2009

Architectural Design for Service Supported Collaborative Work in Extended Enterprises

Sven-Volker Rehm
WHU Otto Beisheim School of Management, sven-volker.rehm@whu.edu

Architectural Design for Service-Supported Collaborative Work in Extended Enterprises

Sven-Volker Rehm
WHU – Otto Beisheim School of Management
sven-volker.rehm@whu.edu

ABSTRACT

The idea of service-orientation has contributed to an increased interest in Enterprise Architecture (EA) within information systems research and business. One central, yet often underestimated factor of service-orientation is the integration of services. This paper discusses the issue of service integration in the context of Extended Enterprises (EE) from an architectural perspective. Following a design science approach, the design of a Service-Oriented Architecture (SOA) integration platform is aligned with collaboration characteristics in EE. The platform enables the creation of collaborative information and communication technology (ICT) applications built from services, and facilitates the development of shared service repositories in EE contexts. The paper reflects aspects of the architectural design process leading to the platform design. An observational field study allows a qualitative assessment of the resulting service repositories. It suggests that shared service repositories lead to synergy effects within EE contexts.

Keywords

Service-Oriented Architecture (SOA), Extended Enterprise, Enterprise Architecture, SOA integration platform, Computer-Supported Cooperative Work (CSCW), Enterprise Resource Planning (ERP).

INTRODUCTION

Most industrially manufactured products, as well as an increasing number of services, are resulting today from cooperation of several value added partners within networked relationships. In the recent decades, a variety of alternatives to organize those relationships have emerged. For either way to create value from enterprise networking, the integration of (ICT) technology, business processes and people factors, is considered a prerequisite (Buhmann, Kekre and Singhal, 2005). From a network perspective, in particular the inter-organizational coordination of the value added partnerships, adjusted to the distinct form of networking, has been identified as a basic business need for collaboration. In this context ICT infrastructure development and workflow coordination have been emphasized amongst others as major issues of applied Information Systems (IS) research (Buhmann et al., 2005; Camarinha-Matos and Afsarmanesh, 2004; Filos and Banahan, 2001; Lee and Ng, 1997).

The service-oriented alignment of ICT design to business needs in general is considered a critical issue (Bieberstein, Bose, Walker and Lynch, 2005). This holds true for the development process of service-oriented ICT infrastructures for enterprise networks, which requires addressing the characteristics of specific organizational forms of networking. It is still a challenge in the case of Extended Enterprises (EE), as one particular form of enterprise networking, where service integration is of increased importance (Jagdev and Thoben, 2001; Konsynski, 1993; Krishnan, Rai and Zmud, 2007; Malhotra, Gosain and El Sawy, 2007). It appears likely that the development process of appropriate infrastructures can be improved if it is guided by architectural design considerations. Hence, this paper intends to investigate the contribution service-orientation can offer to architectural design of ICT infrastructures for Extended Enterprises.

In this paper, the term architecture is understood as a principle construction plan, used to classify or characterize major components of a design task and their relations to each other (Aier, Kurpjuweit, Saat, and Winter, 2009). The process of architectural design aims at transforming distinct requirements (problem space) into a schematic or conceptual model of the system to be designed (solution space). An architectural perspective thus considers the design task based on certain pre-assumptions in order to narrow the solution space. ICT infrastructure is understood as the basic, manifest structure facilitating the functioning of ICT systems.

Our research work primarily aimed at formulating an approach for designing collaborative ICT applications adjusted to EE contexts. Taking an architectural perspective, the design of a Service-Oriented Architecture (SOA) integration platform had to be aligned with the business needs of EE. Following a design science approach (Hevner, March, Park and Ram, 2004) oriented at the service-orientation paradigm, the following paragraphs identify basic collaboration characteristics of EE.

Further chapters describe the alignment of the service perspective with the architectural design, the platform design, as well as its evaluation regarding the development of shared service repositories. Finally, contributions to practice and research are summarized.

SERVICE-ORIENTATION IN THE EXTENDED ENTERPRISE CONTEXT

Our research considered collaborative production planning processes in a community of networked small and medium sized enterprises (SMEs) in European textile industries. Such value added communities, often comprising hundreds of partners, have traditionally emerged as economically and socially networked structures (Porter, 1998). However, in view of today's turbulent economic environment, these organically grown structures have become inefficient from a multitude of perspectives, not least due to missing tools for creating and coordinating more adaptable and dynamic collaboration structures. The economically changing environment had resulted in a vivid adoption and evolution of organizational forms within these communities during the recent three decades, mirroring the entire spectrum of economic coordination mechanisms (Williamson 1975). Yet a prevailing characteristic of partnerships in European textile industries is their widespread adoption of the Extended Enterprise paradigm, which is partly due to the complex nature of materials utilized, and partly due to the complexity of the textile value chain, both promoting the adoption of close and long-term relationships (for a discussion of the arguments in this paragraph refer to Rehm, 2007).

As a general phenomenon within these communities its members tend to flexibly form enterprise networks with each other according to emerging business needs, creating for instance close partnerships for production and delivery of specific product lines. Each partner thereby adds complementary value contributions to the network, and creates individual relationships with network partners in terms of information exchange. Our analysis of a particular community in the northern part of Italy has identified 63 different networks within one community. Each network comprised about five to fifteen partners along the value chain and was characterized by distinct forms of partnerships between each of its members (Rehm, 2007). Those partnerships to a significant extent resembled EE relationships. It is obvious that in this context service-orientation for flexible configuration and coordination of collaboration processes is vital for the competitiveness of such a community.

Our research had the goal to design ICT applications supporting collaborative production planning processes within these networks. Consequently, also an adequate ICT infrastructure needed to be provided to the community. The analyzed EE relationships showed some characteristics which could be interpreted as boundary conditions for the design of a supportive ICT infrastructure. The following paragraphs detail the collaboration in this EE context. Four major characteristics are formulated, which have influenced the process of designing an appropriate ICT support.

Composition of information objects (A). A basic prerequisite for collaboration in an EE relationship is the willingness to share relevant knowledge, for instance knowledge about the status of a production plan or the scheduling plan of a machine (Fischer and Rehm, 2004b). The data expressing this knowledge are usually stored in the information systems of an enterprise, e.g. ERP (Enterprise Resource Planning) systems or other data base systems. It is a common approach to deploy message-oriented information exchange in order to enable collaborative work with business partners. The information contained in those messages is then usually replicated and stored at each partner's systems in a distinct (and usually differing) format. Various communication standards, catalogue formats etc. exist, which are however not yet widely adopted throughout the European textile value chain (Aalst and Kumar, 2003; European Commission, 2005). We have observed that significant effort is dedicated to handle the exchange of such business information objects, e.g. for manually entering data in an ERP system, due to lacking ICT interfaces. Most often, mapping processes in between different data structures were not adequately supported by existing systems.

Coordination of business processes (B). A general shortcoming showed the coordinated knowledge exchange in between network partners. A common business requirement was to systematize setup, configuration and coordination of collaborative processes, which were subsequently also to be supported by matching applications. Loh, Koh and Simpson (2006) have focused on extending existing enterprise systems such as ERP systems, and define such "Extended Enterprise Applications" as tools offering industry players "an architecture that could link suppliers and customers with the internal business processes within an enterprise" (which requires building applications providing ERP functionalities to partners without such a system). Accordingly, the ICT infrastructure had to enable configuration of collaborative business applications, e.g. for collaborative production planning. Such applications are usually implementing workflow-coordination, and provide interfaces allowing for the integration of complex data structures from heterogeneous data sources.[1] Thus the idea of a centralized integration

[1] The challenges and opportunities for integration of Knowledge Management or other types of applications and systems cannot be adequately treated here. A detailed discussion can be found e.g. in Xu, Wang, Luo and Shi (2006).

platform emerged, allowing for the creation of new collaboration functionalities without the need to largely modify existing enterprise systems.

Collaboration on basis of shared information objects (C). In the EE context analyzed, business interoperation adopts an intensive form of interaction. Fischer and Rehm (2004a) have proposed a framework for classifying integration strategies for the coordination of production planning, oriented at the degrees of autonomy and mutual synchronization of collaborating partners. As the highest degree of integration, they denote the integration of activities on an operational level. Serrano and Fischer (2007) emphasize that "collaboration implies the full integration of information, of goals, of performances and the integration of actions in terms of operations as well". Collaboration thus requires the synchronization of operational activities (i.e. workflow-coordination) as well as an integration of operations on the data level, i.e. providing common views to a shared data basis (see also Jagdev et al., 2001). As a result of this understanding of collaboration, the design needed to make sure that networking partners can store relevant collaboration data in their own systems, controlling it by selectively giving access to distinct partners during a collaboration process, e.g. to production planning data. At the same time, modifying this data might be allowed also to the collaboration partner, if mutually agreed in the setup of the collaboration process. The basic idea is that partners share data by selectively linking corporate systems to collaborative applications on a common platform. Thus the platform needs to mediate (as a middleware) in between heterogeneous systems.

Composition of business processes (D). As mentioned above, networking partners tend to set up individual business processes for each collaboration case, i.e. exchanged information objects and process coordination within each single partnership are individually designed. This common practice leads to a high flexibility concerning the adaption to technical and planning requirements of highly specific production processes, but at the same time creates an almost unmanageable heterogeneity of business processes, if not supported by workflow-coordination (Nunes, Ferreira and Mendonça, 2005). Thus this "service-orientation" regarding the specificity of the business relation needed to be met with an approach enabling re-usability of process modules, or functionalities, in order to efficiently compose equally specific collaborative business applications. At the same time, in order to enable the management of the various process modules and applications, a modeling approach for the composition of applications was required.

The previous four paragraphs contained a rather informal description of collaboration characteristics and business needs within our research setting (marked A to D). They ought to be understood as guidelines for an ICT infrastructure design process. The following chapter illustrates their role in the design process in more detail.

ARCHITECTURAL DESIGN TOWARDS SERVICE-SUPPORTED COLLABORATIVE WORK

In order to illustrate the process of ICT infrastructure design in this paper we shall limit our focus on two aspects. From our practical experience, it has become evident that in particular the way of handling both, information objects as well as business processes, characterizes the collaboration. Leaving alone further considerations on knowledge and information management of the networked relationships, an adequate ICT infrastructure will at least require meeting the demands imposed by this observation.

A service can be understood as a software component encapsulating a high-level business concept (Krafzig, Banke and Slama, 2005). Services supporting the collaboration in the EE context considered will, according to the character of collaboration, require to be designed in a cooperative manner. This means that on the one hand each partner will contribute own business concepts to a collaborative task, on the other, for each partner an individual view to a shared business concept (or task) is required. As a consequence, the integration of services, as well as their composition in the first place, are of major interest to this architectural design task. Figure 1 illustrates this service perspective defining the principle "solution space" for the architectural design.

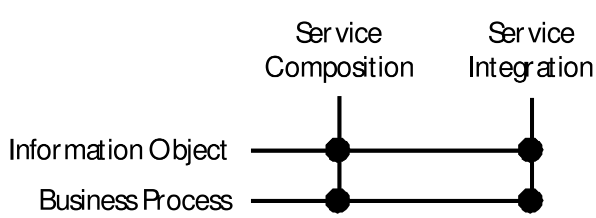

Figure 1. Service perspective inspiring the architectural design

As the first aspect to be considered in the architectural design it needs to be asked how services are composed, "service composition". This involves the question of how information objects and business processes are described with help of services. The collaborative ICT application to be designed is then composed of several services providing the functionality (and shared business concepts) of the included business processes' and information objects' descriptions.

The second aspect to be considered looks at how these applications are executed, i.e. how business processes are coordinated. In other words, the "service integration" aspect describes how services are integrated with each other and orchestrated respectively. Accordingly, service integration refers to the technical processes of data integration and coordination of business processes, while service composition refers to their configuration (adaption) in the first place.

A complete review of the architectural design process leading to the development of the Service-Oriented Architecture (SOA) integration platform would exceed the limits of a research paper. Thus, table 1 resumes the implications of the analyzed collaboration characteristics (A to D) and introduces the major platform components developed, according to the two considered aspects. In the following chapter, a description of the overall system design is given, highlighting the components' details.

	Service Composition	**Service Integration**
Information Objects	(A) Composition of information objects • Identifying knowledge and information to be shared • mapping of data sources and structures • *Data Dictionary (construct)*	(C) Collaboration on basis of shared information objects • Selectively sharing knowledge and information • enabling middleware functionality • *Data Connect*
Business Processes	(D) Composition of business processes • Collaborative modeling and configuration of business processes • defining services • *Composition Environment and Service Repository*	(B) Coordination of business processes • enabling service orchestration • *Service Bus*

Table 1. Aspects considered in the architectural design and SOA integration platform components for service composition and integration

SYSTEM DESIGN OF THE SERVICE-ORIENTED ARCHITECTURE (SOA) INTEGRATION PLATFORM

The Service-Oriented Architecture (SOA) integration platform described in the following paragraphs has been developed by a software development company specialized in ERP systems and Enterprise Application Integration (EAI). The platform provides an environment for development and execution of service-oriented applications. It is intended to enable the integration between the levels of enterprise application systems, services, workflow-integration and desktop integration. In professional business environments, the platform is implemented as a service hub for enterprise networks, based on a Software-as-a-Service business model. It is a freely available commercial product and has been implemented within various business cases in a couple of manufacturing sectors in recent years.[2]

Applications are developed by (1) model-based design (eventually using external tools) and (2) configuration of applications on the platform. The platform provides functionality to connect heterogeneous data sources and a Web Services interface. It includes a graphical user interface (GUI) as well as a variety of other features common to SOA integration platforms.

In this chapter, the major components and further constructs of the platform are introduced, and their contribution to collaboration support is highlighted. An overview is given in Figure 2. Components are understood as pieces of software which use the platform constructs (and data) in order to execute their functionality. The platform constructs are understood as the necessary tools, or (model) language, for the development of services. They are either implemented as software code or defined as models (describing a particular data structure) which can be interpreted by the Service Bus. The platform constructs can be principally grouped in three layers, Database Abstraction Layer, Business Logic Layer, and Presentation Layer.

[2] The platform is distributed by Diasfalisis Ltd. under the name Efikton® (http://www.efikton.com/).

Access to data as well as controlling data manipulation happen on the Database Abstraction Layer. The platform provides middleware functionality, i.e. it can connect to different types of data sources, e.g. data bases, with help of a Data Connect component. The Data Dictionary (DD) construct allows the mapping of existing data base structures (or schemes) from external data sources and represents them internally to the platform. It also stores information on how to connect to external sources. The DD further allows the creation of individual data structures (views, data formats) which are independent from the external sources conditions. It thus hides the heterogeneity of the external environment.

In order to configure applications, the platform provides a couple of constructs, which are defined on the Business Logic Layer. Those constructs allow creating a model of the application, by defining functionalities, process parameters, roles, data objects etc. This model is represented in form of data base entries, which are linked together in the Application (APP) construct. The model refers to the data structures which are defined in the DD. For the composition of applications various further constructs have been defined.

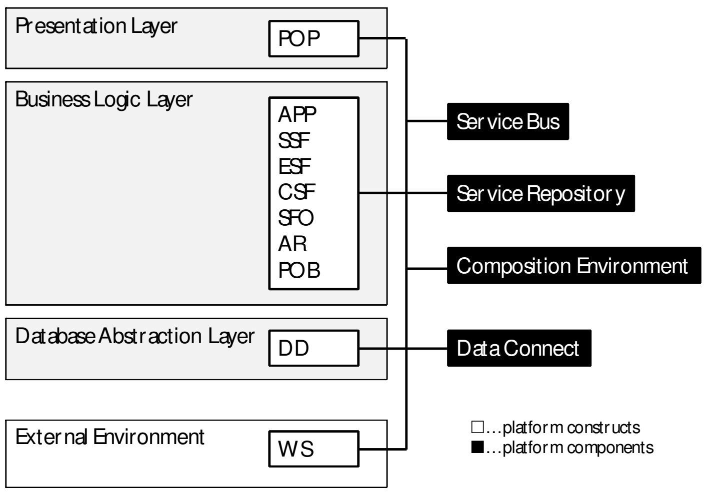

Figure 2. Major components and constructs of the SOA integration platform (abbreviations are explained in the text)

The Service Function construct provides functionalities for presenting and manipulating data. A Simple Service Function (SSF) can consolidate data from several data sources, e.g. data base tables, but allows manipulating data only within one source table, in order not to compromise the consistency of the data source. Those SSF can be linked hierarchically in order to enable manipulation of data from different data sources within one functionality. The resulting object is termed Enhanced Service Function (ESF). Complex Service Functions (CSF) combine SSF and ESF and are capable of representing highly complex services. The construction of Service Functions is already taking into account business rules and service contract parameters. This way, a Service Function can for instance describe the functionality required to fulfill a particular task within a workflow sequence. This functionality is independent from the data sources linked, and manipulations of data using this Service Function can principally be carried out also in the systems of various business partners.

The business logic describing the workflow is modeled with help of the Activity Rule (AR) construct. Activity Rules define the sequence of Service Functions in order to form a workflow. For each of the involved Service Functions, Service Function Operations (SFO) define the basic data manipulation activities a user (or software function respectively) can execute, such as "Edit", "Save" or customized activities changing the workflow status. They also provide links to (eventually external) reporting functionalities etc. Processing Options are used to describe the processing of data within Service Functions in terms of basic statistical or context-sensitive functionality. They are specified for the Business Logic Layer (POB) and the Presentation Layer (POP), the latter comprising functions for displaying the data in GUIs.

The mentioned constructs together provide the tools required to compose models of the information objects and business processes (workflows) of an application, comprising also the business logic and rules to coordinate the respective workflows. The Composition Environment component allows creating instances of each of the platform's constructs. These instances are stored as data sets within the Service Repository. The Service Repository thus is a data base containing the models, or services respectively, which have been defined with help of these constructs. The Service Bus component interprets these models and orchestrates them as services. It relies on the Data Connect functionality to communicate with external data sources.

Each component (and construct) of the platform can be called through an external Web Services (WS) interface, which allows further extension of functionalities.

The above described platform design yields a design characteristic which needs to be emphasized. Though it plays a prominent role in the implications for practice described below, its technical implementation cannot be adequately described here. On the platform, the services are defined in the form of data base entries, and not as software code. A service is thus defined as a model, comprising several constructs in order to describe the structure and behavior of the service. The Service Bus is able to interpret these models. Within the model definition, no software code needs to be created in order to describe a service. The Web Services interface allows the extension of the principle platform-specific functionality. For modeling and configuration of applications, a dedicated modeling approach has been developed, following the idea of the Object Management Group's Model-Driven Architecture (Object Management Group, 2003). This modeling approach allows for efficient configuration and adaption of applications and services. In particular the transformation (or mapping) processes between Platform Independent Model (PIM) and Platform Specific Model (PSM) are extensively simplified, as well as model interchange (Seifert and Beneken, 2005).

VALUE PROPOSITION OF SHARED SERVICE REPOSITORIES IN EXTENDED ENTERPRISES

With the presented platform a number of applications for collaborative production planning have been configured in the named community. Those implementations provided the ground for a qualitative observational field study, which contributed in turn to the further development of the platform. Those implemented applications were used by a network for production of home textiles. The implementation process was accompanied by extensive business process re-engineering in order to provide appropriate information for configuration of applications on the platform. The platform itself was hosted by one of the network members, providing access to its business partners. In the named scenario a group of about 10 SMEs on the value added stage of textile confection which were acting as sub-contractors had to be connected to a common client. The business processes (and services) defined between the client and two "lead" partners were then accustomed to the particularities of the collaboration processes with the other sub-contractors. This way within this pilot scenario, which was chosen from a total of about 150 similar scenarios in the entire community, the implementation time could be reduced from about 4 weeks to a few days per partner.

As has been mentioned before, in EE contexts to a large extent individual business processes are defined between partners, demanding for equally specific collaborative applications. It can be stated that while the initial effort to set up an infrastructure for collaboration was high, through the definition of services in the service repository, a scale effect can be achieved in such an environment. In this context, particularly the creation of a shared service repository contributes to the task of developing collaborative business applications in EE. On the developed platform, it is possible to re-use parts (i.e. selected construct instances) of existing services, for instance those parts of a service that describe certain business rules within a business process context. Those parts can be combined or added to (new) applications, eventually integrating new data sources. This way, a higher efficiency can be reached in the application development process, e.g. in cases when a new partner needs to be connected to a network.[3] A further simplification is achieved by storing services as data models, and not as software code. This explicit representation of services expressed through platform-specific constructs enables an easier handling and adaption (or "replication") of services in a new business case. Furthermore can be stated that the applications developed in our particular implementation scenario were complementary to ERP functionality, respectively providing ERP functionalities for partners without such a system. Especially important to networks of manufacturing SMEs is that implementation costs are considerably reduced through the use of a shared collaboration platform. A shared service repository can in principle be opened up to an entire community, providing all networks with blueprints of services.

Feedback from the manufacturing community's stakeholders suggests a trend of Business Process Modeling (BPM) being capable of achieving increased significance through efforts taken for service-orientation. Integration platforms like the one described in this paper rely on models, and users as well as other stakeholders are confronted with a new quality and relevance of modeling, as a vehicle for formulation of shared business concepts. A more open use of services and service models within collaborative enterprise networks might also pave the way for a new type of "open business architectures" enabling new modi of collaboration (Pisano and Verganti, 2008).

---

[3] This opportunity to re-use (parts of) services however yields further challenges regarding the efficient retrieval of appropriate instances (Riempp and Gieffers-Ankel, 2007; Wang, Xu, and Zhan, 2006).

To date, the platform has been implemented in a couple of additional scenarios, extending the use in EE contexts to other forms of networking. Observations within a number of innovation networks are likely to validate the findings regarding the efficiency of service integration in the EE context.

SUMMARY AND CONCLUSIONS

This paper considers the issue of service integration in the context of Extended Enterprises (EE) from an architectural perspective. Following a design science approach, characteristics of collaboration within a particular business environment of EE are identified. Those influence the architectural design process leading to the development of a Service-Oriented Architecture (SOA) integration platform. Said platform enables the creation of collaborative ICT applications built from services. The components and constructs defined by the platform are presented and the prominent platform characteristic of service definition by modeling is highlighted. Used as an ICT infrastructure for EE, the integration platform facilitates the development of shared service repositories. A short discussion on the value proposition those shared service repositories contribute to EE from an architectural perspective is attached. An observational field study has shown that such shared service repositories can lead to synergy effects within EE.

The overall methodological approach followed in this paper focuses on aligning ICT infrastructure design to characteristics of EE collaboration. A service perspective is taken, and is reduced to two basic aspects, service composition and integration. A more general discussion of architectures for enterprise networks will have to consider a multitude of further aspects of Enterprise Architecture, Knowledge Management or Value Net Governance (Binder and Clegg, 2006; Rai, Sambamurthy and Agarwal, 2008; Wang and Wei, 2007).

More generally, this paper intends to tackle the question to what extent service-orientation can contribute to architectural design processes for ICT infrastructures, i.e. whether service-orientation already provides such a rigorous perspective as does for instance object-orientation. This paper approaches this issue for the case of service integration in EE. It further intends to motivate considerations on developing an extended conception of what could be termed “Service-Supported Collaborative Work” or SSCW, extending the established Computer-Supported Cooperative Work (CSCW) concept by service-orientation and architectural viewpoints. The business case presented on basis of implementation of the SOA integration platform has been a first step towards conceiving service-supported collaborative work also in other contexts of business interoperation. Consequently, the presented platform currently is applied in several innovation networks, and research is focusing on the development of applications for collaborative innovation management.

REFERENCES

1. Aalst, W. and Kumar, A. (2003) XML-Based Schema Definition for Support of Interorganizational Workflow, *Information Systems Research* 14, 1, March 2003, 23-46.
2. Aier, S., Kurpjuweit, S., Saat, J. and Winter, R. (2009) Enterprise Architecture Design as an Engineering Discipline, *AIS Transactions on Enterprise Systems* 1, 1, 36-43.
3. Bieberstein, N., Bose, S., Walker, L. and Lynch, A. (2005) Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals, *IBM Systems Journal*, *44, 4*, 691-708.
4. Binder, M. and Clegg, B. (2006) A conceptual framework for enterprise management, *International Journal of Production Research*, 44, 18/19, 3813-3829.
5. Buhman, C., Kekre, S. and Singhal, J. (2005) Interdisciplinary and Interorganizational Research: Establishing the Science of Enterprise Networks, *Production & Operations Management*, 14, 4, 493-513.
6. Camarinha-Matos, L.M. and Afsarmanesh, H. (2004) The emerging discipline of collaborative networks, in Luis M. Camarinha-Matos (Ed.) *Virtual Enterprises and Collaborative Networks*, Kluwer Academic Publishers, Boston, Dordrecht, London, 3-16.
7. European Commission, Enterprise & Industry Directorate General, e-Business W@tch (2005) ICT and Electronic Business in The Textile and Clothing Industry - ICT adoption and e-business activity, July 2005, Retrieved from http://www.ebusiness-watch.org/events/documents/WS051014_Conclusions.pdf.
8. Filos, E. and Banahan, E.P. (2001) Will the organisation disappear? The challenges of the new economy and future perspectives, in Luis M. Camarinha-Matos, Hamideh Afsarmanesh, and Ricardo J. Rabelo (Eds.), *E-Business and virtual enterprises: managing business-to-business cooperation*, Kluwer Academic Publishers, Boston, Dordrecht, London, 3-20.

9. Fischer, T. and Rehm, S.-V. (2004a) A Cybernetic Approach Towards Knowledge-based Coordination of Dynamically Networked Enterprises, in Luis M. Camarinha-Matos (Ed.) *Virtual Enterprises and Collaborative Networks*, Kluwer Academic Publishers, Norwell, 135-144.
10. Fischer, T. and Rehm, S.-V. (2004b) Knowledge-based Manufacturing Management in Dynamically Networked Enterprises, in Klaus-Dieter Thoben, Kulawant S. Pawar and Frithjof Weber (Eds.) *Proceedings of the 10th International Conference on Concurrent Enterprising: Adaptive Engineering for Sustainable Value Creation*, Nottingham, University of Nottingham, 309-313.
11. Hevner, A., March, S., Park, J. and Ram, S. (2004) Design science in information systems research, *MIS Quarterly*, 28, 1, 75-105.
12. Jagdev, H. and Thoben, K. (2001, July) Anatomy of enterprise collaborations, *Production Planning & Control*, 12, 5, 437-451.
13. Konsynski, B.R. (1993) Strategic control in the extended enterprise, *IBM Systems Journal*, 32, 1, 111- 142.
14. Krafzig, D., Banke, K. and Slama, D. (2005) Enterprise SOA, Service-Oriented Architecture Best Practices, Prentice Hall International, New Jersey.
15. Krishnan, M., Rai, A. and Zmud, R. (2007) The Digitally Enabled Extended Enterprise in a Global Economy. *Information Systems Research*, 18, 3, 233-236.
16. Lee, H.L. and Ng, S.M. (1997) Introduction to the special issue on global supply chain management, *Production and Operations Management*, 6, 3, 191-192.
17. Loh, T., Koh, S. and Simpson, M. (2006) An investigation of the value of becoming an extended enterprise, *International Journal of Computer Integrated Manufacturing*, 19, 1, 49-58.
18. Malhotra, A., Gosain, S. and El Sawy, O. (2007) Leveraging Standard Electronic Business Interfaces to Enable Adaptive Supply Chain Partnerships, *Information Systems Research*, 18, 3, 260-279.
19. Nunes, Á., Ferreira, J. and Mendonça, J. (2005) Distributed business process coordination: a functionally oriented infrastructure, *International Journal of Computer Integrated Manufacturing*, *18, 5*, 418-426.
20. Object Management Group (2003), MDA Guide Version 1.0.1, Retrieved from http://www.omg.org/docs/omg/03-06-01.pdf.
21. Pisano, G. and Verganti, R. (2008) Which Kind of Collaboration Is Right for You?, *Harvard Business Review*, *86,* 12, 78-86.
22. Porter, M. (1998) Clusters and the new economics of competition, *Harvard Business Review*, 76, 6, 77-90.
23. Rai, A., Sambamurthy, V. and Agarwal, R. (2008) How CIOs can enable governance of value nets, *MIS Quarterly Executive*, 7, 4, 193-204.
24. Rehm, S.-V. (2007) Architektur vernetzter Wertschöpfungsgemeinschaften der Textilwirtschaft. (Architecture of Networked Value Added Communities of Textile Industries). Dissertation, Universität Stuttgart, Online Publication/URL http://elib.uni-stuttgart.de/opus/volltexte/2007/3197/.
25. Riempp, G. and Gieffers-Ankel, S. (2007) Application portfolio management: a decision-oriented view of enterprise architecture, *Information Systems & e-Business Management*, *5, 4*, 359-378.
26. Seifert, T. and Beneken, G. (2005) Evolution and Maintenance of MDA Applications, in Sami Beydeda, Matthias Book and Volker Gruhn (Eds.) *Model-Driven Software Development*, Springer, Berlin, 269-286.
27. Serrano, V. and Fischer, T. (2007) Collaborative innovation in ubiquitous systems, *Journal of Intelligent Manufacturing*, 18, 5, 599-615.
28. Wang, E. and Wei, H. (2007) Interorganizational Governance Value Creation: Coordinating for Information Visibility and Flexibility in Supply Chains, *Decision Sciences*, 38, 4, 647-674.
29. Wang, Z., Xu, X. and Zhan, D. (2006) Component reuse based agile reconfiguration for Enterprise Resource Planning (ERP) systems in manufacturing enterprises, *International Journal of Production Research*, *44, 23*, 5107-5129.
30. Williamson, O.E. (1975) Markets and Hierarchies: Analysis and Antitrust Implications / A Study in the Economics of Internal Organization, The Free Press, New York.
31. Xu, L., Wang, C., Luo, X. and Shi, Z. (2006) Integrating knowledge management and ERP in enterprise information systems, *Systems Research & Behavioral Science*, *23, 2*, 147-156.