyright protection, trade secret protection is an important means of protecting software. In fact, together these two often provide the most effective way to protect the software. Whenever possible, the software should be protected under the trade secrets law. If the source code contains information that can be classified as a trade secret, trade secret protection can be achieved. In cases where the software is sold or licensed to another party, such contracts should be used which require the receiving party to maintain the trade secret status of the software and not to disclose or use the software except in accordance with express terms of the license or contract. [Halligan 1995]

The trade secret question is very relevant in the OCM relationship, because in the case of tailoring the software component there is usually a need to reveal the source code to the other party. In this case, appropriate licensing and contract terms should be used to protect the owner's rights.

Patentiability issue related to software is a complex subject, too, for example regarding the commercial value of the software. If the commercial life of the software is only few years, as it may rather often be, patent protection may not make much sense – as it will take at least two or three years to issue the patent. Further, the issuance of a patent will destroy the trade secret status of the software. That is because once the patent issues, the patented features of the invention will be in the public domain. [Halligan 1995] Also the fact that obtaining a patent is an expensive task prevents many software developers from seeking it [Chávez et al. 1998]. In the case of OCM software components, the question of patentiability may be even less important because the expected commercial value of a certain software component may not be high enough to fulfill the patent requirements.

Virtually any piece of software purchased annually is licensed, not sold outright [Sicilian and Underhill 1996]. The difference between a license sold and an outright sold is that the license deal precludes from enjoying all of the benefits of ownership. Invariably, license deals contain restrictions on copying or transferring the software. For example, some licenses provide that the license is cancelled, if the licensee makes an unauthorized copy or sells the software to another user. Others permit resale of the software, but only if the licensee does not retain any copies and only if the buyer agrees to the terms of the license.

Software licenses also vary by the number of people allowed to use the software. Individual licenses permit only single users, whereas suite licenses allow a defined number of users, and site licenses or company licenses typically allow an unlimited number of users at a single company. [Sicilian and Underhill 1996]

In the case of software components, the licensor owns the intellectual property rights of the software component, usually because the licensor has developed the component. In a license deal the licensor will allow the licensee the right to use those property rights related to the component, as an analogous way to whole software products. However, assumptions regarding the size, quality and flexibility of software components differ from those underlying typical licensing scenarios. These three assumptions affect the license grant, payment, ownership, liability, warranty, maintenance and confidentiality terms of software component licenses. [Chávez et al. 1998]

A license between the component owner and the licensee should at least specify the rights the licensor authorizes the licensee to exercise in the software i.e. license grant, payments to the licensor, support, maintenance and warranties for the licensed component and who owns the licensed component and modifications to it. Also the risks and liability each party assumes under the license and the confidentiality of the licensed component should be specified. [Chávez et al. 1998] In the OCM case, the question of modifications is usually important.

The modification question may be solved so that the supplier takes the responsibility to carry out the necessary modifications. However, the OCM company must in this case trust on the supplier so much that it is ready to give enough information concerning the whole product. On the other hand, the OCM can itself do the modifications. In this case, the problem is that the supplier should now trust the OCM enough to reveal the necessary parts of the source code. This problem can be solved either by an escrow agreement, in which the source code is returned once the defined conditions have changed, or by giving only a necessary part of the source code and make sure that it will be used only as specified [Rosenberg 2000].

License agreements that fail to affirm the flexibility, quality and interoperability of software components attenuate the credibility of the software reuse movement. On the other hand, license agreements that certify the component's quality and the licensor's ongoing commitment to maintain it cultivate the reputation of excellence that make components so appealing to licensees. For the licensor, a successful license agreement that safeguards his ownership of the component and maintains its confidentiality offers appropriate financial incentives and in this way increases the willingness of component developers to license their components onto the use of a third party. [Chávez et al. 1998]

From the suppliers point of view, the problem in the OCM case may be the difficulty to get the OCM to believe that its customers want the specific functionality that the supplier's software component can provide. Also, the supplier has to get the OCM to believe that it is faster and cheaper and more reliable to license it from the supplier than to develop it in house or buy it from someone else. [Rosenberg 2000]

10.5 Summary

The cooperation between OEMs and suppliers has increased in many industries, for cooperation is at great part carried out in the product development process, which the first-tier suppliers have been taken more and more responsibility of.

At the same time, for example in the electronics industry, the increasing complexity of products have created a trend, in which the design work is emphasized on an earlier phase of the product development compared to how it was done before [Soininen 1997]. It can be argued that in the future, suppliers will also participate in the product design process more than they have done so far.

The issues discussed above are illustrated in Figure 13.

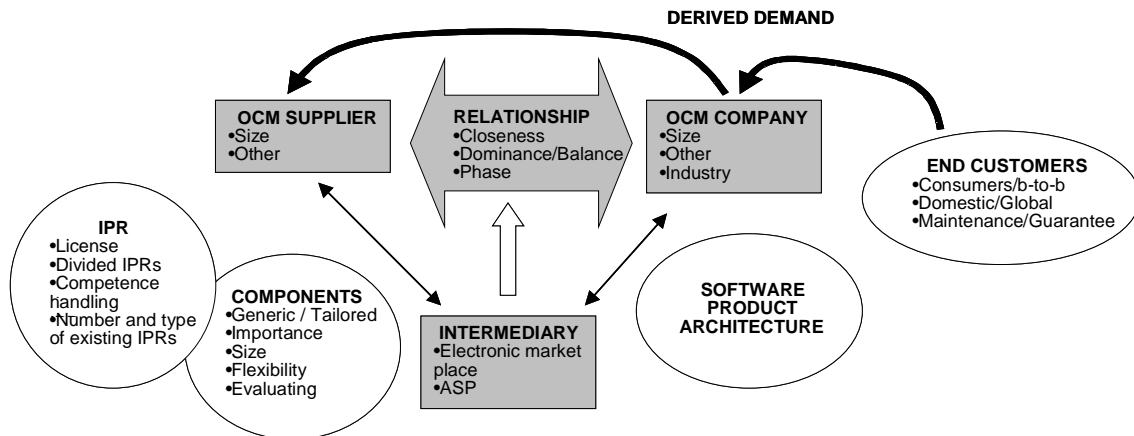


Figure 13. The detailed OCM context.

The OCM business is mostly characterized by the closeness of the relationship between the supplier and the buying OCM company. Close relationships have positive and negative effects. They also raise the question whether it is possible to have an intermediary in the OCM relationship. In Table 3 the opportunities and problems of the OCM business are gathered together concerning three different viewpoints: the viewpoints of the supplier, the OCM company and the possible intermediary.

Besides these issues, also the larger business network in which both the supplier and the OCM company operate, affects the success or failure of an OCM relationship. As discussed earlier in the report, the Japanese network is rather vertically structured, whereas the European and Finnish way of network structuring is rather horizontal. However, a shift towards more vertical networks can be seen in the European industries as well, at least in the automotive industry. This is due to the fact that big OEM companies want to give more responsibility to few large and competent suppliers. On one hand, the increased first-tier supplier responsibilities decrease the quantity of direct buying relationships from the viewpoint of the OEM. On the other hand, the largest suppliers will have suppliers and subcontractors of their own. It would be interesting to study this kind of a shift from horizontal to vertical networks in the software industry.

Table 3. Problems and opportunities in the OCM business.

	Opportunities	Problems
OCM Supplier	<ul style="list-style-type: none"> • Easy distribution channel • Can help to reach new end user groups • Does not require brand building skills • Support from the OCM company, e.g. educational 	<ul style="list-style-type: none"> • How to get the intention of a big OCM • How to deal with IPR issues • The lack of the visibility of one's own brand wrt. COTS markets • Profit margin • Generality vs. specificity of the component • Component maintenance
OCM Company	<ul style="list-style-type: none"> • Can concentrate on one's own core competence • New ideas from supplier to product development 	<ul style="list-style-type: none"> • Finding good suppliers • Component evaluation • Component maintenance • Shared IPRs • Can tailoring needs be taken into account? • Existence of product architecture • Component tailoring needs
Inter-mediary	<ul style="list-style-type: none"> • Can help in finding right partners • Can help in IPR issues • Can help in component evaluation • Possible value added offerings 	<ul style="list-style-type: none"> • Is an intermediary possible in close supplier – buyer relationships? • Electronic market places vs. component tailoring • ASP: is it possible to rent a component to a product that is sold further?

11. Suggestion for further R&D

Based on the analysis of the information presented above, we will outline in the following the actions that could be taken to carry R&D on original software component manufacturing further. We will start by presenting the context in which these actions would be taken, based on revising the initial view discussed above. The revised context is shown in Figure 14.

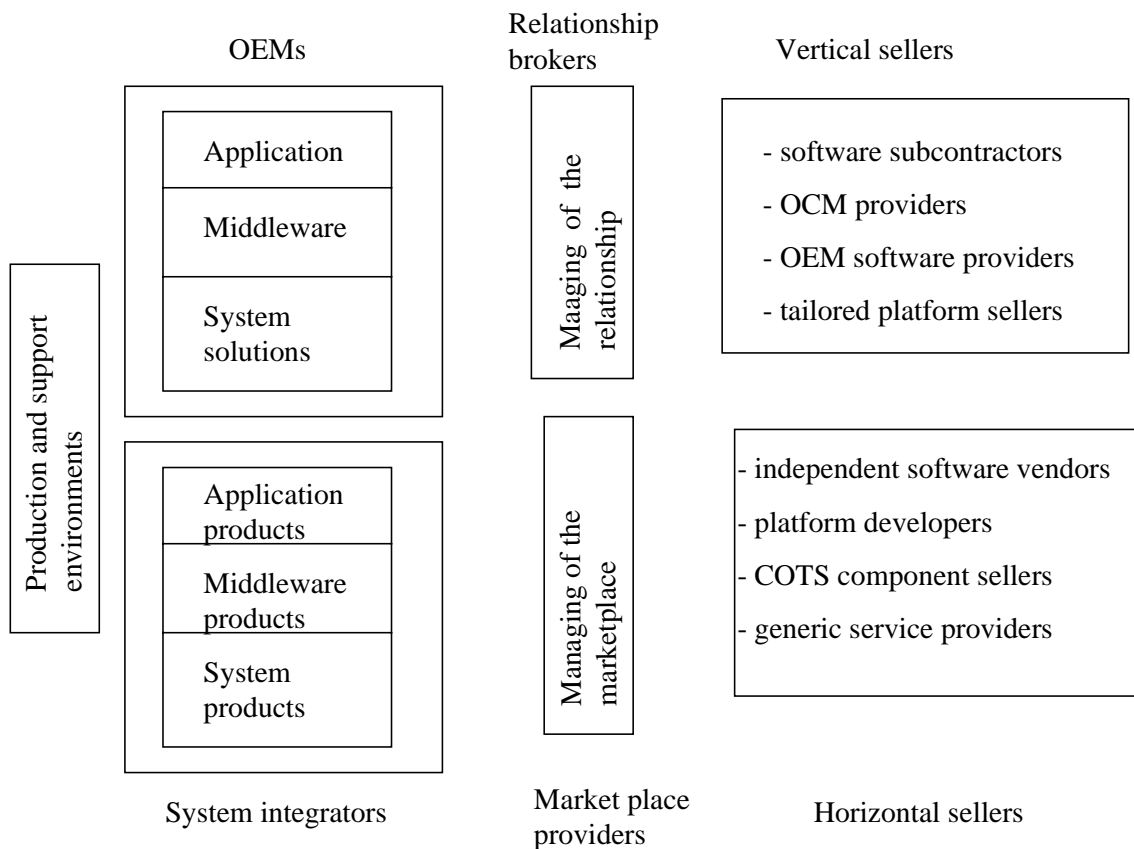


Figure 14. Context for OCM research and development.

First of all, the context of OCM is divided in Figure 14 into three areas corresponding to what has been said earlier in this report: buyers, intermediaries, and sellers. However, the buyers are considered as system integrators or OEM companies (depending on the visibility and modifiability of the purchased components or software products), the sellers as vertical (including OEM software providers among others) or horizontal (including e.g. COTS component sellers), and the intermediaries as relationship brokers that mainly serve vertical sellers or market place providers that focus on horizontal sellers and buyers.

The buyers' needs are viewed through a three-tier product architecture, supplemented with the systems that are needed to build products, such as software development and testing environments. The "small" or "large" number of software products or components to be exchanged has a key effect on all the three types of actors. On the buyer's side, the corresponding effect could be emphasized further by using the concepts of a production company (small number of purchased parts) and an assembly company (large number of purchased parts), but this difference has been omitted.

Based on Figure 14, at least three different types of foci can be taken to R&D in the OCM context: development of component market places and partner relationships, process creation and improvement, and product-oriented R&D.

Development of component market places and partner relationships

The first R&D area would include an active development of market place management services in particular. One starting point could be an identification of a few large buyers and a new or newly established market place management firm. An alternative would be to help domestic companies to make use of existing global software component market places, such as the two discussed earlier in this report. Software component developers should be helped to exploit market places, be they domestic or global.

If domestic market places were established, one of the most obvious opportunities would be to focus on a certain domain of applications, e.g. mobile Internet-based applications. However, strong participation of the future buyers of the components would be needed. One possibility might be to establish a software component buyers' circle in the domain of some applications. The circle could at first test-use the existing market places, but aim finally at helping to establish a new market place.

Process creation and improvement

What is actually not shown in Figure 14 could be one of the most important topics for further research: making explicit the interplay of the processes of various types of actors. Taking the OCM context into account, there are at least the following problems to study with regard to the processes.

Definition and optimization of marketplace-driven processes: offering, marketing, evaluating and delivering of software components, products and platforms; analyzing markets for the needs of different parties; helping vendors and buyers to make use of the market place processes; building and management of software delivery and maintenance channels (cf. ASP); definition and optimization of system integration and OCM software production processes, and optimization of software acquisition.

Improvement of relationship-based processes: productization of subcontracted software; building of product lines and system family platforms; screening and evaluation of relationship partners; building of contract-based software supplier networks; establishing system integration and OCM environments; and mixing of COTS products and customized pieces of software.

Component-based software development has already changed, and will continue to change, the internal processes of the most advanced software developer organizations. What has not been achieved, though, is the definition and effective use of processes meant to help using external products, components and platforms. The strategic change of the biggest buyer companies towards product lines, software platforms and component-based software development could be used as a driving force behind the “externalization” of component-based software production. What should be done in general is to make use of the existing software process improvement expertise for the needs of the acquisition of external pieces of software, to be integrated into the buyer’s products through a process in which both the buyer and the seller participate.

Product-oriented R&D

Perhaps the most obvious direction of further R&D, taking into account what is already known and has been done in the nineties, is to follow the product-oriented view: in other words, to continue to develop software architectures (both generic reference models and domain- or customer-specific platforms), component principles, and middleware type integration techniques. Product-line architecture research will, for example, result in better possibilities for the biggest buyers to make use of external components, products and platforms. However, as part of the product-line architecture research, not only different types of techniques to evaluate acquired pieces of software should be studied, but also environments by which buyers and sellers can rapidly test new ideas and existing solutions. Various types of simulation environments are available for application developers. These could be exploited and instrumented for the needs of “ultra-rapid prototyping” based on software components.

Computer applications will become more distributed and flexible than what they used to be, and the “lowest” (enabling technologies) and “highest” (service and network access) platforms will become standardized. What is needed “in between” is an interesting question. Big buyers will make use of product lines to structure the “intermediate” level, but suppliers will have hard times to segment their offerings. They should be helped to develop software that can be used in similar product lines of different buyers. Ultimately, this will lead to the biggest component sellers’ own product lines.

References

- Alajoutsijärvi, K. 1991. Investointihyödykkeen markkinointi. Jyväskylän yliopisto, taloustieteen laitos, julkaisuja no 83/1991. 51 p. (in Finnish)
- Amuso, C., Fluss, D., Hope-Ross, D. & Ross, C. 1999. Customer Service and Support Strategies. Gartner Group.
- Anderson, J.C. & Narus, J. A. 1998. Business Marketing: Understand What Customers Value. Harvard Business Review, Nov/Dec 98, Vol. 76, Issue 6.
- Anderson, J.C. & Narus, J.A. 1999. Business Market Management. Understanding, Creating and Delivering Value. New Jersey: Prentice Hall.
- Anon. 1999a. The Virtual Supply Chain. Part 1. 1999. November. <http://www.startmag.com/v3n9/v3n9p060.asp>. 16.2.2001
- Anon. 1999b. The Evolution of Unit-Level Supply Chain Manufacturing. <http://www.datasweep.com/>. 16.2.2001.
- Anon. 2000a. Understanding the ASP Market. An ISV's Guide to Transitioning from Packaged Product to Online Service. Sound Consulting, LLC. Document No. ASP-0600-021. 19 p.
- Anon. 2000b. ASP Pulse. Computer Reseller News. Aug/2000, Issue 909, p. 55 - 56.
- Anon. 2000c. Metalli- ja elektroniikkateollisuuden päätoimialat pähkinänkuoressa. Lähde: Tilastokeskus, Tullihallitus 20.3.2000. <http://www.met.fi/images/pahki99.gi> (in Finnish), 21.2.2001
- Anon. 2000d. Metalli- ja elektroniikkateollisuuden suhdannetiedote 3/2000. <http://www.met.fi/ajankohtaista/suhd300.htm> (in Finnish), 21.2.2001.
- Anon. 2001a. A horizontal approach to vertical markets. Bonita software. <http://www.vertogo.com/papers/horizontal.htm> 2.2.2001
- Anon. 2001b. Datasweep's Architecture for Web-Centric Supply Chain Manufacturing. <http://www.datasweep.com/> 2.2.2001
- Anon. 2001c. <http://www.zdnet.com/spacial/isp/stories/glossary/>. 19.2.2001

Anon. 2001d. The Future Internet Impact Resulting from Network Hosted Applications. <http://www.aspisland.com/opinion/mandelbaum1.asp> 19.2.2001

Anon. 2001e. Copyright Protection. <http://www.softwareprotection.com/copyright.html>. 21.2.2001.

Autere, J., Lamberg, I. & Tarjanne, A. 1999. Ohjelmistotuotteilla kansainväliseen menestykseen. Toimialan kehittämistarpeet ja toimenpide-ehdotukset. Tekes, Teknologiaakatsaus 74/99. (in Finnish)

Auvinen, A., Tunkelo, E. & Lehtomäki, P. 1993. Tuotantohyödykkeiden markkinointi. Opintomoniste 2/1993. Oulun yliopisto, Industrial Engineering and Management Laboratory. 138 p. (in Finnish)

Axelsson, B. & Håkansson, H. 1990. The Development Role of Purchasing in an Internationally Oriented Company. In: Ford, D. (ed.) 1990. Understanding Business Markets: Interaction, Relationships and Networks. Academic Press. London. Pp. 318 - 333.

Axelsson, B. & Easton, G. (eds.) 1992. Industrial Networks. A New View of Reality. Routledge. London. 265 p.

Aoyama, M. & Yamashita, T. 1998. Software Commerce Broker over the Internet. IEEE 9/98, pp. 430-435.

Bairstow, L. 1999. The Virtual Supply Chain. Start Magazine, November 1999, Vol. 3, No. 9.

Bazerman, M.H. & Gillespie, J.J. 1999. Betting on the Future: The Virtues of Contingent Contracts. Harvard Business Review, Sept-Oct, pp. 155-160.

Blackwell, G. 2000. ERP Vendor Committed to ASP Model. [<http://www.aspisland.com/focus/jdedwards2.asp>]

Bowersox, D.J. & Cooper, M.B. 1992. Strategic Marketing Channel Management. McGraw-Hill, Inc. 488 p.

Cerrone, K.L. & Villeneuve, T. 1996. Power to the Patents. How to maximize the value of your company's technology portfolio. [<http://www.ipmag.com/cerrone.html>]

Chávez, A., Tornabene, C. & Wiederhold, G. 1998. Software Component Licensing: A Primer. IEEE Software, Sept/Oct 1998, pp. 47 - 53.

Chisnall, P. M. 1995. Strategic Business Marketing. Prentice Hall. UK. 506 p.

Cook, W. J. 1996. Deputizing the ISPs. [<http://www.ipmag.com/acook.html>]

Cross, R. & Smith, J. 1997. The Customer Value Chain. American Demographics, Jan/Feb 97, pp. 14 - 18.

Cunningham, M. T. & Homse, E. 1990. Controlling the Marketing-Purchasing Interface: Resource Development and Organisational Implications. In: Ford, D. (ed.) 1990. Understanding Business Markets: Interaction, Relationships and Networks. Academic Press. London. Pp. 104 - 123.

Davis, P. & Weitz, D. 1997. Leveraging your IP portfolio. [<http://www.ipmag.com/davis.html>]

Dent, S.M. 1999. Partnering Intelligence. Creating Value for Your Business by Building strong Alliances. Davies-Black Publishing, California. 239 p.

Dwyer, F.R., Schurr, P.H. & Oh, S. 1987. Developing Buyer-Seller Relationships. Journal of Marketing, Vol. 51, April, pp.11 - 27.

Ellram, L.M. 1990. The Supplier Selection Decision in Strategic Partnerships. Journal of Purchasing and Materials Management, Vol. 26, No. 3, pp. 8 - 14.

Fisher, M., Ramdas, K. & Ulrich, K. 1999. Component Sharing in the Management of Product Variety: A Study of Automotive Braking Systems. Journal of the Institute for Operations Research and Management Sciences, Vol. 45, No. 3, pp. 297 - 315.

Ford, D. 1980. The Development of Buyer-Seller Relationships in Industrial Markets. European Journal of Marketing, Vol. 14, Issue 5/6, pp. 339 - 354.

Ford, D. 1990. Understanding Business Markets: Interaction, Relationships and Networks. Academic Press. London. 556 p.

Ford, D., Gadde, L.-E., Håkansson, H., Lundgren, A., Snehota, I., Turnbull, P. & Wilson, D. 1998. Managing Business Relationships. John Wiley & Sons Ltd. England.

Franks, J. 2000. Supply Chain Innovation. MCB Work Study, Vol. 49, Issue 4.

- Fusco, P. 2000. Analysts Send Mixed Signals to ASPs. [http://www.aspnews.com/news/weekly/article/0,2350,4271_439101,00.html]
- Gadde, L.-E. & Håkansson, H. 1993. Professional Purchasing. Routledge. London. 186 p.
- Gecowets, G.A. & Bauer, M.J. 2000. The e-effect of the Internet on supply chain & logistics. World Trade Magazine, Vol. 13, Issue 9, pp. 71 - 78.
- Graham, G. & Hardaker, G. 2000. Supply-chain management across the Internet. International Journal of Physical Distribution & Logistics Management, Vol. 30, Issue 3 / 4. 7 p.
- Grice, C. 2000. Phone giants may gain most from ASP market. [<http://news.cnet.com/news/0-1004-200-1573662.html>]
- Grönroos, C. 1997. Relationship Marketing: Interaction, Dialogue and Value. Swedish School of Economics and Business Administration. Working Papers No 344.
- Gummesson, E. 1997. Relationship Marketing as a Paradigm Shift: Some Conclusions from the 30R Approach. Management Decision, Vol. 35, Issue 4.
- Haarmann, P.-L. 1994. Immateriaalioikeuden oppikirja. Lakimiesliiton kustannus. Helsinki. 224 p. (in Finnish)
- Halinen, A. 1994. Exchange Relationship in Professional Services: A Study of Relationship Development in the Advertising Sector. Turku School of Economics and Business Administration, A-6. Turku.
- Halligan, R.M. 1995. How to protect intellectual property rights in computer software. [<http://execpc.com/-mhalign/computer.html>]
- Heikkilä, J., Saarinen, T. & Sääksjärvi, M. 1991. Success of software packages in small business. Helsinki School of Economics, Working Papers F-260. 25 p.
- Hoch, D.J., Roeding, C.R., Purkert, G., Lindner, S.K. & Müller, R. 1999. Secrets of Software Success. Management Insights from 100 Software Firms around the World. Harvard Business School Press, Boston. 309 p.
- Hutt, M. D. & Speh, T. W. 1998. Business Marketing Management: A Strategic View of Industrial and Organizational Markets. The Dryden Press. 777 p.

- Hyötyläinen, R. 2000. Development mechanisms of strategic enterprise networks: Learning and innovation in networks. Espoo, Technical Research Centre of Finland, VTT Publications 417. 142 p.
- Håkansson, H. & Johanson, J. 1992. A Model of Industrial Networks. In: Axelsson, B. & Easton, G. (eds.) 1992. *Industrial Networks: A New View of Reality*. Routledge. London. Pp. 28 - 34.
- Jensen, A. 1993. Strategies against market power in marketing channels. *Scandinavian Journal of Management*. June 1993, Vol. 9, No. 2, pp. 101 - 128.
- Karlsson, C. & Norr, C. 1994. Total Effectiveness in a Just-In-Time System. *International Journal of Operation & Product Management*, Vol. 14, Issue 3.
- Kivi-Koskinen, T. 1999. Teollisoikeudet pienen ja keski-suuren yrityksen kilpailukeinona. Tekes, Kansainvälisten verkostojen raportti 2/98. Helsinki 1999. 60 p. (in Finnish)
- Kotsalo-Mustonen, A. 1995. Mystery of Success. An analysis of system sale as an outcome of industrial buyer-seller interaction process. Helsingin kauppakorkeakoulun julkaisuja B-151. Helsinki. 135 p.
- Kuivalainen, O., Saarenketo, S. & Jääskeläinen, J. 2000. Creating More Value Through the International Network – The Case of Business-to-Business E-Commerce Software Producer and Value Added Resellers. IMP Conference Paper. 9 p.
- Laine, K., Penttinen, H. & Kotsalo-Mustonen, A. 2000. Sähkö- ja elektroniikka-alan palvelujen kysynnän ja tarjonnan kohtaamisesta. *Teknologiakatsaus 94/2000*. Tekes. Helsinki. (in Finnish)
- Lamming, R. 1993. *Beyond Partnership. Strategies for Innovation and Lean Supply*. Prentice Hall, UK. 299 p.
- Lehtinen, U. 1996. Partnerships among Finnish Manufacturers. Case study on the development of supplier structure and supplier-customer relationships. IPSERA 1996, Conference Eindhoven, pp. 473 - 482.
- Lindqvist, J. & Manninen, J. 1998. Asiantuntijat tulevaisuuden tekijöinä. Asiantuntijapalveluyritysten tulevaisuudenkuvat. Helsingin yliopiston Lahden tutkimus- ja koulutuskeskus. Raportteja ja selvityksiä 29. Gummerus Kirjapaino Oy. (in Finnish)

Lorange, P. & Roos, J. 1992. Strategic Alliances. Formation, Implementation and Evolution. Blackwell Publishers. 295 p.

MacStravic, S. 1999. The Value Marketing Chain in Health Care. Marketing Health Services, Chigago, Spring 99. Vol. 19, pp. 14 - 19.

McIvor, R. & McHugh, M. 2000. Partnership Sourcing: An organization change management perspective. Journal of Supply Chain Management, Vol. 36, Issue 3, pp. 12 - 29.

Messerschmitt, D.G. & Szyperski, C. 2001. Industrial and Economic Properties of Software: Technology, Processes, and Value, University of California at Berkeley Computer Science Division Technical Report UCB//CSD-01-1130, Jan. 18, 2001, and Microsoft Corporation Technical Report MSR-TR-2001-11, Jan. 18, 2001. (available from <http://divine.eecs.berkeley.edu/~messer/PAPERS/01/Software-econ/>)

Moore, G. A. 1991. Crossing the Chasm. Marketing and Selling High-Tech Products to Mainstream Customers. Harper Business.

Morgan, R.M. & Hunt, S.D. 1994. The Commitment-Trust Theory of Relationship Marketing. Journal of Marketing, Vol. 58, No. 3, pp. 20 - 38.

Möller, K.E. & Wilson, D.T. 1989. Interaction perspective in business marketing: An exploratory contingency framework. Working Papers F-233. Helsinki School of Economics and Business Administration.

Niemelä, E., Kuikka, S., Vilkuna, K., Lampola, M., Ahonen, J., Forsell, M., Korhonen, R., Seppänen, V. & Ventä, O. 2000. Teolliset komponenttiohjelmistot. Kehittämistarpeet ja toimenpide-ehdotukset. Teknologia katsaus 89/00. Tekes, Helsinki. (in Finnish)

Niemelä, E. & Seppänen, V. 2000. Tuotelinjasta tehoa ohjelmistokehitykseen: Viiteotsikko: Ohjelmistokomponentit ja -kehykset teollisissa tuotteissa. Proessori, November 2000. (in Finnish)

Nukari, J. & Forsell, M. 1999. Suomen ohjelmistoteollisuuden kasvun strategia ja haasteet. Tekes, Teknologia katsaus 67/99. (in Finnish)

Paul, L. 2000. CenterBeam: old method, new method and the transition process. [<http://www.aspisland.com/success/centerbeam2.asp>]

Prahalad, C.K. & Hamel, G. 1990. The Core Competence of the Corporation. Harvard Business Review, May-June, pp. 79 - 91.

Reeder, R. R, Brierty, E. G. & Reeder, B. H. 1991. Industrial Marketing: Analysis, Planning and Control. Prentice Hall. 670 p.

Robinson, W.N. 1997. Electronic Brokering for Assisted Contracting of Software Applets. IEEE 1997. Pp. 449 - 458.

Rosenberg, D.K. 2000. Stromian's OEM Software Licensing Guide. [<http://www.stromian.com/giude.htm>]

Saarnilehto, A. 1996. Sopimusoikeuden perusteet. Lakimiesliiton kustannus. 154 p. (in Finnish)

Salo, A., Pahlavan, K. & Salmenkaita J.-P. 2000. R&D Programmes in Electronics and Telecommunication. ETX, TLX, INWITE and Teletronics. Technology Programme Report 5/2000. Mid-Term Evaluation. Tekes. Helsinki.

Sicilian, J. & Underhill, S. 1996. The Software Police Take A Hard Line. [<http://www.ipmag.com/softcop.html>]

Spear, S. & Bowen, H.K. 1999. Decoding the DNA of the Toyota Production System. Harvard Business Review, Sep-Oct, pp. 97 - 106.

Soininen, J.-P. 1997. Asiakstarvelähtöisyys elektronisen tuoteperheen suunnittelussa. Technical Research Centre of Finland, Espoo. VTT Publications 822. (in Finnish)

Sperling, E. & Gage, D. 2000. ASPs Look For Their Own Channel. [<http://www.zdnet.com/sp/stories/news/o,4538,2565669,00.html>]

Sprott, D. 2000. Open Market Components. A CBDi Forum Report, January 2000. [http://www.componentsource.com/AboutUs/cbdiopen_market.asp]

Storbacka, K., Blomqvist, R., Dahl, J. & Haeger, T. 1999. Asiakkuuden arvon lähteillä. WSOY. Porvoo. (in Finnish)

Storbacka, K. & Lehtinen, J.R. 1997. Asiakkuuden ehdoilla vai asiakkaiden armoilla. WSOY. Porvoo. (in Finnish)

Storbacka, K., Sivula, P. & Kaario, K. 1999. Create Value with Strategic Accounts. Kauppakaari Oyj. Helsinki. 175 p.

Sääksjärvi, M. 1998. Tuoterunko. Uusi ajattelu ohjelmistotuotteiden strategisessa kehittämisessä. Teknologiakatsaus 62/98. Tekes. Helsinki. (in Finnish)

Tapscott, D., Ticoll, D. & Lowy, A. 2000. Digital capital: harnessing the power of business webs. Harvard Business School Press.

Taskila, H., Hyry, M. & Toiviainen, E. 1995. Automaatioyrityksen strategiset suunnitelmat – liiketoiminnan perusta. [<http://www.tacnet.sci.fi/opetus/autom95/ag/41/1esitelm.htm>] (in Finnish)

Torode, C. & Hagendorf-Follet, J. 2000. Empowering SMBs With Hosted Services – Selling ASP services presents new opportunity for SMB solution providers. CRN, Aug/2000, Issue 908. [<http://www.techweb.com/se/directlink.cgi>]

Wainewright, P. 2000. Anatomy of an ASP – Emergence of the ASP Channel. [http://www.aspnews.com/profiles/asp_basics/article/0,2350,4521_388191,00.html]

Webber, S. 2000. European developments in intellectual property. American Society for Information Science. Bulletin of the American Society for Information Science, Vol. 26, Issue 4, pp. 28 - 29.

Wilde, C. 2000. Telco's answer customer's calls for more services. Informationweek. Aug/2000, Issue 800.

Zwicker, D. 2000. Should you build your own VPN or outsource? Build Network World. Aug/2000, Vol. 17, Issue 33.



Author(s) Seppänen, Veikko, Helander, Nina, Niemelä, Eila & Komi-Sirviö, Seija			
Title Towards original software component manufacturing			
Abstract <p>This report is a deliverable of a self-funded strategic pre-study carried out by VTT Electronics in 2000 on software products developed for and used by the electronics, automation and telecommunication industries. The focus of the study was on software component products, especially on such software components that can be tailored by component suppliers or buyers before their actual use. In the report, this type of business is called original software component manufacturing, referring to the analogy of original equipment manufacturing that is well-known in other industries. Another phrase that could be used is partnership software component production.</p> <p>However, there is a continuum from tailored software components that are developed by the product manufacturer itself, who is committed to component-based software engineering, to fully commercially supplied off-the-shelf components. Therefore, this report includes certain aspects along the entire continuum. An important issue related to this is to analyze how and why a supplier of tailored components would develop towards off-the-shelf products. A related question on the buyer's side is the change of an independent product manufacturer to a system integrator that puts together products with the help of fully commercial component suppliers.</p> <p>One of the main goals of the report, which was produced mostly based on information available on the Internet, is to try to structure the present state of software components, businesses and production processes. The information that was gathered includes therefore both business-related and technical subjects, as well as the views of component developers, buyers, sellers, and so called brokers. The latter is emerging as an intermediary role in software component distribution. The role of a broker can also be played by a first-tier supplier, i.e. a company that manages a portion of the supplying network for the system integrator.</p> <p>Regarding these roles, the information acquired from the Internet was supplemented by interviewing a few companies that mostly represent potential software component sellers. Overall, the information indicates that there is no well-organized information on original software component markets and businesses available yet. To compare, the acronym "OEM" (original equipment manufacturing) produced hundreds of relevant Web links about parts and product manufacturing in mechanical and other industries. In order to pave the way towards the structuring of the field, an integrative view is taken at the end of this report using the roles and processes of different firms involved. Based on this view, a suggestion is made for further research and development activities.</p>			
Keywords original equipment manufacturing, software component markets, component-based software engineering, software acquisition, software supply, software booking			
Activity unit VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box1100, FIN-90571 OULU, Finland			
ISBN 951-38-5822-7 (soft back ed.) 951-38-5823-5 (URL: http://www.inf.vtt.fi/pdf/)		Project number E0SU00291	
Date May 2001	Language English	Pages 105 p.	Price C
Name of project Oktetti		Commissioned by VTT Electronics	
Series title and ISSN VTT Tiedotteita – Meddelanden – Research Notes 1235-0605 (soft back edition) 1455-0865 (URL: http://www.inf.vtt.fi/pdf/)		Sold by VTT Information Service P.O.Box 2000, FIN-02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374	