1.1 Introduction

Competition forces firms in the software industry to reduce development costs, shorten time-to-market, and improve software quality and maintainability. In response to these competitive pressures, software companies increasingly try to “industrialize” their development processes by constructing new software systems from existing software rather than building them from scratch (Greenfield and Short, 2004). Systematic application of software reuse allows software companies to create an internal supply chain of reusable software assets: one development department may become responsible for populating and maintaining a library of reusable components, while another development department may become responsible for transforming these components into finished systems that can be delivered to customers. In addition to reusing and extending internally produced components, software companies may also decide to purchase one or more externally developed components from the market. This results in software supply chains that span both internal development departments as well as external suppliers.

Supply chains of physical goods have already been studied elaborately since the mid 1980s. This has resulted in a large body of knowledge in the form of

1. descriptive concepts, such as customer-order decoupling point (Hoekstra and Romme, 1992), production phase (Bertrand et al., 1998), and just-in-time delivery (Stock and Lambert, 2001);

2. empirical statements explaining the (causal) relationships among these concepts, such as the relationship between information sharing and supply chain performance (Slack et al., 2007);

3. normative methods and instruments (qualitative as well as quantitative), such as the economic order quantity model for deciding how much to order of any particular item when stock needs replenishing (Stock and
Lambert, 2001; Slack et al., 2007), and Fisher’s framework for aligning the supply chain structure with the type of product produced (Fisher, 1997).

The supply chain of (enterprise) software, however, has not been studied in the same rigorous way. Although the term supply chain is occasionally mentioned in the computer science literature, most authors focus on technical issues like communication protocols, component models, and standards for web service discovery, description, and composition. Less research has been conducted on organizational and managerial aspects, such as how software companies structure their supply chains and how they manage the relationships with their suppliers and customers. Noteworthy exceptions include the work of Messerschmitt and Szyperski (2003) on the “software ecosystem”, the work of Brereton (2004) on the characteristics of the customer/supplier relationship across three different procurement paradigms, and the work of Gao and Iyer (2008) on partnerships between software companies that produce in adjacent layers of the “software stack”.

In this thesis, we will continue along the line of Messerschmitt and Szyperski (2003), Brereton (2004), and Goa and Iyer (2008) by developing concepts and tools for the analysis and design of the software supply chain in the enterprise domain. Although the software industry is much broader, we will focus on enterprise software because the supply chains have already considerably evolved in this part of the software industry: enterprise software is delivered to customers through an extensive network of specialized dealers and resellers, and producers of this type of software generally incorporate some externally produced components into their products.

This introductory chapter is organized as follows. First, the reader is provided with some background information on the supply chain of physical goods as well as on the domain of enterprise software. Subsequently, the research objectives are formulated, which is followed by a brief outline of the remainder of the thesis.

1.2 The supply chain of physical goods

There is little consistency in the use of the term supply chain in the academic and business literature. Harland (1996) has identified four different interpretations:

1. the supply of goods within a firm’s boundaries;
2. the supply of goods within a dyadic customer-supplier relationship;

3. the supply of goods in a chain of businesses including a supplier, a supplier’s supplier, a customer, a customer’s customer, and so on;

4. the supply of goods in a network of firms involved in the provision of products and services required by end users.

In this thesis, the term supply chain is used to refer to the last of these four interpretations. Taking the manufacturer as our focal firm, we will use the term supply network to refer to the upstream part of the supply chain and the term distribution channel to refer to the downstream part of the supply chain. The term internal supply chain is used to refer to the supply of goods within the boundaries of the manufacturing firm.

A supply chain can be analyzed from many different perspectives. This thesis focuses on three main research areas: strategy, structure, and coordination. Each of these three research areas is briefly described in the sections below.

1.2.1 Strategy

Companies have to think carefully about how they want to position themselves against competitors that are active in the same market and therefore competing for the same customers. In a supply chain context, a company’s business strategy—i.e. the decision of the company of how it will compete (e.g. by providing high quality products or services), what goals it will have (in terms of market share, revenue, profitability targets, etc), and what policies it will support to achieve these goals (Harmon, 2007)—is important to consider because it determines the competitive priorities of the company’s manufacturing function. It also influences the way in which the company selects its suppliers (e.g. suppliers that focus on product performance, or suppliers that focus on cost) and how it chooses its distribution channels (e.g. sales through high-street retailers, or sales through specialty stores).

The range of products or services produced is one of the strategic decisions that manufacturing firms have to make. Lambel and Mintzberg (1996) have identified five distinct product delivery strategies, each corresponding to a different degree to which product design, manufacturing, and distribution are standardized. These are:

- pure standardization: a single product is commonly distributed to all customers;
• segmented standardization: a customer can choose among several predefined product variants;
• customized standardization: a customer can specify his or her own configuration by selecting from a set of standardized components;
• tailored customization: the company presents a product prototype to a potential buyer and then adapts or tailors it to the individual’s wishes or needs;
• pure customization: products are designed according to customer specifications.

When the strategies of pure standardization and segmented standardization are selected, the customer can choose between a limited variety of predefined products. Consequently, each of these products is sold in high volume, giving rise to economies-of-scale in production. The strategies of pure standardization and segmented standardization are therefore generally adopted by manufacturing firms that focus on providing low-priced products. The variety in finished products is much larger if one of the other three strategies is selected. In the case of customized standardization, the variety in finished products is caused by the variety of a limited number of standardized components (Erens, 1996). This makes customized standardization an appropriate strategy for manufacturing firms that focus on providing specialized, personal service: products are customized for each individual customer, while costs can still be kept down through high-volume production at the component level. Finally, when the strategies of tailored customization and pure customization are selected, also the intermediate products are produced in low volumes. These latter two strategies are therefore more likely to be adopted by manufacturing firms that focus on innovation and performance leadership.

1.2.2 Structure

When looking at supply chain structure, a distinction can be made between actors (i.e. the network structure of the supply chain) and activities (i.e. the business processes that are performed by these actors).
Network structure of the supply chain

Lambert and Cooper (2000) distinguish two dimensions along which the network structure of a supply chain can be described and analyzed: its horizontal structure (i.e. the number of tiers across the supply chain) and its vertical structure (i.e. the number of actors within each tier). The horizontal structure of a supply chain can be long, with many tiers, or short, with only a few tiers. Similarly, a tier can have a narrow vertical structure, with only a few companies, or a wide vertical structure, with many suppliers and/or customers. A car manufacturer, for example, generally has a limited number of first-tier suppliers, so the vertical structure of this tier is relatively narrow. The amount of second-tier and third-tier suppliers, on the other hand, is much larger. These two tiers are therefore characterized by a relatively wide vertical structure. The same holds for the car manufacturer’s distribution channel: its cars are sold through a relatively small number of importers (first-tier customers), who deliver them to a large number of car dealers (second-tier customers).

The network structure of a supply chain is influenced by the product delivery strategy of the manufacturer. For example, when the strategies of pure standardization, segmented standardization, customized standardization, and tailored customization are adopted, the manufacturer’s material requirements do not depend on the needs of individual customers. This allows the manufacturer to engage into long-term relationships with its key suppliers, thereby resulting in a relatively stable supply network. The case is different when the strategy of pure customization is adopted. Because the manufacturer’s products are now completely developed to customer specifications, a different group of suppliers may be required for each customer. As far as the structure of the distribution channel is concerned, the strategies of customized standardization, tailored customization, and pure customization favor a direct channel between the manufacturer and the end user. On the other hand, manufacturers that apply the strategies of pure standardization and segmented standardization generally sell their products through a large network of channel intermediaries, such as wholesalers and retailers.

Business processes

A supply chain can be seen as a system that transforms inputs, such as raw materials and customer orders, into outputs, such as finished products. Figure 1.1 illustrates this schematically by providing a high-level systems view
of a manufacturer’s internal supply chain. In this example, the system is composed of two interrelated subsystems, called processes, that contribute some part to fulfilling customer needs: (i) the planning and control process and (ii) the production process. The planning and control process responds to customer orders by generating production orders. These, in turn, trigger the production process to transform the purchased materials into finished products that can be delivered to the customer.

![Figure 1.1: A systems view of a manufacturer’s internal supply chain.](image)

The production process of a manufacturing firm can be accurately described by making a distinction between production phases and stock points (see Figure 1.1). During a production phase (rectangle), goods change intrinsically, whereas a stock point (triangle) indicates the storage of goods between two successive production phases. Stock in a stock point is waiting for a decision with regard to further transformation. As a result, they add flexibility to the production process: the products that are produced in the production phase upstream of a stock point can be transformed into several different products in the production phase downstream of the stock point. The reason for manufacturers of physical goods to hold intermediate stock, such as components and subassemblies, is that the production phase upstream of the stock point can be controlled separately from the production phase downstream of the stock point. This allows them to produce the intermediate products in large volumes, whereas the different end products can be produced in much smaller batches. In such a way, the risk of holding too much stock of products that are not required is significantly reduced, whereas production costs are still kept down because of economies-of-scale at the component level.
Supply chain management is a management philosophy that takes a systems approach to viewing the supply chain as a single entity (Mentzer et al., 2001). It seeks integration of the business processes that make up the system across organizational boundaries to increase the performance of the supply chain as a whole. The management techniques by which these business processes are integrated across the supply chain are referred to as coordination mechanisms.

Three modes of supply chain coordination are often distinguished (Lee, 2000; Simatupang et al., 2002; Simatupang et al., 2004; Piplani and Fu, 2005): information sharing, decision synchronization, and incentive alignment. Information sharing refers to the exchange of information and knowledge between supply chain members. A retailer sharing sales information with its distributors and manufacturers to prevent large fluctuations in production rates and inventory levels upstream in the supply chain is a well-known example (Lee, 2000). Decision synchronization is “the extent to which participating actors become involved in joint-decision making such as resolving conflicting objectives, mitigating uncertainty, redesigning workflow, and allocating resources” (Simatupang et al., 2004). An example is vendor-managed inventory, which refers to a supplier taking full responsibility for replenishing the stock held by its customer (Slack et al., 2007). Finally, the term incentive alignment is used to refer to the sharing of risks, costs, and rewards in order to give supply chain members the incentive to participate in supply chain integration (Lee, 2000; Piplani and Fu, 2005). Examples include volume-based quantity discounts, promotional allowances, and quantity flexibility contracts (Simatupang et al., 2002).

1.3 Enterprise software

At the highest level of abstraction, software can be divided into three broad categories (see Figure 1.2):

- Infrastructure software, which manages and controls the computer resources, such as the memory and processor. Examples include operating systems such as Microsoft Windows and Linux, database management systems such as Oracle and Microsoft SQL server, and the runtime environments of programming languages such as Java (Chaffey and Wood, 2005).
• Application software, which provides functionality to end users by building on top of the capabilities provided by the underlying infrastructure software. These end-user applications can be further divided into enterprise software, i.e. software sold to businesses, and consumer software, i.e. software sold to individual buyers (Xu and Brinkkemper, 2007).

• Embedded software, which is integrated to and bundled into equipment or appliances, such as televisions and microwaves (Messerschmitt and Szyperski, 2003).

![Diagram of software classification](image)

**Figure 1.2:** Classification of different types of software.

As discussed in Section 1.1, this thesis focuses mainly on enterprise software. Figure 1.2 shows that within the category of enterprise software, a distinction can be made between enterprise information systems, departmental information systems, and personal application software. Enterprise information systems provide support for key business processes across the organization, such as sales, finance, and logistics (Turban et al., 2005). Some well-known examples of such enterprise-wide systems include enterprise resource planning (ERP) systems, customer relationship management (CRM) systems, and human resource management (HRM) systems. The second category of enterprise software, the so-called departmental information systems, encompasses the specialized applications used within an organizational depart-
ment, such as computer-aided design (CAD) software, computer-aided manufacturing (CAM) software, and simulation software (Chaffey and Wood, 2005). The final category of enterprise software comprises personal application software, i.e. “applications that support general types of processing, rather than being linked to any specific business function” (Turban et al., 2005). Examples of software applications belonging to this latter category include spreadsheets, word processors, and web browsers.

1.4 Research objectives and approach

Supply chains of physical goods have been studied elaborately since the mid 1980s, which has resulted in a comprehensive framework for describing and analyzing these phenomena. The supply chain of enterprise software, however, has not been studied in the same rigorous way. In this thesis, we will investigate whether it is possible to apply some of the existing descriptive concepts from the analysis and design of the supply chain of physical goods to the supply chain of enterprise software or whether other constructs are needed for describing and analyzing these kind of networks. In particular, we pursue the following research objective:

**RO1** To develop a framework for the analysis and design of the supply chain of enterprise software.

The theoretical framework that we develop provides a “lens” for looking at the supply chain of enterprise software: it provides researchers with a sound basis for exploring how organizations in the software industry organize and manage their supply chains. For software companies, the framework may serve as a tool for systematically describing and analyzing the way in which they position themselves against their competitors (in terms of their product delivery strategy) and how they conduct business with their suppliers and customers.

In terms of Gregor’s (2006) taxonomy of different types of theory in information systems research, our framework for the analysis and design of the supply chain of enterprise software can be classified as a “theory for analysing”, which is particularly valuable when little is known about the phenomena that the framework intends to describe. Also, the framework may serve as a basis for the development of all other types of theory, including “theory for explaining and predicting” and “theory for design and action” (Gregor, 2006). This brings us to our second research objective:
RO2 To use the framework to (a) explore how the specific characteristics of the software industry enable and/or constrain the supply chain management initiatives of software companies, and (b) to develop quantitative models for assisting software companies in making appropriate design decisions.

The paradigm(s) that a researcher adopts to achieve his or her research objectives can be described by using the 3x3 “research matrix” proposed by Graighead and Meredith (2008). According to these authors, research activity can be classified along two dimensions: (i) the researcher’s framework ranging from rational (i.e. highly deductive, axiomatic) to existential (inductive, interpretive), and (ii) the source of the data ranging from natural (i.e. empirical, directly observed) to artificial (typically hypothetical reconstruction). Relative to the rational/existential dimension, Graighead and Meredith (2008) distinguish the following three categories:

- **axiomatic**, which represents the theorem-proof world of research as well as reasoning and logic models,
- **logical positivist/empiricist**, which represents research that is deductive and objective with the findings directly based on the data collected, and
- **interpretive**, which represents research that tends to be more inductive and subjective, the purpose being to understand how others construe, conceptualize, and understand events and concepts.

The natural/artificial dimension of Graighead and Meredith’s classification scheme also consists of three categories. These are:

- **direct observation of object reality**, as with field experiments,
- **people’s perceptions of object reality**, as with surveys and interviews, and
- **artificial reconstruction of object reality**, where based on the researcher’s own belief object reality is recast into another form that is more appropriate for experimentation and testing, as with analytical, descriptive, and physical models.

The 3x3 research matrix that results from combining the two dimensions is depicted in Table 1.1, with our research objectives placed in their appropriate cell(s). The development of a framework for the analysis and design of the supply chain of enterprise software can be seen as a form of conceptual modeling. RO1 is therefore positioned in the lower-right cell of the matrix (Meredith et al., 1989): based on one or more underlying case studies,
1.4. Research objectives and approach

A mental picture of the problem area is constructed, which is then described and evaluated according to the established theory of supply chain analysis and design. In doing so, we will gain insight in the indications and contraindications for the deployment of known descriptive supply chain theory to the new field of enterprise software. Whether existing concepts from the supply chain of physical goods can be successfully applied and adapted to the supply chain of enterprise software or whether other constructs are needed for describing and analyzing these kind of networks depends on the extent to which the resulting framework is “useful in aiding analysis is some way” (Gregor, 2006). This issue will be addressed to in the concluding chapter of this thesis.

To explore how the specific characteristics of the software industry enable and/or constrain the supply chain management initiatives of software companies, semi-structured interviews will be conducted at two companies in the domain of ERP software. RO2a has therefore been placed in the middle diagonal cell of Table 1.1. The positioning of RO2b is relatively straightforward (Bertrand and Fransoo, 2002): by giving reference to the theoretical framework for the analysis and design of the supply chain of enterprise software, a conceptual model of the decision problem is constructed, which is subsequently turned into a quantitative model by defining mathematical relationships between the variables that according to the conceptual model need to be included.

**Table 1.1: Overview of the different research paradigms that are adopted in this thesis.**

<table>
<thead>
<tr>
<th>Direct observation of object reality</th>
<th>People’s perceptions of object reality</th>
<th>Artificial reconstruction of object reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axiomatic</td>
<td>RO2b</td>
<td></td>
</tr>
<tr>
<td>Logical positivist/empiricist</td>
<td>RO2a</td>
<td></td>
</tr>
<tr>
<td>Interpretive</td>
<td>RO1</td>
<td></td>
</tr>
</tbody>
</table>
1.5 Outline of the thesis

This thesis is composed of several published and unpublished articles, so all chapters are readable as individual contributions, each related to a different aspect of the development, sales, and delivery of enterprise software: Chapters 2, 3, and 5 directly contribute to the achievement of our two research objectives; Chapters 4 and 6 contain additional material on related research topics. The content of each of these chapters will be briefly described in the paragraphs below. For an overview of how the different chapters can be positioned within the research matrix of Graighead and Meredith (2008), the reader is referred to Table 1.2.

In line with Section 1.2 on the supply chain of physical goods, our framework for the analysis and design of the supply chain of enterprise software comprises three elements:

1. the product delivery strategy of the software company,

2. the structure of the software company’s internal supply chain, and

3. the actors involved in the software company’s supply network and distribution channel.

The first element of the framework is discussed in Chapter 2, where we will identify two dimensions along which the product delivery strategies of software companies can be classified: (i) the position of the customer-order decoupling point, which determines a software company’s investment in software assets prior to receiving a customer order, and (ii) the degree of customer specification freedom, which indicates to what extent the customer is allowed to tailor these assets to his or her specific requirements. The typology that results from combining these two dimensions contributes to a better and systematic understanding of the different positions software companies can take on the continuum ranging from low flexibility and costs (standard software) and high flexibility and costs (bespoke software). The typological issues discussed in this chapter may also be helpful to support software vendors in selecting an appropriate product delivery strategy.

In the first part of Chapter 3, we will address the second element of our framework by providing concepts and tools for analyzing the structure of a software company’s internal supply chain. The premise of this chapter is that an analogy can be made between the role of inventory in the production of physical goods and the role of reusable software assets in the development
Table 1.2: Overview of how the different chapters can be positioned within the research matrix of Graighead and Meredith (2008).

<table>
<thead>
<tr>
<th>Direct observation of object reality</th>
<th>People’s perceptions of object reality</th>
<th>Artificial reconstruction of object reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axiomatic Chapter 3 (2nd part)</td>
<td>Chapter 4</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>Logical positivist/empiricist Chapter 5 (2nd part)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interpretive Chapter 2</td>
<td>Chapter 3 (1st part)</td>
<td>Chapter 5 (1st part)</td>
</tr>
</tbody>
</table>

of software. In particular, it is reasoned that systematic application of software reuse allows software companies to divide their development processes into two or more phases. The production of reusable software assets, often referred to as development for reuse, takes place in the first development phase, whereas concrete software systems are produced in the final development phase, also referred to as development with reuse. Intermediate phases exist when software reuse is applied recursively, i.e. when software systems are obtained by reusing and extending one or more existing software assets that have been developed with reuse as well. For manufacturers of physical goods, it holds that the number of phases in the production process is an important design decision. The same is true for the number of phases in the software development process: although software reuse has the potential of significantly reducing development time and costs, it does not always result in cost reduction because variations among related applications have to be accounted for. To assist software companies in determining the optimal number of software development phases, the second part of Chapter 3 provides a quantitative model for assessing whether it is economically feasible to apply software reuse recursively.

In Chapter 4, we shift our focus from cost minimization to profit maxi-
mization by comparing two strategies for the pricing of packaged software: fixed-fee and pay-per-use licensing. The mathematical model considers a monopoly software vendor that is selling packaged software to customers who are homogeneous in marginal value of software use but heterogeneous in level of use. In addition to obtaining the software package from the market, customers can develop the required software in-house. We start by determining the software vendor’s optimal licensing strategy for the case when in-house development is equally expensive for each customer. Subsequently, the assumption of a constant in-house development cost is relaxed by letting it vary among customers.

The third and last element of our framework for the analysis and design of the supply chain of enterprise software is discussed in the first part of Chapter 5, where the general supply chain concept is adapted to the specific characteristics of the software industry. In particular, we will argue that a component supplier can considered to be part of a software company’s supply network if the use of this component is covered by the license agreement between the software company and the end user, whereas a reseller, systems integrator, or consultancy firm can considered to be part of the software company’s distribution channel if it adds value by distributing the system, installing it at the customer’s site, and tailoring it to the customer’s specific requirements. In the second part of Chapter 5, we will use our conceptualization of the software supply chain to explore how software companies manage the relationships with their suppliers and customers by conducting two case studies in the ERP domain.

Over the past few years, software companies have been moving away from products that require local installation on the customer’s hardware towards remote, web-based services that can be accessed over the Internet (Cusumano, 2008). If the current trend towards software as a service (SaaS) continues to develop, significant changes in the industrial organization of the software industry can be expected as traditional channel intermediaries such as dealers and resellers are no longer required. Instead, proponents of the SaaS paradigm envision a supply chain structure in which service aggregators combine the services from different service providers to create composite services that are capable of performing higher-level business transactions. In order to assist potential service aggregators in deciding whether or not to enter the market, Chapter 6 develops a quantitative model for estimating and predicting whether the accumulated benefits from offering a composite service are sufficiently large to compensate for the total cost of developing
and operating the service.

Finally, the thesis concludes by summarizing our main findings in Chapter 7. Limitations and possible directions for future research are outlined as well.

1.6 Included publications

The chapters of this thesis correspond to the following papers:


Chapter 4 - D. Postmus, J. Wijngaard, and H. Wortmann, An economic model to compare the profitability of pay-per-use and fixed-fee licensing, Information and Software Technology 51 (2009) 581-588.

