Available online at www.sciencedirect.com

SCIENCE DIRECT

Computers in Industry 53 (2004) 165–178

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

Software component architecture in supply chain management

Martin Verwijmeren*

MP Objects, P.O. Box 29126, 3001 GC Rotterdam, The Netherlands

Received 17 January 2003; accepted 23 July 2003

Abstract

We present a software component architecture for supply chain management across dynamic organisational networks. The local management in the architecture is done by existing enterprise resource planning (ERP) systems, warehouse management systems (WMS) and transportation management systems (TMS). The integral management in the architecture is executed by supply chain engines (SCEs). These software components make up a layer of systems for supply chain management running on top of ERP, WMS and TMS. The SCEs provide both the intelligence and the flexibility as required for the integral management of dynamic supply chains. The software component architecture can be implemented with technology for components, interfaces and services as available in Java Enterprise, CORBA and Web Services.

Keywords: Supply chain management; Networked organisations; Inventory management; Distribution management; System architecture; Software components; Distributed systems; Object-oriented systems; Supply chain engines; ERP; WMS; TMS; Java; CORBA; Web Services

1. Introduction

Increased global competition and demand for greater product variety, result in the emergence of dynamic supply chains, which need to be co-ordinated without sacrificing their flexibility. Co-ordination across dynamic supply chains requires inter-organisational information systems. However, rigid automation of supply chain co-ordination takes away the flexibility of the networked organisations.

Traditional ERP systems and EDI interfaces are not geared to supply chain management across dynamic organisation networks [1]. Enterprise resource planning (ERP) systems provide integration for supply chain management. However, due to their central server and procedural software, the ERP systems lack the autonomy and flexibility required by networked organisations. Local systems linked by electronic data interchange (EDI) support the flexibility for networked organisations. However, EDI interfaces just focus on data exchange, and therefore miss the decision rules required for supply chain management.

Given the drawbacks of the traditional ERP systems and EDI interfaces, we are in search for a new software architecture. The main question of our research is: what is a suitable software architecture for supply chain management across dynamic organisational networks? Our study has resulted in a software architecture based on components, meant to support the required co-ordination and flexibility simultaneously.

First, we will outline dynamic supply chains, including supply chain management and networked organisations. We then position existing systems (ERP, WMS and TMS) and reveal complementary components for

* Tel.: +31-10-2900304; fax: +31-10-2900305.
E-mail address: martin.verwijmeren@mp-objects.com (M. Verwijmeren).

0166-3615/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2003.07.004

dynamic supply chain management. The software component architecture for supply chain management will be illustrated for one practical area. The inventory management architecture explains the design of inventory management engines and their use in supply chains for consumer electronics.

Furthermore, we have a look at available technology (Java Enterprise, CORBA and Web Services) for the implementation of our software component architecture. We will conclude with the major findings of the study.

2. Dynamic supply chains

2.1. Supply chain management

In business we face the trend of increased global competition, which forces companies to improve their efficiency. One of the measures for efficiency improvement is supply chain management. Supply chain management focuses on the inter-organisational management of goods flows between independent organisations in supply chains, such as raw material winners, component manufacturers, finished product manufacturers, wholesalers and retailers. This integrative approach to planning, control and monitoring of product flows, from suppliers to end users, aims at improved customer service at reduced overall costs [2,3].

Supply chain management starts when the scope of integration is extended from internal to external co-ordination, a stage that takes into account customers and suppliers in the supply chains [4,5]. The core principle behind supply chain management is the reduction of uncertainty in decision making processes of organisations in supply chains. The co-ordination of the management processes in a supply chain requires information exchange between the organisations in the supply chain. The extra availability of information in decision making units reduces uncertainty, resulting in better control and finally in improved performance [6]. For the exchange of proprietary information, such as sales and forecasts, a collaborative attitude of the supply chain partners is needed [7].

In most cases, supply chain management has been realised by one organisation dominating the other organisations in the supply chain. As presented in

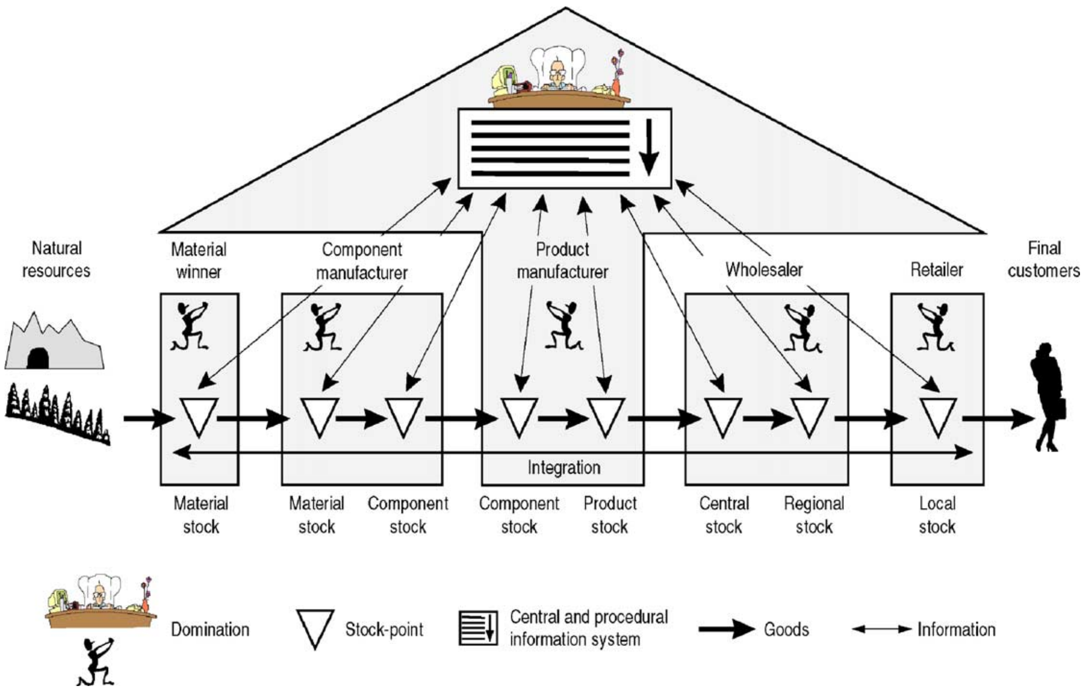

Fig. 1. Supply chain management by domination of one organisation over others.

Fig. 1, a dominant organisation can make use of hierarchical control to achieve integral logistics management across organisational boundaries. The subordinate organisations in the supply chain are forced to obey to the instructions of the dominant organisation. The integral management enforced by one dominant organisation is usually accomplished with one central information system with fixed procedures, ruling the underlying processes of the subordinate organisations in the supply chain.

This type of supply chain management, where one organisation dominates the others, could improve productivity of the supply chain, including lower costs and better quality (customer service). This results in a better price-quality ratio of products, as required by demanding customers. However, supply chain management imposed by one dominant organisation, does not contribute to the flexibility of the supply chain, which is needed to offer customers a more dynamic product assortment. Because subordinate organisations are bound to one dominant organisation, they can not easily change their portfolio in the supply chain or switch to other supply chains.

2.2. Networked organisations

Another business trend is the demand for greater product variety, requiring companies to become more flexible. A means to increase flexibility are networked organisations, which can continuously reform to accommodate changing market demand. Typical characteristics of networked organisations are autonomous control, common goals, mutual trust, information exchange, close co-operation and variable coupling [8–13]. Organisation types similar to networked organisations are the value-adding partnership [14], the virtual corporation [15], the lean enterprise [16] and the extended enterprise [17].

Networked organisations apply a co-ordination mechanism which is an intermediate between market co-ordination and hierarchical co-ordination. Hierarchies and markets support productivity and flexibility, respectively, but can hardly support them simultaneously. In contrast to an open market, there is some joint commitment among networked organisations to establish and cultivate relationships [13]. Whereas in a closed hierarchy there is unity of ownership, power and loyalty, in a network of networked organisations there is no single trinity, but distributed ownership, power and loyalty instead [18]. Networked organisations are able to couple and uncouple other organisations with less cost and time than hierarchies [11]. Hierarchies are bound to an installed base of dedicated resources that can not be adapted immediately. It is easier to start or end co-operation with a partner than it is to add or remove an organisational unit [18].

To obtain the required flexibility, supply chains could introduce supply chain management by co-operation across a network of organisations. As is shown in Fig. 2, networked organisations apply lateral control to achieve supply chain management beyond their organisational boundaries. When compared to supply chain management by domination of one organisation, co-operation across networked organisations

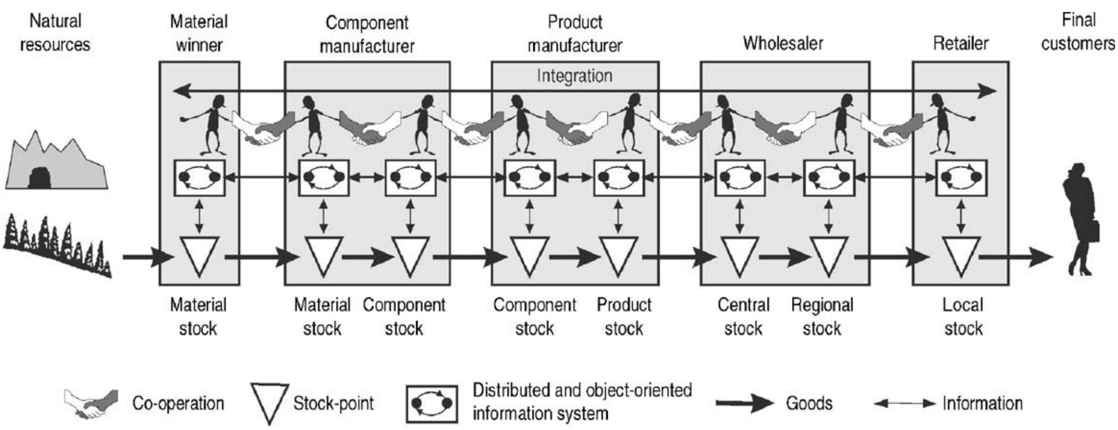

Fig. 2. Supply chain management by co-operation across networked organisations.

can also increase the flexibility of the supply chains. Networked organisations can frequently change their positions in supply chains to respond to changing market circumstances. In the context of networked organisations, new supply chains emerge to create promising products, while supply chains for outdated products disappear. The changing relationships of networked organisations result in dynamic supply chains.

Traditional ERP systems and EDI interfaces are not equipped for supply chain management across networked organisations. Enterprise Resource Planning systems focus on the integral management of processes within a company [19,20]. Because of its central architecture, an ERP system assumes one central organisation, but in dynamic networks there is no central point of authority [21]. Moreover, the procedural ERP software can not easily support the coupling and decoupling of organisations to the dynamic network. ERP systems have intelligence for co-ordination, but miss the flexibility needed for networked organisations. Traditional company systems can mutually be linked through electronic data interchange (EDI) [22–24]. The systems and interfaces can resemble the network structure, but do not provide additional functions for logistics control across organisations. They can align with the structure of networked organisations, but lack the intelligence for co-ordination across the supply chain.

3. Supply chain management architecture

3.1. ERP, WMS and TMS

The fundament of a system architecture for supply chain management is constituted by ERP systems, WMS and TMS (see Fig. 3). These information systems can be either standard software packages with parameter configuration or software programs tailor made to company specific needs. The basic systems in a supply chain provide specific functions for typical users:

- ERP, enterprise resource planning systems:
  - functions: purchase, materials management and sales;
  - users: manufacturers and trading companies.
- WMS, warehouse management systems:
  - functions: receipts put-away, bin management and order picking;
  - users: logistics service providers and wholesalers.
- TMS, transportation management systems:
  - functions: transport booking, planning and monitoring;
  - users: forwarders and carriers.

A system like ERP, TMS and WMS has its strength in the consistent management of elementary business

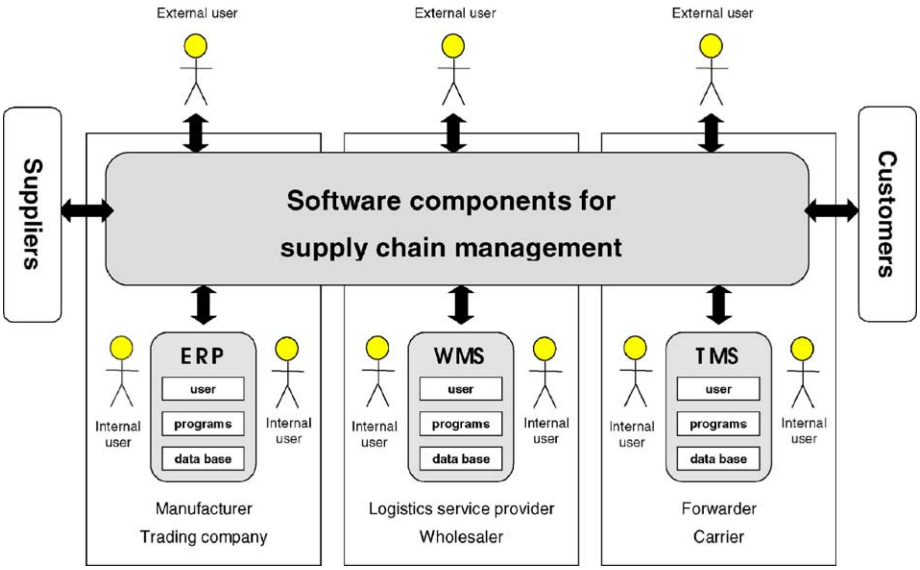

Fig. 3. ERP, WMS and TMS in the supply chain management architecture.

data, such as: customers and sales orders, items and prices, warehouses and bins, resources and work orders, suppliers and purchase orders.

Each of the systems has its own database in which these data are stored. The software programs in an ERP system, WMS or TMS provide functions for the transformation of the stored data during the course of the business process. The users can access the programs they need for their tasks through a user interface.

The ERP, WMS and TMS focus on co-ordination of the processes within the organisation. Their databases store the internal data, the programs provide intelligence for internal co-ordination and the user interfaces give access to internal users. So, these traditional systems are fit for internal logistics management, but they miss the co-ordination capabilities required for the management of dynamic supply chains. Possible EDI interfaces between the local systems enable the exchange of information, but they do not add the required intelligence for supply chain management.

3.2. Software components

For supply chain management across dynamic organisation network, the basic ERP, WMS and TMS (including EDI interfaces) are not enough. Additional systems are needed in the supply chain management architecture to support the co-ordination of logistics processes which are distributed over different organisations. Therefore, in the supply chain management architecture (see Fig. 4), we introduce software components on top of ERP, WMS and TMS to provide extra intelligence for co-ordination as well as greater flexibility to cope with dynamics. The required intelligence is built in the software components, whereas the component structure enables the required flexibility.

The software components in the architecture are called supply chain engines (SCEs), to express the power they provide for supply chain management as an extension to the local ERP, WMS and TMS. The supply chain engines can run on the computers of the different organisations in the supply chain. Together, the software components make up a system layer dedicated to supply chain management. Whereas the ERP, WMS and TMS focus on internal management, the SCEs add functions and data for external management to the supply chain architecture. The software components in the architecture can be classified in three categories, differentiated to the level of supply chain management supported by the engines:

- Communication engines:
  - function: basic communication between the systems (and users) in the supply chain;
  - examples: data communication, message conversion and flow control engines.
- Information engines:
  - function: transparent information over the systems (and users) in the supply chain;
  - examples: stock visibility, track and trace and report query engines.
- Management engines:
  - function: advanced management across the systems (and users) in the supply chain;
  - examples: inventory management, production management and distribution management engines.

In the communication layer, the message communication engines, for example, support the exchange of messages between systems using either SMTP (e-mail), FTP (file transfer) of HTTP (web protocol). A message communication engine is installed on each of computers to be connected. Then, the local systems can exchange messages through the local engines, which pack, send, transfer, receive and unpack the messages.

At the information level, the stock visibility engines, for example, can present stock data from different local systems. External users can specify their information request via a Web browser. The engines retrieve the stock levels from the local systems and integrate the information in a stock overview. This overview can be displayed to an external user or can be imported in another system in the supply chain.

The management layer includes engines for advanced management of the supply chain. They have intelligent rules for fully automatic decision making and semi-automatic decision support. For example, distribution management engines can be used for the integral management of distribution services which are provided by independent organisations in a physical distribution network. Below, we will explain the design and application of inventory management engines in the software component architecture for supply chain management.

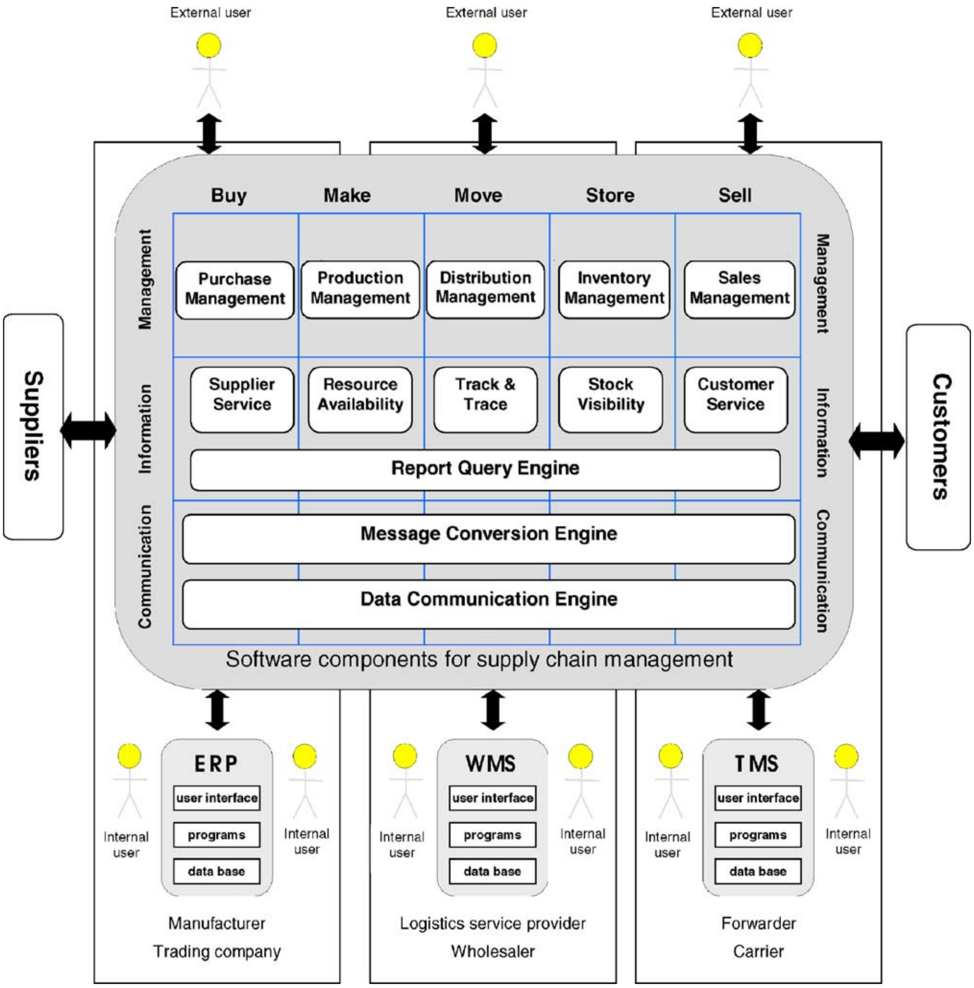

Fig. 4. Software component architecture in supply chain management.

4. Inventory management architecture

4.1. Inventory management engines

Inventory management engines enable integral inventory management across networked organisations (see Fig. 5). These software components together with existing systems in the supply chain represent an inventory management architecture. An inventory management engine (IME) is an extremely elementary system, managing the stock level of just one single SKU stock-point. The software components are loosely coupled to one another in networks, in order to support integral inventory management in dynamic supply chains [25]. The local management systems (ERP, TMS or WMS) can be used by the IMEs to get and set stock information. The IMEs can be distributed over several locations and organisations in the supply chains.

The inventory management engines support integral inventory management according to base stock control (BSC), material/distribution requirements planning (MRP/DRP) and line requirements planning (LRP) [26–30]. With the help of the system variables and system equations in a network of IMEs, integral inventory management can be supported according to one of these algorithms. The software components integrate stock levels in the time dimension as well as in the place dimension, by using extra information on time-phased demand (order plans) and integral inventory (local plus downstream stock).

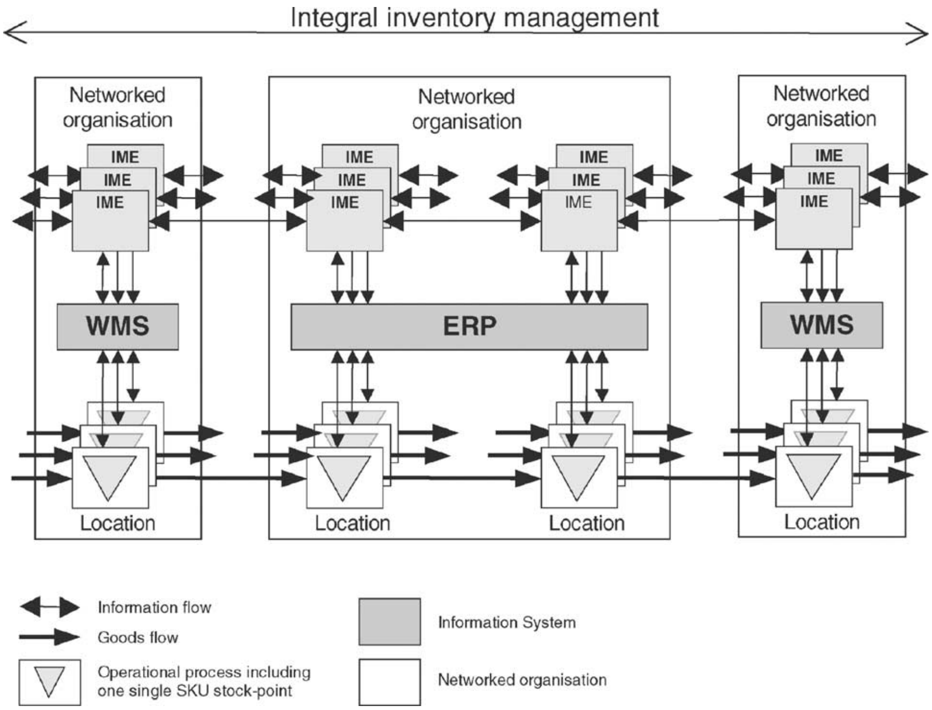

Fig. 5. Inventory management architecture.

For the support of networked organisations some supplementary system variables and equations are included in the IMEs. The software components support configuration flexibility, timing flexibility and algorithm flexibility. Configuration flexibility is the ability to couple and decouple of IMEs in a dynamic network. Timing flexibility refers to coping with different review moments and plan moments. Algorithm flexibility includes algorithm selection in an IME and algorithm transition across the engines. These properties allow autonomy of networked organisations with respect to the place, the timing and the type of management.

The software components for integral inventory management across networked organisations have an object-oriented design (see Fig. 6). An information system for supply chain management can be built of objects that represent real-world entities in the supply chain [31,32]. Forthcoming objects are durable, and have data and functions that naturally belong together. An inventory management engine consists of five object classes, each responsible for the functions and data related to an entity type in the system environment: customers, suppliers, operational process, inventory (stock-point) and strategist (supervisor) [33].

The object classes in an IME are equipped with attributes and operations which enable them to perform in accordance with their targets [34,35]. The operations of an object class give the ability to provide its functions, while the attributes give the ability to maintain its data. The objects work together by sending messages, representing a request for a service and the response to that service request. Customers and suppliers in the environment can again be equipped with IMEs, but they may also use other information systems or no system at all. If a customer or a supplier also makes use of an IME, the same system associations apply from the viewpoint of the customer or the supplier.

4.2. Supply chains for cordless phones

The inventory management architecture has been tested against a network of supply chains for manufacturing and distribution of cordless phones [36,37].

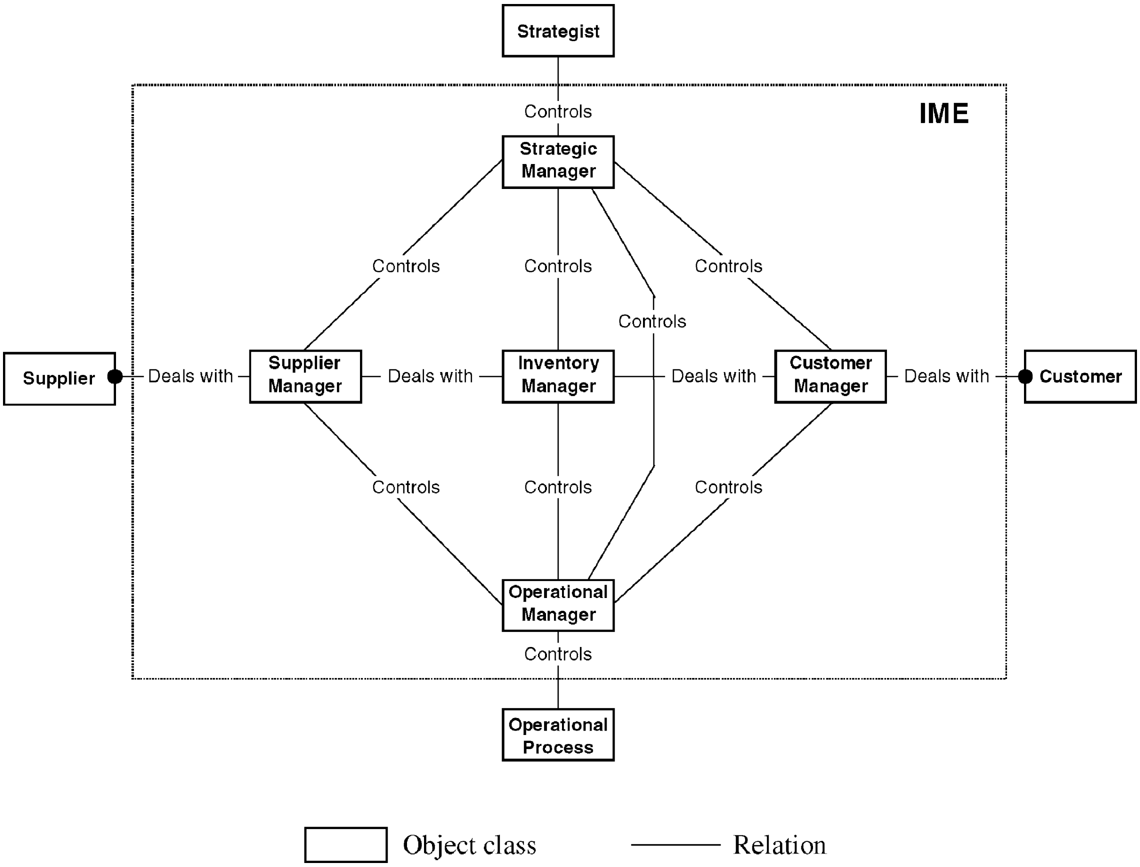

Fig. 6. Object classes in the inventory management engine.

A dozen of manufacturers in various countries supply a telecommunications retailer with cordless telephones (see Fig. 7). The retailer sells the products through more than hundred outlets to customers in the consumer and business market. Each of the organisations has its own information systems in place for local management. Due to market and technology changes, the network of organisations in the supply chain is expanding and changing more frequently. At the same time, the supply chains need to be managed in a more integral way to remain competitive.

In the supply chain for cordless telephones there is the need for integral inventory management across the networked organisations. However, it is practically unfeasible to develop and maintain the networked inventory management with the existing information systems in the supply chain. The heterogeneity and instability of the local systems would require a prohibitive number of dedicated interfaces. Networked inventory management can be achieved by application of IMEs for every single SKU stock-point. Loose coupling enables the IMEs to combine integration and flexibility. The software components form a system layer for supply chain management, abstracted from the existing systems for local management.

5. Implementation technology

5.1. Components, interfaces and services

Implementation technology is needed for the development and deployment of the software components in the supply chain management architecture. The technology used for the implementation of the supply chain engines includes: application components,

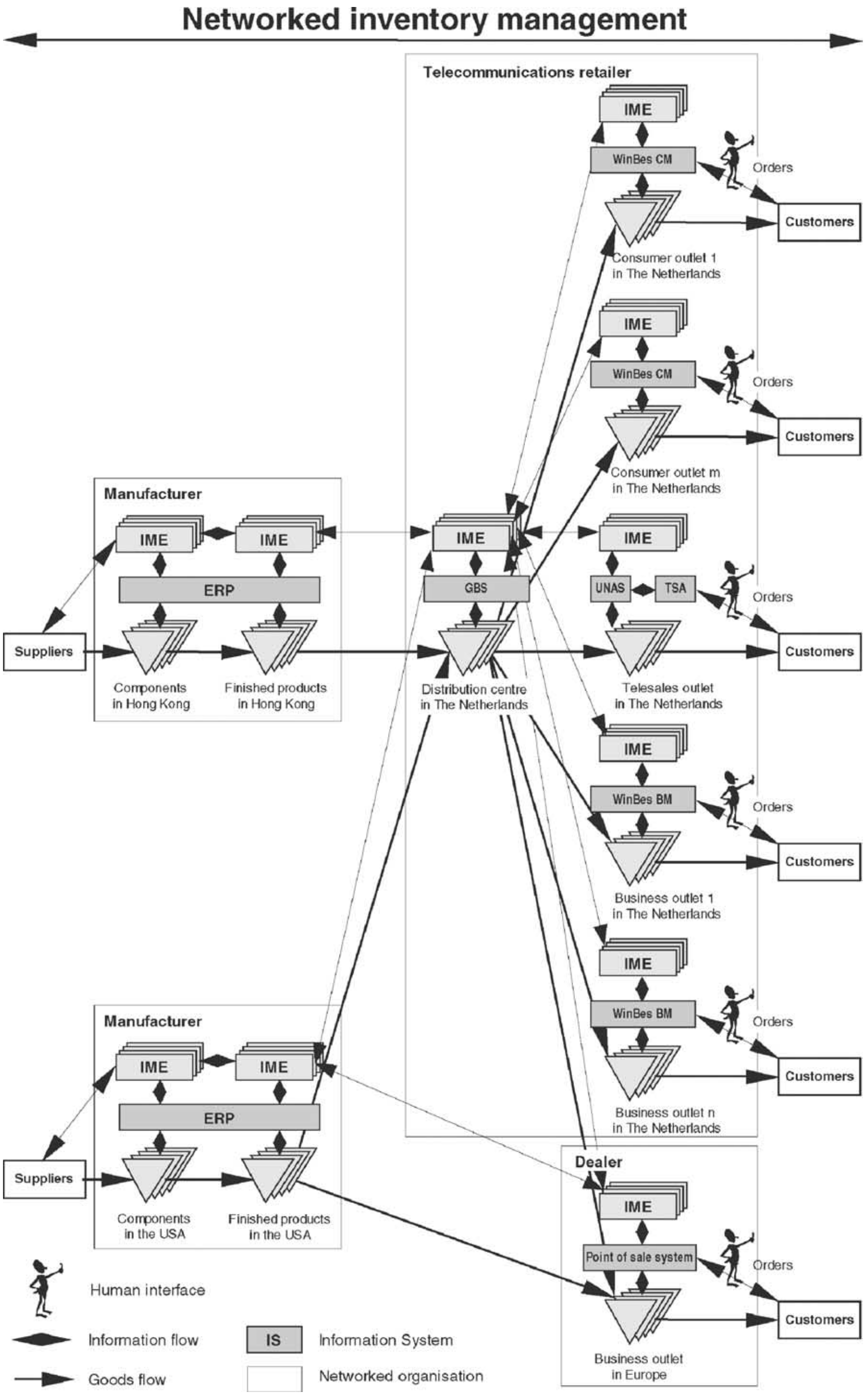

Fig. 7. Inventory management engines in a supply chain for cordless telephones.

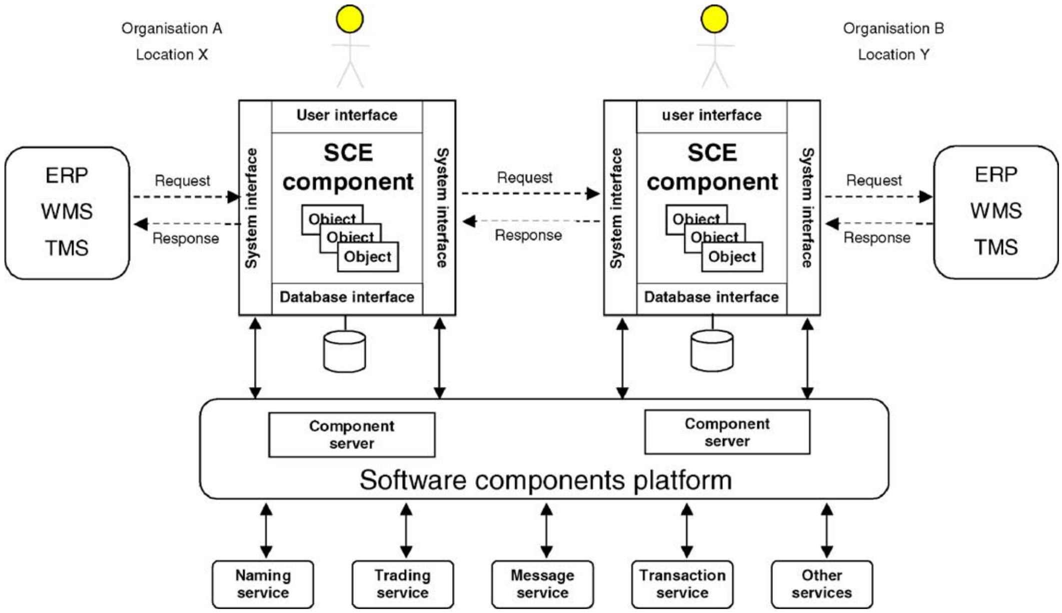

Fig. 8. Components, interfaces and services in the supply chain architecture.

interfaces of the components and common services for the components (see Fig. 8). The supply chain engines are the application components in the architecture, which can be distributed over different servers, locations and organisations.

The SCEs have an object-oriented design and are also built as a collection of object classes and instances [38]. The objects have attributes to store data and have methods to execute functions. Only objects with a remote interface can be accessed by other components. The components provide services to other components through these interfaces [39]. A client component (object) can send a request for a service to a server component (object), which can send back a response to the client component.

The software components have interfaces for communication in the organisation and across the supply chain. The supply chain engines can be equipped with the following interfaces:

- System interfaces to other supply chain engines, for example, interfaces between distribution management engines to support integral distribution management.
- System interfaces to local information systems, for example, interfaces between the inventory management engines and ERP systems to retrieve the stock levels.
- User interfaces to users, for example, the supervisor (strategist) of the inventory management engine who sets the parameters for integral inventory management.
- Database interfaces, for example, to store the details of a customer order in a distribution management engine persistently in a database.

Common services are available in the supply chain management architecture to facilitate the interaction between the distributed components. Some common services in the software component architecture are [38–41]:

- Naming service: the naming service contains a directory of logical names and technical addresses of components. With the help of a naming service, a client component can send a service request to a server component by using its logical name. The naming service provides the technical address of the server component.
- Trading service: components can use a trading service to publish their service interfaces with names, attributes and types. Components can search for available services in other components. The

trading service provides references to the discovered services, so that a client component can send a service request to the server component.

- Messaging service: a messaging service uses intermediate queues for the exchange of messages (requests/responses) between components to guarantee message delivery. Asynchronous communication makes that a client component does not have to wait for a response after a request has been sent. Publish-subscribe features facilitate distribution of messages from a publisher to all subscribers.
- Transaction service: a transaction is a sequence of operations in which several components can be involved. The transaction service makes sure that a transaction complies with the requirements of atomicity, consistency, isolation and durability. A two-phase commit protocol ensures that all components commit to transaction completion or roll back to an original state in the event of a failure.

5.2. Java Enterprise, CORBA and Web Services

Available technologies for the implementation of the software component architecture for supply chain management are Java Enterprise, CORBA and Web Services (see Fig. 9). In the Java Enterprise architecture, the objects in a supply chain engine are written in the Java programming language [39,41–44].

Some Java objects of an application component have a RMI interface, which makes the interface available to Java objects in other components. Remote method invocation (RMI) is a Java technology for interfacing between distributed objects written in Java. A RMI client object can send a service request to a RMI server object. The remote invocation is handled in the Java platform by an RMI stub at the client side and a RMI skeleton at the server side. The protocol for RMI communication is Java Remote Method Protocol (JRMP) or Internet Inter-ORB Protocol (IIOP), both running on top of the TCP/IP Internet protocols.

The Java Enterprise platform consists of J2EE (Java 2 Enterprise Edition) servers, which offer a range of common services to the application components. The Java Naming and Directory Interface (JNDI) service is a naming service, for translation between logical names and technical addresses. In addition, Jini provides a trading service in Java for registration and discovery of distributed services.

The supply chain engines could also be implemented as CORBA components [40,45–48]. The objects in CORBA components can be written in any programming language, like C, C++ or Java. CORBA components interact via IDL interfaces, specified in the neutral interface definition language (IDL) and independent from the programming language of the objects.

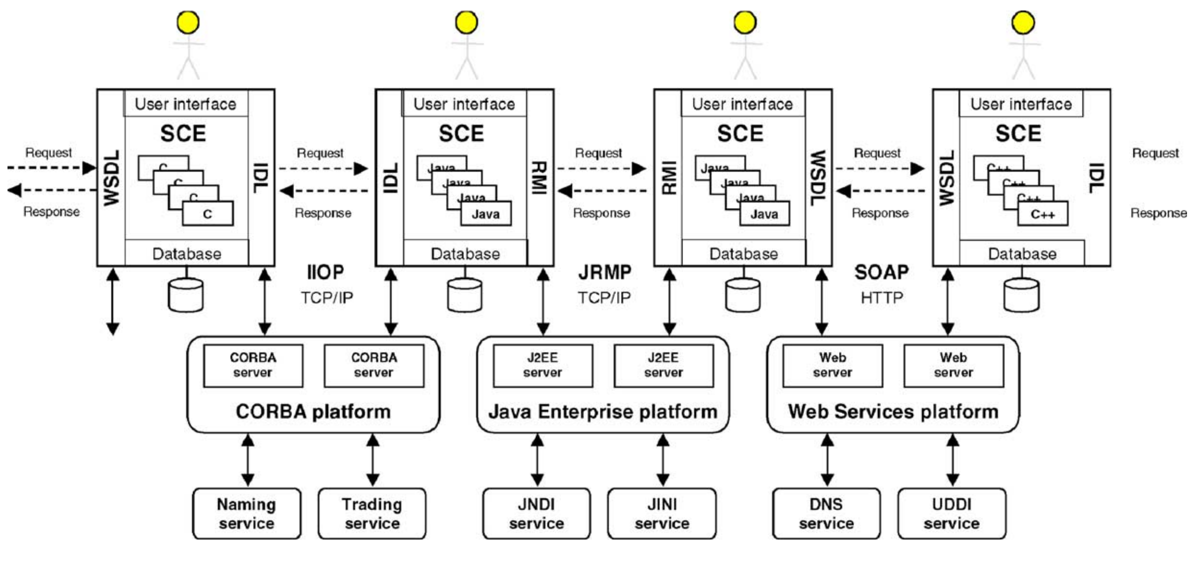

Fig. 9. Java Enterprise, CORBA and Web Services in the component architecture.

When a client component invokes a service on a service component, the CORBA platform internally uses stubs and skeletons to forward the request and to feed back the response. The IIOP is used over TCP/IP to communicate the messages between the components.

The CORBA platform has its own CORBA Naming service for name resolution and CORBA Trading for service discovery. By IDL interfaces in addition to RMI interfaces, Java components can also work together with components in a CORBA platform.

Web Services refer to XML, WSDL, SOAP and UDDI as a set of emerging web technologies for the implementation of a software component architecture [49]. The supply chain engines in the Web Services platform can be built in any programming language as long as they provide the services as specified in the WSDL interfaces.

The Web Services Definition Language (WSDL) uses eXtensible Markup Language (XML) to specify what service is offered by a component and how this service can be requested by other components [50]. The components in the Web Services platform interact by sending messages in XML format that comply to the WDSL interfaces.

The communication protocol for the messages is Simple Object Acces Protocol (SOAP) [51]. SOAP is an XML-based protocol that defines the message envelop, the encoding rules and the transport options. The SOAP protocol makes use of the web protocol HyperText Transfer Protocol (HTTP) to communicate messages between widely available Web servers. Alternatively, SOAP can run over the e-mail protocol Simple Mail Transfer Protocol (SMTP).

The low level naming service for the Web Services platform is Domain Name Service (DNS) which translates URL-names into IP-addresses. The trading service in the Web Services architecture is based on Universal Description, Discovery and Integration (UDDI) [52]. Server components can publish their services in the UDDI service. Client components can get extensive information from the UDDI service about server components, including names, addresses, services and WDSL descriptions.

6. Conclusion

Supply chain management can increase productivity as a result of improved co-ordination across organisations. Networked organisations can provide greater product variety as a result of higher flexibility of the supply chain. In dynamic supply chains there is the need for information systems which support co-ordination and flexibility at the same time.

The software component architecture for supply chain management across dynamic organisational networks consists of existing systems supplemented with new software components. The local management in the architecture is done by existing enterprise resource planning systems, warehouse management systems and transportation management systems.

The integral management in the architecture is executed by supply chain engines. These software components make up a layer of systems for supply chain management running on top of ERP, WMS and TMS. The SCEs provide both the intelligence and the flexibility as required for the integral management in dynamic supply chains. The supply chain engines can be classified in communication, information and management engines.

Inventory management engines can make up a system layer for integral inventory management, using the algorithms BSC, MRP or LRP and providing flexibility for algorithm selection, network configuration and review timing. The IMEs have an object-oriented design, with the classes based on entities in the environment managed by the system. The IMEs can be applied for the integral inventory management across networked organisations in the supply chain of cordless phones.

The software component architecture can be implemented with technology for the application components, their interfaces and common services. The application components can be built from the objects in the object-oriented design. The interfaces are used to send requests and receive responses between the client and server components. Examples of facilitating services are naming, trading, messaging and transaction services.

Java Enterprise, CORBA and Web Services are technologies for the implementation of the software component architecture. The Java platform makes use of RMI-interfaces, the JRMP or IIOP protocol, and provides the JNDI naming service and Jini trading service. The CORBA solution is based on IDL interfaces, the IIOP protocol and the CORBA naming and trading services. The Web Services architecture applies WSDL (XML) interfaces, communicates with the SOAP protocol, and use DNS and UDDI as naming and trading service.

References

[1] M.A.A.P. Verwijmeren, Networked inventory management by distributed object technology, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 1998.

[2] L.M. Ellram, Supply chain management: the industrial organisation perspective, International Journal of Physical Distribution & Logistics Management 21 (1) (1991) 13–22.

[3] T.C. Jones, D.W. Riley, Using inventory for competitive advantage through supply chain management, International Journal of Physical Distribution & Materials Management 15 (5) (1985) 16–26.

[4] J. Stevens, Integrating the supply chain, International Journal of Physical Distribution & Materials Management 19 (8) (1989) 3–8.

[5] M. Christopher, Logistics and Supply Chain Management, Financial Times, London, UK, 1994.

[6] H.S. Sheombar, Understanding logistics co-ordination: a foundation for using EDI in operational (re)design of dyadical value adding partnerships, Ph.D. thesis, Katholieke Universiteit Brabant, Tilburg, The Netherlands, 1995.

[7] M. Walker, Supplier–retailer collaboration in european grocery distribution, Logistics Information Management 7 (6) (1994) 23–27.

[8] H.B. Thorelli, Networks: between markets and hierarchies, Strategic Management Journal 7 (1) (1986) 37–51.

[9] W.W. Powell, Neither market nor hierarchy: network forms of organisation, Research in Organisational Behaviour 12 (1990) 295–336.

[10] M.S. Scott Morton (Ed.), The Corporation of the 1990s: Information Technology and Organisational Transformation, Oxford University Press, New York, USA, 1991.

[11] R.E. Miles, C.C. Snow, Causes of Failure in Network Organisations, California Management Review (1992) 53–72.

[12] F.E. Webster, The changing role of marketing in the corporation, Journal of Marketing 56 (1992) 1–17.

[13] C. Ching, C.W. Holsapple, A.B. Whinston, Toward IT support for co-ordination in network organisations, Information & Management 30 (1996) 179–199.

[14] R. Johnston, P.R. Lawrence, Beyond Vertical Integration: The Rise of the Value-Adding Partnership, Harvard Business Review (1988) 94–101.

[15] W.H. Davidow, S.M. Malone, The Virtual Corporation: Structuring and Revitalizing the Corporation for the 21st Century, Harper Business, New York, USA, 1992.

[16] J.P. Womack, D.J. Jones, From Lean Production to the Lean Enterprise, Harvard Business Review (1994) 93–103.

[17] J. Browne, P.J. Sackett, J.C. Wortmann, Future manufacturing systems: towards the extended enterprise, Computers in Industry 25 (1995) 235–254.

[18] J.E. Aken, L. van Hop, G.J.J. Post, The virtual organisation: a special mode of strong inter-organisational co-operation, in: M.A. Hitt, J.E. Ricart, R.D. Nixon (Eds.), Managing Strategically in an Interconnected World, Wiley, Chichester, UK, 1998.

[19] F.L.G. van Lierop, H. Vanmaele (Eds.), Multi-site Business Information Systems, DELWEL, Den Haag, The Netherlands, 1996.

[20] Ovum, ERP for Manufacturers, Ovum, London, UK, 1997.

[21] M.J. Euwe, Decentralized control systems for logistical coordination, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 1999.

[22] P. van der Vlist, W.J. de Jong, A.E. Kolff, D.J. van der Net, A. van Overbeek, A.T.C. Siebbeles (Eds.), EDI in de Handel (EDI in Trade), Samsom, Alphen aan den Rijn, The Netherlands, 1991 (in Dutch).

[23] P. van der Vlist, W.J. de Jong, A.E. Kolff, D.J. van der Net, A. van Overbeek, A.T.C. Siebbeles (Eds.), EDI in de Industrie (EDI in Industry), Samsom, Alphen aan den Rijn, The Netherlands, 1992 (in Dutch).

[24] P. van der Vlist, W.J. de Jong, A.E. Kolff, D.J. van der Net, A. van Overbeek, A.T.C. Siebbeles (Eds.), EDI in de Transportsector (EDI in the Transport Sector), Samsom, Alphen aan den Rijn, The Netherlands, 1994 (in Dutch).

[25] M.A.A.P. Verwijmeren, P. van der Vlist, K. van Donselaar, Networked inventory management information systems: materialising supply chain management, International Journal of Physical Distribution & Logistics Management 26 (6) (1996) 16–31.

[26] J.F. Magee, Production Planning and Inventory Control, McGraw-Hill, New York, USA, 1958.

[27] A.J. Clark, H. Scarf, Optimal policies for a multi-echelon inventory problem, Management Science 6 (1960) 475–490.

[28] J. Orlicky, Material Requirements Planning, McGraw-Hill, New York, USA, 1975.

[29] A.J. Martin, DRP: Distribution Resources Planning, Oliver Wight, Essex Junction, USA, 1983.

[30] K.H. van Donselaar, Material coordination under uncertainty, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 1989.

[31] J.H.A. Harink, K.M. Duursma, P. van Raad, M.A.A.P. Verwijmeren, Architectuur Voor Besturing van Logistieke Ketens (Architecture for Management of Supply Chains), PTT Research, Groningen, The Netherlands, 1993 (in Dutch).

[32] M.A.A.P. Verwijmeren, Object-oriented design of integral distribution services, KPN Research, Leidschendam, 1996.

[33] M.A.A.P. Verwijmeren, Exploiting distributed object technology to achieve networked inventory management, Computers in Industry 2000 (41) (2000) 239–250.

[34] L.F. Toia, Prototyping networked inventory management information systems, KPN Research, Groningen, The Netherlands, 1997.

[35] E. van het Hof, Networked inventory management information systems: ontwerp en implementatie (networked inventory management information systems: design and implementation), KPN Research, Leidschendam, The Netherlands, 1997 (in Dutch).

[36] L.B.P. Rijen, Herontwerp van een logistieke keten met behulp van simulatie (redesign of a supply chain with the help of simulation), PTT Research, Groningen, The Netherlands, 1994 (in Dutch).

[37] R.J.W. van Grinsven, Case-studie onderzoek bij PTT telecom naar networked inventory management information systems (case study research at ptt telecom into networked inventory management information systems), KPN Research, Groningen, The Netherlands, 1997 (in Dutch).

[38] Object Management Group (OMG), OMG Unified Modeling Language Specification, Version 1.4, Object Management Group, Framingham, USA, September 2001.

[39] P.J. Perrone, V.S.R.R. Chaganti, Building Java Enterprise Systems with J2EE, SAMS Publishing, Indianapolis, USA, 2000.

[40] J. Siegel, CORBA: Fundamentals and Programming, Wiley, New York, USA, 1996.

[41] C.J. Berg, Advanced Java Development for Enterprise Applications, Prentice-Hall, Upper Saddle River, USA, 2000.

[42] B. Shannon, M. Hapner, V. Matena, J. Davidson, E. Pelegri-Llopart, L. Cable, Java 2 Platform Enterprise Edition, Addison-Wesley, Boston, USA, 2000.

[43] N. Kassem, The Enterprise Team Designing Enterprise Applications with the Java 2 Platform Enterprise Edition, Addison-Wesley, Boston, USA, 2000.

[44] Sun Microsystems, Java™ 2 Platform, Enterprise Edition Specification Version 1.4 (Public Draft), Sun Microsystems, Santa Clara, USA, July 2002.

[45] R. Ben-Natan, CORBA: A Guide to Common Object Request Broker Architecture, McGraw-Hill, New York, USA, 1995.

[46] T.J. Mowbray, R. Zahavi, The Essential CORBA: Systems Integration Using Distributed Objects, Wiley, New York, USA, 1995.

[47] R. Orfali, D. Harkey, J. Edwards, The Essential Distributed Objects Survival Guide, Wiley, New York, USA, 1996.

[48] Object Management Group (OMG), The Common Object Request Broker: Architecture and Specification, Version 2.6.1, Object Management Group, May 2002.

[49] R.J. Brunner, I. Cohen, F. Curbera, D. Govini, S. Haises, M. Kloppmann, B. Marchal, K. Scott Morrison, A. Ryman, J. Weber, M. Wutka, Java Web Services, SAMS Publishing, Indianapolis, USA, 2000.

[50] World Wide Web Consortium (W3C), Web Services Description Language (WSDL), Version 1.2 (Working Draft), World Wide Web Consortium (W3C), July 2002.

[51] World Wide Web Consortium (W3C), SOAP, Version 1.2, Part 0, Primer (Working Draft), World Wide Web Consortium (W3C), June 2002.

[52] UDDI.org, UDDI, Version 3.0, Specification (Open Draft), UDDI.org, July 2002.

Martin Verwijmeren graduated cum laude in Industrial Engineering and Management Science. He obtained a PhD from the Eindhoven University of Technology for his research into systems for integral inventory management across networked organisations, using distributed and object-oriented technology. He worked as project manager for KPN Research, the R&D department of a telecommunications company. Martin was senior project manager at TNT, responsible for engineering and IT projects at the logistics service provider. Currently, he is director of MP Objects, a company that develops software components for supply chain management. He has published in the *European Journal of Operations Research*, *The International Journal of Physical Distribution & Logistics Management*, *Computers in Industry* and several logistics journals in The Netherlands.