

Agile Information Systems for Mastering Supply Chain Uncertainty

C.N. Verdouw^{1,2}, A.J.M. Beulens²,
T. Verwaart¹ and J. Wolfert^{1,2}

¹*LEI, part of Wageningen UR, LS The Hague*
²*Logistics, Decision and Information Sciences,
Wageningen University, Wageningen,
The Netherlands*

1. Introduction

Mastering both demand and supply uncertainty is a key challenge for many companies. Markets are increasingly turbulent and also the vulnerability of production and logistics processes is growing. The management of uncertainty has been addressed as an essential task of supply chain management (among others by Davis 1993 and van der Vorst & Beulens 2002).

For coping with the addressed uncertainties, Supply Chain Management (SCM) literature initially has focused on creating so-called lean supply chains that efficiently push products to the market. Lean supply chains build upon reduction of demand uncertainty, especially by product standardisation. Customers must choose from a fixed range of standard products that are made to forecast in high volumes. Business processes in lean supply chains can be highly automated by Enterprise Resource Planning (ERP) systems (Davenport and Brooks, 2004).

In the late 1990s, the then dominant approach of leanness was criticised more and more. It was argued that in volatile markets it is impossible to remove uncertainty. Companies therefore should accept differentiation and unpredictability, and focus on better uncertainty management. Agility was proposed as an alternative approach that aims for rapid response to unpredictable demand in a timely and cost-effective manner (Fisher, 1997, Christopher, 2000). It is founded on a mass customisation approach that combines the seemingly contradictory notions of flexible customisation with efficient standardisation (Davis, 1989, Pine *et al.*, 1993). Efficient standardisation is realised by fabricating parts of the product in volume as standard components; distinctiveness is realised through customer-specific assembly of modules (Duray *et al.*, 2000).

Until then, SCM focused on strategies for coping with demand uncertainty. Lee (2002) was one of the first who stressed the impact of supply uncertainty on supply chain design. Supply chains characterised by high supply uncertainty require the flexibility to deal with unexpected changes in the business processes. Disturbances of logistics, production or supply of materials should rapidly be observed and lead to process changes including re-planning and re-scheduling, purchasing new material, hiring alternative service providers, or negotiating new customer requirements.

Supply chains that are characterised by a high uncertainty of both demand and supply require a combination of responsiveness to changing demand and the flexibility to deal with unexpected changes in the business processes. Following Lee (2002), we use the term agility to characterise these types of supply chains. In agile supply chains, demand requirements and supply capabilities, *i.e.* products and processes including resources, should be continuously attuned (Verdouw et al., 2011). Therefore, both front-office and back-office systems need to be flexible and smoothly integrated (Verdouw et al., 2010a).

The main objectives of this chapter are to define the requirements to information systems in agile supply chains and to develop strategies for implementation of agile information systems. For the identification of requirements, the concept of mass customisation is applied to information systems.

The chapter first introduces a typology of supply chain strategies and the role of information systems in these strategies. Next, it focuses on information systems in the quadrant of agile supply chains. It is argued that these supply chains information systems should support an ICT (information and communications technology) mass customisation approach and the basic requirements for such an approach are defined. In the next section the role of ERP systems, configurators and Service Oriented Architecture (SOA) to enable ICT mass customisation is described. The chapter concludes with the introduction of three basic strategies for the implementation of agile information systems. The strategies involve different divisions of product configuration, process configuration and management of the order fulfilment among ERP systems, dedicated configurator software and SOA platforms.

2. Information systems and Supply Chain Management

2.1 Typology of supply chain strategies

Fisher (1997) introduced the idea that supply chain design should match the degree of demand uncertainty. Fisher discriminates between functional and innovative products. For functional products, having low demand uncertainty, efficient or lean supply chains perform best. For innovative products, that have a high degree of demand uncertainty, flexible or agile chains are a better match. Lee (2002) extends Fisher's analysis by adding the dimension of supply uncertainty. Lee distinguishes between stable and evolving supply processes. Stable processes are characterized by controllable production, mature technology and settled industry. In evolving supply processes production and technology are under development and more or less unpredictable. Lee matches four supply chain types with characteristics of supply and demand (see figure 1):

		Demand Uncertainty	
		Low (Functional Products)	High (Innovative Products)
Supply Uncertainty	Low (Stable Process)	Efficient supply chains	Responsive supply chains
	High (Evolving Process)	Risk-hedging supply chains	Agile supply chains

Fig. 1. Supply chain strategies and demand and supply characteristics [8] (Lee 2002)

- *Efficient* supply chains focus on cost reduction and match with low supply uncertainty - i.e. a controllable production process - and low demand uncertainty.
- *Risk-hedging* supply chains focus on pooling resources to reduce supply uncertainty; this type of chain matches with high supply uncertainty and low demand uncertainty.
- *Responsive* supply chains focus on flexibility through make-to-order process and mass customization; they match with low supply uncertainty and high demand uncertainty.
- *Agile* supply chains combine risk-hedging and responsive strategies, aiming to cope with both high supply uncertainty and high demand uncertainty.

The present chapter focuses on agile supply chains. A firm operating in such a supply chain lacks information about future demand and cannot reliably plan the order fulfillment process. After having defined the current position, two types of strategic options for dealing with the accompanying uncertainty can be distinguished: i) uncertainty reduction strategies that focus on decreasing the need for information, and ii) strategies for better management of uncertainties that focus on improving the information processing capacity.

Firstly, a firm should determine whether reduction of uncertainty is possible and desirable. Uncertainty reduction would imply a shift toward efficient, responsive, or risk-hedging supply chains in the framework of Figure 1. Reduction strategies aim to reduce differentiation by standardization and to eliminate the sources of disruptions. Demand-related examples are product standardization, sharing demand information exchange for improved forecasting and establishment of long-term contracts. Supply-related examples of reduction strategies are improved production control, sharing supply information for synchronized planning, cooperation with technology suppliers, hubs for supplier-managed inventory, and production standardization e.g. by fixed batch volumes, standard carriers, or fixed delivery schedules. Reduction of, especially, demand uncertainty - for instance by product standardization and reducing available product options - is not always desirable. In particular this is not desirable for firms that find their market niche in flexibly fulfilling specific customer needs.

Firms that cannot sufficiently reduce supply and demand uncertainties must find ways to manage the uncertainties. Such firms can consider possibilities for uncertainty management, which leave differentiation and unpredictability as is, but aim to manage it by better organization, maintaining close relationships with suppliers and service providers, usage of advanced decision support tools and better utilization of information.

The mentioned strategies show that information systems are important means for uncertainty reduction and uncertainty management. However, in particular agile supply chains entails specific information system needs, as discussed below.

2.2 Agile supply chains

In the 1990s Supply Chain Management (SCM) evolved towards an integrated process approach in which the concepts of logistics management were extended to incorporate the integration of firms in its supply chain. In the beginning, the focus in Supply Chain Management (SCM) was very much on so-called lean supply chains. The origins of lean manufacturing can be traced to the Toyota Production System (TPS), which focuses on the reduction and elimination of waste (Womack et al. 1991). Thus, lean supply chains focus on efficient streamlined pipelines that push raw material to the market in order to supply predictable demand in high volumes at the lowest costs.

During the 1990s the focus on supply chains as static physical pipelines was criticized more and more. In definitions from the Supply Chain Management (SCM) literature, the network character of supply chains was emphasized (among others by Christopher 1998): "A Supply Chain is the network of organizations that are involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services in the hands of the ultimate consumer."

The enrichment of the supply chain concept with the network dimension was no conclusive answer to the criticisms on supply chains as static physical pipelines. Also supply chain networks can be focused on the pushing products efficiently to the ultimate customers. As a consequence, in the beginning of this century there was an intensive debate in the SCM-field on agility as an alternative for the then dominant approach of leanness (Christopher 2000). It was argued that a fundamental shift was required in the dominant underlying approach. According to Christopher (2000), the lengthy and slow-moving "pipelines" have become unsustainable due to the turbulence and volatility of current marketplaces. He suggests that the key to survival in these changed conditions is through "agility", in particular by the creation of responsive supply chains that are market sensitive.

Agility can be defined as "using market knowledge and virtual corporation to exploit profitable opportunities in a volatile market place" (Naylor et al. 1999). Agile supply chains are required to be market sensitive and hence nimble (Christopher and Towill 2000). The primary purpose of responsive chains is to respond quickly to unpredictable demand in order to minimize stock outs, forced markdowns and obsolete inventory (Fisher 1997). This thinking is based on dynamics of business systems, which has been a major issue in management research for a long time, while the concept of agility can be traced back to (Goldman et al. 1995).

From the debate emerged that both leanness and agility are no mutually exclusive strategies. On a strategic level it is a matter of strategic choice (as argued yet by Fisher 1997). There is no one best chain network design ("one size fits all"), but companies continuously have to decide in which supply chain they want to participate, which role they are able to play the best and how they deliver added value in these networks. Furthermore, on an operational level it is always a balancing process between push and pull elements (Naylor et al. 1999; Mason-Jones et al. 2000). Nevertheless, there is a trend towards more agile supply chains because of increasing demand and supply uncertainty.

2.3 Role of information systems in Supply Chain Management

Supply chain management aims to manage the complex of business processes performed by numerous interdependent supply chain actors as an integrated whole. Information systems are vital to make the resulting complex, frequent and inter-enterprise information flows manageable by offering tools to automatically capture, process, transfer and communicate information in the supply chain. They can support the planning, control and coordination of supply chains in the following aspects:

- Communication of goals, plans and orders based on actual demand and supply information;
- Assurance of the required process execution by triggering the right activities and guiding the appropriate usage of resources and material (instructions);
- Continuous and chain-wide registration of monitoring information and effective alert mechanisms;

- Rapid and integrated decision-making based on aggregated and enriched monitoring information and information about external variables;
- Fast communication and implementation of the decided corrective and preventive actions.

As a consequence, one can distinguish between the following roles of information systems in Supply Chain Management (Verdouw et al., 2005b):

1. *Platform for shared communication*: to enable integration of control activities in supply chains, there should be in the first place an integrated technical information infrastructure. This requires an effective integration of the information systems of the individual chain actors, with respect to the information definitions, data exchange, applications and technical infrastructure. Examples of enabling ICT are:
 - a. Inter-organizational technical communication infrastructures, including the Internet and Virtual Private Networks;
 - b. EDI/XML based techniques for data exchange;
 - c. Enterprise Application Integration (EAI): software to integrate the applications of individual chain actors, nowadays based on service-oriented architecture (SOA);
 - d. Central, mostly web based information systems that are used by all involved supply chain actors to manage the basic information flow;
2. *Exchange of demand and supply information*: if there is a shared information infrastructure in place, demand and supply information can be communicated in the entire supply chain. ICT can help to capture this information, translate it to the involved chain actors and integrate it with the back office systems. Examples of enabling ICT are:
 - a. Product configuration tools that help to specify customer specific orders in interaction with the customer within the process constraints (guided selling) and convert generated customer orders automatically into detailed production, sourcing or distribution orders;
 - b. Point of Sales (POS) applications that help to replenish retail stocks on basis of actual consumer transactions;
 - c. Integrated Planning Systems (CPFR: Collaborative Forecasting Planning and Replenishment), in which the planning of the involved companies is aligned;
3. *Management of supply chain process execution*: the triggering, guiding and registration of customer specific task execution, including early detecting and signalling of (potential) disturbances. Examples of enabling ICT are:
 - a. Enterprise software for production management, distribution, warehouse management, sales, purchase and finance (ERP systems);
 - b. Early Warning Systems that continuously measure the process conditions and alert if there is serious risk based on an intelligent reaction on condition changes;
 - c. Inter-organisational Management Information Systems that translate the basis process information into high-level management information about the realization of Performance Indicators, often in the form of management cockpits or dashboards;
4. *Decision support*: tools to analyse demand and supply information and information about process fulfilment, determine the alternatives of corrective and preventive action and to compare and advice about the best solution. Examples of enabling ICT are:
 - a. Demand Forecasting Models that help to analyse consumer behaviour e.g. on basis of Point of Sales data and predict future consumer demand in order to improve planning;
 - b. Chain Process Simulation Models, that analyse the process behaviour in various levels of demand orientation and help to improve the fulfilment of consumer demand;

5. *Process configuration and implementation*: the adjustment of control variables and the supporting information systems in order to support customer-specific process execution. Vital enabling element is the ability to configure and reconfigure ICT rapidly. Furthermore, ICT can support the required changes in human behaviour, e.g. by stimulating problem awareness (diagnosis tools), vision development (gaming and simulation) and intervention design (Verdouw et al. 2005a).

A firm's information systems should match the type of supply chain it operates in. For further analysis we distinguish between front-office systems (coping with the demand side) and back-office systems (coping with the supply side). Front-office systems include order management, contract management, sales configurator, demand forecasting, and customer relations management systems. Back-office systems include resource planning and scheduling, stock management, purchasing, and supplier relations management systems. The type of supply chain determines the required flexibility of front- and back-office systems (Verdouw & Verwaart 2008):

- *Efficient supply chains* require stable, straight-forward planning systems for both front-office and back-office. The systems must be well-integrated to reduce waste of resources. Back-office systems support large volume production of standardized products based on long-run forecasts. Front-office systems support efficient order processing, long-run contracts and demand forecasts. Traditional ERP systems cover the demands of efficient supply chains.
- *Risk-hedging supply chains* require the same type of stable front-office systems as efficient supply chains do. However, they require flexible back-office systems, integrated with production control systems and supplier's systems. Disturbance of production or supply of materials should rapidly be observed and lead to re-planning and rescheduling. The rigid planning and scheduling systems of traditional ERP systems may cause problems in this type of supply chain.
- *Responsive supply chains* place high demands on the ability to combine fluctuations in demand and available supplies with respect to product specifications and lead times. The most common approach to organize responsiveness is mass customization in an assemble-to-order (ATO) production environment. This type of supply chain quickly responds to demand variability by efficient assembling of order-specific products from standard components. It requires stable back-office systems for efficient production of standardized components and rapid assembly. Traditional ERP systems can meet this demand. However, front-office systems require a flexibility usually not offered by traditional ERP systems. A responsive supply chain may require a more sophisticated sales configurator.
- *Agile supply chains* require flexibility in both front-office and back-office systems. They demand flexible ERP in the back-office and sophisticated configurator and customer communications systems in the front-office. Tight integration is required between front-office and back-office and with systems of both suppliers and customers.

3. Requirements for information systems in agile supply chains

This section focuses on the requirements for information systems in agile supply chains. Therefore, it applies the concept of mass customisation to information systems.

In agile supply chains, it must be possible to easily set-up, connect and disconnect information systems needed to achieve a specific value proposition. It must be possible to

design and instantiate new or adjusted supply chain configurations rapidly and at low costs. The main challenge in achieving this is to combine flexible customization with efficient standardization in the design and implementation of the logistics information systems introduced above. Mass customization is broadly advocated as a core approach to balance these seemingly contradictory notions (Davis 1989; Pine et al. 1993; Kotha 1995). It is a modular strategy that is intended to accomplish efficiency by reusing standardized components, while achieving distinctiveness through customer-specific assembly of modules (Lampel and Mintzberg, 1996, Duray et al., 2000). Mass-customisation builds on four operational capabilities: i) common building blocks that can be reused maximally, ii) unified architecture providing a structure of the defined components that constrain possible variants, iii) a technical platform for seamless integration of the building blocks, and iv) configuration tools that support the elicitation of customer requirements while considering the possible options (Pine et al. 1993, Duray et al. 2000, Zipkin 2000, Verdouw et al., 2010a, among others).

ICT mass customisation combines the seemingly contradictory notions of efficient standard software and flexible customised software (Verdouw et al., 2010b). It enables customer-specific assembly of information systems from a repository of standard components. As such, mass-customisable ICT can be positioned in the middle of a continuum of standard packaged software and customised software. Software developers pre-design and realise modules based on forecasted functionality. Customers get their own ICT configuration, but constrained by the range of available components, as defined in reference models for the configuration of systems. These components could be supplied by different software vendors, which allows for using best-of-breed solutions in selecting and designing systems. Following the identified requirements for mass customisation systems, the requirements for mass-customisable information systems are (Verdouw et al., 2010b):

- a. *Generic information model*: like product architectures in a mass customisation approach, information models should be set up as generic models, which define the class of architectures that can be assembled. Additional complexity of generic information models is that they comprise different interrelated model types including business process models, product models, semantic data models and ontologies, and information integration standards, e.g. eBusiness messages, web service standards, RFID protocols, and coding standards.
- b. *Modular software*: modules in an ICT mass customisation approach must be application-independent services, in which policy, input and output data, and interfaces are well defined (product modularity). They should not impose technical constraints on development of other modules (process modularity). Furthermore, it should be easy to replace a software module of provider A by a module of provider B, and it must be possible to combine modules of different vendors (network modularity).
- c. *Information integration platform*: a software platform is required that the modules can easily be plugged into, that can enact the execution of modules upon the occurrence of external or internal events, and that enables the exchange of information between the modules. Contrary to mass-customisable products, this platform has a virtual nature. It is not tied to one place. Especially internet-based techniques enable integration of modules that are located all over the world.
- d. *Configuration support*: configuration of ICT elicits the required functionality of specific instantiations of information systems building upon a generic information model. Since

information systems are composed of many interacting components, ICT configuration must be done for different levels of abstraction and different types of subsystems. Consequently, configuring information systems includes many partial configuration tasks that occur at different moments by different people. The dependencies between these different tasks must be well coordinated.

- e. *Component availability*: the availability of software modules that, together, provide the desired functionality, including a specification of the interfaces. A specific characteristic of ICT components is again the virtual nature. This implies that components can be duplicated very quickly and at a negligible cost. On the other hand, availability is dependent on service providers, because users have access to the modules via an often complex information infrastructure.

4. Implementation strategies for agile information systems

This section identifies three basic strategies for the implementation of agile information systems. The strategies involve different divisions of product configuration, process configuration and management of the order fulfilment among ERP systems, dedicated configurator software and SOA platforms. Therefore, we first will introduce the role of ERP systems, configurators and SOA to enable ICT mass customisation.

4.1 Enterprise software (ERP)

An Enterprise Resource Planning (ERP) system is a standardized software package that combines functionality of multiple business functions into one integrated system. It is based on a single database and contains functionality to support the main business processes including production, distribution, warehouse management, sales, purchase and finance. The major advantage of ERP is that it provides a stable backbone for the registration and communication of information among business functions, and consequently ensures the timely and accurate availability for integrated business process management. As such, it helps to overcome fragmentation between organizational units (functional silos) and systems (island automation).

ERP has emerged in the early 1990s as a logical extension of the material requirements planning (MRP) systems of the 1970s and of the manufacturing resource planning (MRP II) systems of the 1980s (Akkermans *et al.*, 2003, Jacobs and Weston, 2007). It has been advocated as essential means for implementation of Business Process Redesign in order to improve efficiency and customer service (Davenport, 2000, Hammer and Champy, 2001). Nowadays, ERP has become a de facto standard in many industries. For example, Aberdeen reported in 2008 that 86% of the manufacturing companies has implemented ERP (Aberdeen, 2008).

Early ERP-systems were not primarily focused on the supply chain (Davenport and Brooks, 2004). Consequently, they failed to meet the demands in current dynamic supply chains. In a critical note, Rettig (2007) argues that the ERP concept of a single monolithic system failed for many companies: "But these massive programs, with millions of lines of code, thousands of installation options and countless interrelated pieces, introduced new levels of complexity, often without eliminating the older systems they were designed to replace." In a study of Akkermans *et al.*, (2003) European supply chain executives address four key limitations of ERP systems in providing effective supply chain support:

1. Their insufficient extended enterprise functionality in crossing organizational boundaries;
2. Their inflexibility to ever-changing supply chain needs;
3. Their lack of functionality beyond managing transactions; and
4. Their closed and non-modular system architecture.

Akkermans *et al.* (2003) argue that the lack of modularity is the root cause for the other shortcomings.

In the research note "ERP is dead – long live ERP II", Gartner was one of the first who put the limitations of early ERP systems on the agenda (Bond *et al.*, 2000). They defined ERP II as a transformation of ERP into next-generation enterprise systems, which are web based, open and componentised. The ERP industry has embraced this new philosophy and started to modularize their systems architectures, in particular by incorporating Service Oriented Architecture (SOA) platforms, e.g. SAP NetWeaver (Møller, 2005). Furthermore, ERP vendors included intelligent modules that go beyond transactions (especially Advanced Planning Systems and Business Intelligence). However, the monolithic nature is deeply embedded in ERP systems. It takes much time to unravel the big jumble of software code into a consistent and coherent set of components. Consequently, the componentizing of ERP is still in progress. This implies that, although valuable advances are accomplished, the basic limitations of ERP systems still exist.

ERP systems perfectly cover the demands of efficient supply chains that are characterized by stable business processes and low demand uncertainty. However, in supply chains with uncertain demand and high vulnerability of production and logistics processes, current ERP is experienced as an obstacle in achieving the required flexibility (Akkermans *et al.*, 2003). The development towards modularized and service-oriented ERP is essential for the implementation of mass-customizable information systems. Such ERP systems ensure the availability of the software modules that, together, provide the desired functionality, including a specification of the interfaces. As such modularized ERP can provide a repository of building blocks that form the heart of mass customizable information systems.

4.2 Configurators

Configurators have emerged from the development of rule-based product design in the field of Artificial Intelligence. A well-known early application was R1, a product configurator for VAX computers (McDermott, 1981). A product configurator is a tool that guides users interactively through specification of customer-specific products (Sabin and Weigel, 1998, Forza and Salvador, 2002). Configurators generate specific product variants by combining sets of predefined components and specifying features according to permitted values. Next, they check the completeness and consistency of configured products based on rules that define the interdependencies between components or features. Product configurators are based on generic product models, which define the class of objects that can be configured (Hegge and Wortmann, 1991).

Currently, configurators play an important role in responsive supply chains, which are characterised by high demand uncertainty and low supply uncertainty (Lee, 2002). They are widely used for product configuration to enable rapid response to customer demands. In interaction with the user, the software generates consistent and complete specifications of customised products, taking into account both customer's requirements (e.g. functional specifications and delivery conditions) and feasibility of production, sourcing and delivery.

Along with the product specification, current configurators can produce commercial offers and draft contracts, and schedules and contracts for support and maintenance of the product. The software can be designed for use either by a sales representative of the supplier, or by a customer, e.g. through the internet. In both cases the configuration process results in a quick and effective order specification that can directly be entered into the production planning and scheduling systems.

Configurators can also be used to manage high uncertainties at the supply-side by supporting the rapid configuration of processes (Verdouw et al. 2010a). This concept of process configuration is introduced by Schierholt (2001), who applied the principles of product configuration to support process planning. Process configuration supports a rapid and consistent specification of the workflow that is needed to fulfil specific customer orders. For example, local deliveries from stock follow a different workflow than exports that are produced to order. Moreover, it supports reconfiguration of the workflow in case of unexpected supply events, e.g. components that were originally planned to be produced can be re-planned to be purchased.

Configurators can provide the configuration support as required in mass-customisable information systems. It helps to elicit the required functionality of specific instantiations of information systems building upon a generic information model.

4.3 Service-Oriented Architecture

Service-Oriented Architecture (SOA) is a software architecture where functionality is grouped around business processes and packaged as interoperable services. The aim is a loose coupling of services with operating systems, programming languages and other technologies, which underlie applications (Newcomer and Lomow, 2004). SOA separates functions into distinct units, or services (Bell, 2008), which are made accessible over a network to be combined and reused in the production of business applications (Erl, 2005). These services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services. Service providers publish web services in a service directory, service requestors search in this directory to find suitable services, bind to that service and use it, based on information from the directory and standardized procedures (Leymann, 2003; Erl, 2005). So, SOA provides the technology that enables timely and flexible sharing of information demands (Wolfert et al., 2010). It is component-based by nature and widely acknowledged as the de facto standard for information integration. SOA enables the definition of components with standardized interfaces, a central repository of published web services and standardized procedures for selection and implementation of components.

1. A technical architecture based on SOA consists of three layers (Erl, 2005):
2. A business process management layer, coordinating the execution of business services: this is a functional integration layer that groups services from the underlying business service layer into business processes. The process services are typically implemented through generic enactment engines, that execute workflows defined in languages like BPEL or BPML. Following the workflow specifications, the enactment engines invoke services in the next layer. Services in the process layer can be rapidly configured or reconfigured using business process management (BPM) tools.
3. A business services layer, delivering information services to the business processes. The business services implement the information processing functions of the actual business

processes. Business services may be either straightforward data registration or reporting services, or complex services based on extensive business logic. They may implement these functions directly, for instance applying the Business Rules Approach, or use application services that connect the business services to (legacy) information processing application systems.

4. A business application layer, executing the application logic and data storage. Applications are wrapped in application services, offering a standard web service interface to the business services, thus enabling enterprise application integration (EAI). The advances towards Service-Oriented Architecture (SOA) has been very important to enable mass-customisation of information systems. It can provide a software platform that the web services can easily be plugged into, that can enact the execution of web services upon the occurrence of external or internal events, and that enables the exchange of information between the modules. Consequently, it can support to meet, in particular, the requirements concerning software modularity and information integration platform (Verdouw et al., 2010b). As such, SOA can help to overcome the limitations of traditional ERP systems and achieve the required backend flexibility in agile supply chains. However, SOA does not include the knowledge required to specify services and to configure business processes as a sequence of services. Furthermore, the required software components must be available packaged as application-independent web services. So, even if a company applies SOA, important remaining challenges include the development of: i) generic information models that specify families of business processes and services, ii) tools that support configuration of specific business process and service architectures, iii) repository of software components software that are packaged as application-independent web services (Verdouw et al., 2010b, Wolfert et al., 2010).

In sum, it can be concluded that enterprise software (ERP), configurators and Service-oriented Architecture (SOA) together could meet the requirements of ICT mass customisation (see Table 1). ERP can ensure the availability of the software modules in a repository of building blocks that form the heart of mass customizable information systems. However, the development towards modularized and service-oriented ERP is a crucial prerequisite to achieve this. Furthermore, configurators can provide the configuration support as required in mass-customisable information systems. It helps to elicit the required functionality of specific instantiations of information systems building upon a generic information model. Last, Service-Oriented Architecture (SOA) can help to meet, in particular, the requirements concerning software modularity and it provides an information integration platform.

Requirements ICT mass customization (Verdouw et al. 2010b)	Enterprise software (ERP)	Configurators	Service-oriented Architecture (SOA)
a. Generic information models		X	
b. Modular software			X
c. Information integration platform			X
d. Configuration support		X	
e. Component availability	X		

Table 1. Main contributions of ERP, configurators and SOA to the requirements of ICT mass customization

The next section discusses some strategies on how the strengths of ERP, configurators and SOA can be combined to enable mass customisation of information systems.

4.4 Implementation strategies and challenges

Following Verdouw et al. (2010a), three basic strategies can be distinguished to implement ICT mass customization by combining the strengths of ERP, configurators and SOA. Each includes a different division of product configuration, process configuration and management of the order fulfilment among dedicated configurator software, ERP systems and service-oriented middleware.

In the *first strategy*, process models are defined, configured and executed in a SOA-based process management platform, which intermediates between front- and back office systems, in particular product configurators and ERP systems for planning and scheduling. At this option, the functionality for product and process configuration is provided by different applications. Process configuration is done outside product configurators in service-oriented middleware.

Implementation of such an approach is complex. To mention some difficulties: the constraints arising from the actual availability of the required resources should be taken into account, as well as the dependences between configuration choices; it must be possible to inherit configuration choices from the product requirement definition to detailed process diagrams; and configuration choices must be translated into graphical diagrams. SOA-based process management platforms do not yet provide tool support for the configuration of process models in a manageable and user-friendly way.

Another important challenge of the first strategy is to find solutions for some technical problems that will arise when implementing process configuration in a SOA-based platform. For example, solutions have to be found to solve the problems arising from the redundancy of process knowledge as the defined in the process models of SOA-platforms and the hard-coded process logic in legacy systems. Furthermore, the incorporation of process model configuration into the run-time system will impact systems performance, in particular if the reference models are used by multiple organisations. In the latter case, also security will be an issue.

The *second strategy* is the inclusion of process configuration in dedicated configurator software. For this option, both product and process configuration are incorporated within one configurator and this tool is integrated with external planning & scheduling systems, either directly or via service-oriented middleware. The main challenge of this option is to manage the intensive interactions between the configurator and with external planning & scheduling systems, in particular in case of frequent reconfiguration of the workflow due to unexpected events. The most natural solution direction is to integrate both types of systems via service-oriented middleware. The implementation issues are similar to the first discussed implementation, except that the tool support for process configuration is not included in the SOA-based platform but in an external tool. Consequently, additional challenges include how to translate the output of the process configurator into a process model notation that can be interpreted by the SOA-based platform.

The *third and last strategy* is to include both product and process configuration into the ERP system and thus integrate all features (product configuration, process configuration and planning and scheduling) within one system. In this case, the ERP system is also the front office for customer interaction. There is no need for defining process models in SOA standards like Business Process Modelling Notation (BPMN) and some of the technical

problems mentioned above can be solved easier. For example, the redundancy of process logic could be solved by using process models as the basis for system parameterisation. Furthermore, most ERP systems include functionality for product model definition and product configuration. This makes it easier to use process models for linking product configuration to the execution in back office systems. However, to do so, ERP systems should contain functionality for process modelling and the process models should be the basis for system usage. Many available ERP systems do not include such functionality. More importantly, ERP systems are not based on a modular approach, which is a fundamental precondition for the usage of process models to guide the workflow planning and execution in run-time information systems. At the same time, for many companies it is no realistic option to replace current systems with new flexible solutions is for many companies, among others because of the significant investments in legacy and the risks of losing stability. Because of these limitations, it might be preferably to keep the configuration of process models out of the ERP system.

5. Conclusions

The main objectives of this chapter were to define the requirements to information systems in agile supply chains and to develop strategies for implementation of agile information systems.

The chapter has first introduced a typology of supply chain strategies and the role of information systems in these strategies. The type of supply chain determines the required flexibility of front- and back-office systems. Efficient supply chains require stable, straight-forward planning systems for both front-office and back-office. Risk-hedging supply chains require the same type of stable front-office systems as efficient supply chains do. However, they require flexible back-office systems, integrated with production control systems and supplier's systems. Responsive supply chains place high demands on the ability to combine fluctuations in demand and available supplies with respect to product specifications and lead times. Agile supply chains require flexibility in both front-office and back-office systems. They demand flexible ERP in the back-office and sophisticated configurator and customer communications systems in the front-office.

Next, the chapter has focused on information systems in the quadrant of agile supply chains. It is argued that these supply chains information systems should support an ICT mass customisation approach. ICT mass customisation combines the seemingly contradictory notions of efficient standard software and flexible customised software. It enables customer-specific assembly of information systems from a repository of standard components. Five requirements for the enhancement of ICT mass customisation have been defined: a) generic information model, b) modular software, c) information integration platform, d) configuration support, and e) component availability.

In the next section the role of ERP systems, configurators and SOA to enable ICT mass customisation is defined. None of these technologies completely satisfy the defined requirements, but together they could enable ICT mass customisation. ERP can ensure availability of the software modules in a repository of building blocks that form the heart of mass customizable information systems. However, the development towards modularized and service-oriented ERP is a crucial prerequisite to achieve this. Furthermore, configurators can provide the configuration support as required in mass-customisable information systems. It helps to elicit the required functionality of specific instantiations of information

systems building upon a generic information model. Last, Service-Oriented Architecture (SOA) can help to meet, in particular, the requirements concerning software modularity and it provides an information integration platform.

The chapter concludes with the introduction of three basic strategies for the implementation of agile information systems. The strategies involve different divisions of product configuration, process configuration and management of the order fulfilment among ERP systems, dedicated configurator software and SOA platforms. All of the strategies entail order-specific configuration of the process model. The strategies differ in the technology to be used for that purpose and the location of the knowledge required for process configuration. The first strategy is to implement intelligent process configuration in the middleware that mediates between front-office systems (sales and product configurators) and back-office (ERP) systems. This approach would require a considerable advancement of the state-of-the-art in service composition. The second strategy is to include process configuration in the product configurator, thus enabling simultaneous configuration of product and process. This second approach would require the extension of current product configurators. Since process configuration depends on current and expected state of the back-office, it would also entail extensive and frequent information exchange between front-office and back-office systems. The third strategy is to implement order specific process configuration in the back-office ERP system. The latter strategy would avoid redundancy of process knowledge, but many current ERP systems do not support the modular process modelling approach and the dynamic configuration support required to realise this strategy. The first or the second strategy might be preferred depending on (among other factors) the extent of supply uncertainty in a particular branch of industry.

6. References

- Aberdeen, 2008. *The 2008 ERP in Manufacturing Benchmark Report*.
- Akkermans, H.A., Bogerd, P., Yucesan, E. & Van Wassenhove, L.N., 2003. The impact of ERP on supply chain management: Exploratory findings from a European Delphi study. *European Journal of Operational Research*, 146, 284.
- Bell, M., 2008. *Service-Oriented Modeling (SOA): Service Analysis, Design and Architecture*. Wiley and Sons, Hoboken, NJ.
- Bond, B., Genovese, Y., Miklovic, D., Wood, N., Zrinsek, B. & Rayner, N., 2000. *ERP Is Dead – Long Live ERP II*. New York, NY.
- Christopher, M., 1998. *Logistics and Supply Chain Management: Creating Value-Adding Networks*. London, Pitman Publishing.
- Christopher, M., 2000. The Agile Supply Chain: Competing in Volatile Markets. *Industrial Marketing Management*, 29 (1), 37-44.
- Christopher, M. and D. R. Towill, 2000. Supply chain migration from lean and functional to agile and customised. *Supply Chain Management*, 5(4): 206-214.
- Davenport, T.H., 2000. *Mission critical: realizing the promise of enterprise systems* Boston, MA: Harvard Business School Press.
- Davenport, T.H. & Brooks, J.D., 2004. Enterprise systems and the supply chain. *Journal of Enterprise Information Management*, 17, 8-19.
- Davis, S., 1989. From future perfect: Mass customizing. *Planning Review*, 17 (2), 16-21.
- Davis, T., 1993. Effective Supply Chain Management. *Sloan Management Review*, 34 (4), 35-46.

- Duray, R., Ward, P.T., Milligan, G.W. and Berry, W.L., 2000. Approaches to mass customization: configurations and empirical validation. *Journal of Operations Management*, 18 (6), 605-625.
- Erl, T., 2005. *Service-Oriented Architecture: Concepts, Technology, and Design*. Englewood Cliffs: Prentice Hall PTR.
- Fisher, M.L., 1997. What is the right supply chain for your product? *Harvard Business Review*, 75 (2), 105-116.
- Forza, C. and Salvador, F., 2002. Product configuration and inter-firm co-ordination: an innovative solution from a small manufacturing enterprise. *Computers in Industry*, 49 (1), 37-46.
- Goldman, S. L., Nagel, R. N., Preiss, K. (1995). *Agile competitors and virtual organizations: strategies for enriching the customer*. New York [etc.], Van Nostrand Reinhold.
- Hammer, M. & Champy, J., 2001. *Reengineering the Corporation* New York: HarperCollins.
- Hegge, H.M.H. and Wortmann, J.C., 1991. Generic bill-of-material: a new product model. *International Journal of Production Economics*, 23 (1-3), 117-128.
- Jacobs, F.R. & Weston, F.C., 2007. Enterprise resource planning (ERP)--A brief history. *Journal of Operations Management*, 25, 357-363.
- Kotha, S., 1995. Mass Customization: Implementing the Emerging Paradigm for Competitive Advantage. *Strategic Management Journal*, 16, 1995.
- Lampel, J. and Mintzberg, H., 1996. Customizing customization. *Sloan Management Review*, 38 (1), 21-30.
- Lee, H.L., 2002. Aligning supply chain strategies with product uncertainties. *California Management Review*, 44 (3), 105-119.
- Leymann, F., Year. *Web Services: Distributed Applications without Limits: An Outlined*. BTW, Leipzig, 2-23.
- Mason-Jones, R., Naylor, J. B., Towill, D.R. (2000). Engineering the leagile supply chain. *International Journal of Agile Management Systems*, 2(1), 54-61.
- Mcdermott, J., 1981. R1: The Formative Years. *AI Magazine*, 2 (2), 21-29.
- Møller, C., 2005. ERP II: a conceptual framework for next-generation enterprise systems? *Journal of Enterprise Information Management Executive*, 18, 483-497.
- Naylor, B., Naim, M.M. and Berry, D., 1999. Leagility: interfacing the lean and agile manufacturing paradigm in the total supply chain. *International Journal of Production Economics*, 62 107-118.
- Newcomer, E. & Lomow, G., 2004. *Understanding SOA with Web Services*: Addison Wesley.
- Pine, B.J., Victor, B. & Boynton, A.C., 1993. Making Mass Customization Work. *Harvard Business Review*, 71, 108-119
- Rettig, C., 2007. The Trouble With Enterprise Software. *MIT Sloan Management Review*, 49, 1, 21-+
- Sabin, D. and Weigel, R., 1998. Product configuration frameworks - a survey. *IEEE Intelligent Systems*, 13 (4), 42-49.
- Schierholt, K., 2001. Process configuration: Combining the principles of product configuration and process planning. *AI-EDAM*, 15 (5), 411-424.
- van der Vorst, J.G.A.J. and Beulens, A.J.M., 2002. Identifying sources of uncertainty to generate supply chain redesign strategies. *International Journal of Physical Distribution & Logistics Management*, 32 (6), 409-430

- Verdouw, C.N.; Beulens, A.J.M.; Trienekens, J.H.; Vorst, J.G.A.J. van der, 2005. Leagile ICT in demand driven chains: a reference model approach. *2nd European forum on Market-driven Supply Chains, 5-6 April 2005, Milan, Italy, EIASM*.
- Verdouw, C.N.; Wolfert, J.; Beulens, A.J.M. , 2005b. Enhancing transformational and incremental innovation with ICT. *5th EFITA conference, , 25-28 July 2005, Vila Real, Portugal*.
- Verdouw, C. & Verwaart, T., 2008. Mastering Demand and Supply Uncertainty with Configurator Software. In P. Cunningham & M. Cunningham (eds.) *Collaboration and the Knowledge Economy: Issues, Applications, Case Studies*. Amsterdam: IOS Press, 1009-1016.
- Verdouw, C.N., Beulens, A.J.M., Trienekens, J.H. & Verwaart, T., 2010a. Mastering demand and supply uncertainty with combined product and process configuration. *International Journal of Computer Integrated Manufacturing*, 23, 515-528.
- Verdouw, C.N., Beulens, A.J.M., Trienekens, J.H. & Verwaart, T., 2010b. Towards dynamic reference information models: Readiness for ICT mass customization. *Computers in Industry*, 61, 833-844.
- Verdouw, C.N., Beulens, A.J.M., Trienekens, J.H. & Van der Vorst, J.G.A.J., 2011. A framework for modelling business processes in demand-driven supply chains. *Production Planning & Control*, 22 (4), 365-388.
- Wolfert, J., Verdouw, C.N., Verloop, C.M. & Beulens, A.J.M., 2010. Organizing information integration in agri-food--A method based on a service-oriented architecture and living lab approach. *Computers and Electronics in Agriculture*, 70, 389-405.
- Womack, J.P., Jones, D.T., Roos, D., 1991. *The Machine That Changed the World*. Perennial, HarperCollins.
- Zipkin, P., 2001. The Limits of Mass Customization. *MIT Sloan Management Review*, 42 (3), 81-86.