# ARCHITEKTURE FOR A SME-READY ERP-SOLUTION BASED ON WEB-SERVICES AND PEER-TO-PEER-NETWORKS

Jorge Marx Gómez, Claus Rautenstrauch
*Institute of Technical and Business Information Systems, Faculty of Computer Science, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany*

Keywords: ERP, Web Service, SOAP, Business Component, Peer-to-Peer (P2P), CoBCoM, SOA

Abstract: Although the requirements of small- to medium-sized enterprises (SMEs) for enterprise resource planning systems (ERP) are very similar to those of big corporations, there is still a lack of solutions for SMEs, because the roll-out as well as the maintenance is very expensive. It has become clear that EDP branch solutions, application service providing and stripped-down software versions do not offer satisfying solutions. For solving these problems, we propose an architecture for a distributed ERP-system based on web-services and peer-to-peer-network technology whose roll-out and maintenance is better affordable for SMEs than traditional systems.

## 1 INTRODUCTION

In Germany, small- to medium-sized production-oriented enterprises (SMEs) are using a differentiation strategy for staying profitable in the world-market. This kind of strategy aims to maintain a leading position by having unique selling propositions. This can be accomplished by following efforts:

- production of customized goods,
- adherence to high quality standards,
- providing of product-related services.

Companies following such a strategy must be able to support all business processes in the context of production-variation by flexible planning. For this, the software-industry offers so-called enterprise resource planning systems (ERP-systems).

Modern ERP-systems, like SAP R/3, Oracle Applications or Microsoft Navision are complex, consist of many software components and must be costly customized to the customers' requirements. Furthermore, expensive high-end hardware needs to be deployed, especially in case of application- and database-servers. During operation the software and hardware needs to be permanently maintained by specially-trained staff. The resulting Total Cost of Ownership (TCO) often is too expensive for SMEs, although ERP-systems are fulfilling their requirements, especially the support of flexible planning. This reveals the real dilemma of ERP-system usage by SMEs: The requirements of SMEs regarding functionality, operation and customizing are very similar to those of large businesses but SMEs cannot effort the deployment of these systems (ERP-SME-dilemma).

The software-industry has identified this dilemma as well and offers following solutions, whose success is at least doubtful:

- Application Service Providing (ASP): Within ASP an external service provider operates the application including all basic systems (system software and hardware). For this service, the customer only pays periodically license fees and needs to install the necessary infrastructure for the local use of the client software. The risks of operation is completely assigned to the provider and the costs is reduced compared to local operation because the provider can utilize scaling effects. However, SMEs must give their trade secrets and internal knowledge to the hands of the provider, which is the main barrier for embracing ASP.
- EDP branch solution: Another solution is the usage of pre-configured systems, so-called EDP branch solution. The installation and customizing are simpler and therefore cheaper compared to

Marx Gómez J. and Rautenstrauch C. (2005).
ARCHITEKTURE FOR A SME-READY ERP-SOLUTION BASED ON WEB-SERVICES AND PEER-TO-PEER-NETWORKS.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 383-390
DOI: 10.5220/0002507503830390
Copyright © SciTePress

the deployment of classical ERP-systems. However, the maintanance of the systems is still very complex.

- "Mini-ERP-Systems": Such systems, like SAP Business One, offer a reduced set of functionality and customizing options and are offered for reduced fees. These restrictions contradict the requirements for flexibility outlined above.

Because of this, a new approach was developed, which we call "Aldi-ERP-system". This system will be deployed as a basic version with a graphical user interface, basic functionality, a workflow-system and a database, which can - exaggeratedly - be purchased in a supermarket. Thus the name. The technical foundation is simply a normal workplace PC and one server equipped with standard PC technology. All other domain-related and company-specific functionality are integrated via web-services, which are offered via a peer-to-peer-network (P2P-network). This concept has following advantages:

- Every none-standard functionality needs to be implemented as a web-service only once. Several companies can then subscribe to it via the Internet. The subscription is considerably cheaper than developing this functionality oneself., because web-services can be used by many customers and are centrally managed by the developers.
- P2P-technology enables providers to redundantly offer their web-services and thus the whole system has a high availability rate.
- All business-critical data and knowledge about the success-enabling business process are only stored locally within the SME organisation.

Because of these properties, the system allows to configure ERP-systems in a way that on the individual requirements of a specific SME can be fulfilled without the need to share business critical data with third parties and with justifiable costs.

## 2 INITIAL SITUATION

Since the mid-90s ERP-systems are very successful world-wide and are used within virtually every large enterprise. The most important software-suppliers of ERP-software today are Oracle, Peoplesoft, Microsoft, Baan and SAP AG, whereas SAP AG is the world-wide market-leader with its SAP R/3 Enterprise system. P2P-networks and web-services are the foundation on top of which the Aldi-ERP-concept should be built. Following, the actual developments in research shall be presented.

### 2.1 ERP-Systems

An ERP-system (ERP = Enterprise Resource Planning) is an integrated standard software for businesses which

- Offers functionality for all parts of the business,
- Is extensible, especially by integrating other information systems, and
- Is usable in many branches (see Rautenstrauch/ Schulze, 2003, p. 314f.).

Consequently, the business functionality of SAP R/3 Enterprise is organized in main components. Those are related to following areas (without claiming completeness):

- Logistics: Distribution, Materials Management, Production Planning, Quality Management, Maintanance
- Human Resource Management: Personell Management
- Accounting: Finance, Controlling, Treasury, Asset Management
- General: Project Management, Service Management

Because ERP-systems need to be independent of the branch the using company is working in, the software needs to be adapted before each deployment to the organizational structure. This process is called Customizing and is done by systematically setting parameters. In practice, this is a very complex task, because one has to find out, which one of the many combinations of parameters is the right setting for the specific business. If there are n possible values of a parameter and m parameters in total, there are $n^m$ possible settings for the whole system because the parameters are independent of each other. Within SAP R/3 Enterprise there are hundreds of parameters with at least 2 possible values.

### 2.2 Peer-to-Peer-Networks

P2P-Networks are compounds of equitable peers, which offer one another resources without central control instances (see Schoder/Fischbach, 2003, p. 313). The term ‚peer-to-peer’ (P2P) refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner (see Milojicic et. al., 2002, p. 1).

As a result, the three most important properties of P2P-networks are (see Schoder/Fischbach, 2003, p. 313) and (Milojicic et. al., 2002, p. 12):

1. Two way offering of resources: Every peer can act as a client as well as a server, that means it can offer or request resources attachted to the network,
2. Decentralization: direct exchange between quitable peers without central control,
3. Autonomy: Peers choosing which resources are offered

Further properties of P2P-networks are:

- Scalabiltiy and ad-hoc compound: P2P-networks can be reconfigured in run-time, that means peers can be added or removed,
- Anonymity: By using several techniques users of P2P-networks can be anonymous,
- Distributed Ownership and distributed property: The total cost of ownership is reduced because peers are sharing otherwise unused resources,
- Reliability: Because of the decentral architecture there is no single point of failure. If one peer fails others can take over its jobs.

P2P-networks can be categorized within four categories. Used for distributed computing, computing intensive tasks are broken down into small pieces which are solved simultaneously by many peers. One prominent example of this kind of network is SETI@Home. File sharing enables the virtual aggregation of many peers' storage space. Popular implementations are Napster and Gnutella. The third category consists of instant messaging (IM)-systems and collaborative tools. Examples for these are ICQ and AOL Instant Messenger. The fourth category are platforms, e.g. JXTA by Sun Microsystems and Microsoft's .NET My Services (see Milojicic et. al., 2002, p. 7).

## 2.3 Web-Services

The term web-service has no standard definition (see Berner Fachhochschule, 2003). Within this paper, we use the definition of the working group "Development of web-service-based applications" of the German Society for Informatics (German Society for Informatics, 2003): "Web-Services are self-descriptive, encapsulated software-components, which offer an interface which can be used to remotely invoke their functionality and which can be loosely combined via the exchange of messages.

There are lots of concepts, protocols and models for developing web-services, whose standardization is driven by different consortia, most importantly the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS). The following standards, all based upon the eXtensible Markup Language (XML), are the foundation of the web-service infrastructure (see Bettag, 2001):

- Simple Object Access Protocol (SOAP) (see W3C, 2003): SOAP defines a light-weight protocol for message-exchange and function calls between applications and supports asynchronous and synchronous request/response communication.
- Web Service Description Language (WSDL) (see W3C, 2003): Via WSDL the interfaces and functionality of web-services are described.
- Universal Description, Discovery and Integration (UDDI) (see OASIS, 2002): In order to find web-services, the services, their interface- and functionality-description can be registered in a central repository. The standard for this as well as the central global register is called UDDI (see Beimborn, Mintert and Weitzel, 2002, p. 278).

The connection of web-services is called web-service-choreography, web-service-composition or web-service-orchestration. For accomplishing this task, there is no single accepted standard available. Instead competing standard proposals by different vendors exist, e.g. BPEL4WS (see Curbera et al., 2003), WSFL (see Leymann, 2001), XLANG (see Thatte, 2001) or WSCI (see Arkin et. al., 2002). Some of these are built upon workflow-description languages, like WSFL (IBM). The most important difference between these, sometimes also referred to as flow-languages is the support of state: While WSDL is a stateless language, flow-languages support different states of the corresponding web-services (see Aalst , 2003, p. 74).

## 3 THE COBCOM-ARCHITECTURE AS THEORETICAL FOUNDATION

The theoretical foundation for the Aldi-ERP-system is the CoBCoM-architecture (Common Business Component Model) (Turowski, 2003 and Rautenstrauch/Turowski, 2001). A business component (BC) is thereby defined as a component which offers a set of services of a specific operative application domain. A component is defined as a reusable, self-contained, and marketable software artefact, which offers services via well-defined interface and can be used in combination with other components within applications in a manner not foreseeable in its development. In this way, web-services can be interpreted as an implementation of business components.

## 3.1 BCArch

An orchestration which is based solely on putting more or less complex business components together, fails because of the complexity of interface definition. Because of this, BCArch (Business Component Architecture) has been developed to provide a framework for the system design (see figure 1). Corresponding to BCArch an application design consists of following parts:

- A C*omponent-Application-Framework* is a part of the system which offers application-domain specific (standard-)services to the business components and likewise acts as an integration plattform for them.
- A *Component-System-Framework* is a part of the system, in which business components offer application-invariant, middleware-related services.

from the view of BCArch, the (business-related) basic functionality is placed in the Component-Application-Framework, whereas domain-invariant software-artefacts, like database management software and workflow-systems, belong to the Component-System-Framework. The workflow-models are themselves responsible for execution the components' services in the right order suitable for the application (Inter-Component-Coordination). Thus, they are application-specific and part of the Component-Application-Framework (see figure 2).

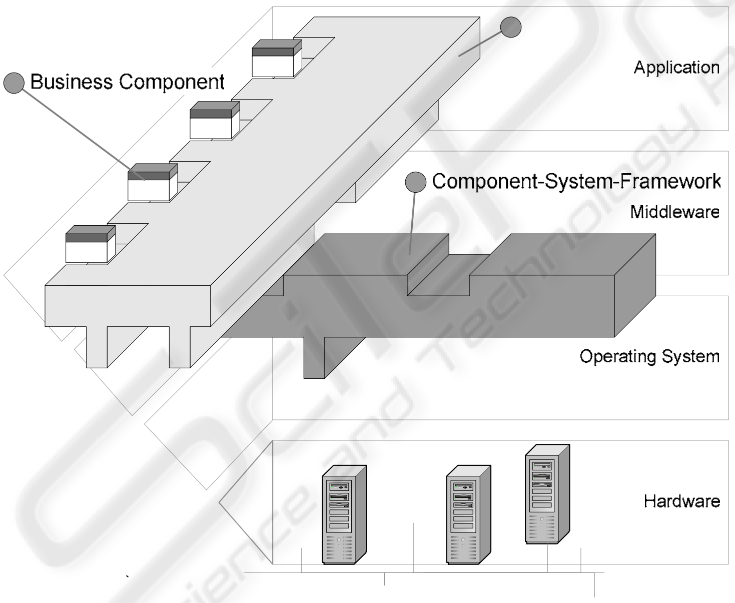

Figure 1: BCArch

## 3.2 Business Components versus Web-Services

Very suitable to differentiate between web-services and business-components is the definition of (Krammer/Turowski, 2001, p. 2) following (Bettag, 2001), because it used software-specific properties. Following, web-services are "…a together-belonging set of marketable services, which are offered via the WorldWideWeb to an authorized group of users using standardized communication protocols and well-defined interfaces. Web-Services contain interface-descriptions of the offered services, which are written in a standardized description language. Details of the implementation are hidden from the user."

The main difference between a business component and a web-service is that with web-services only their services are marketed and not the software itself. Because of this, it is not necessary that the implementation of a web-service is reusable. Correspondingly, web-services need not be self-contained, because it does not matter for the user of

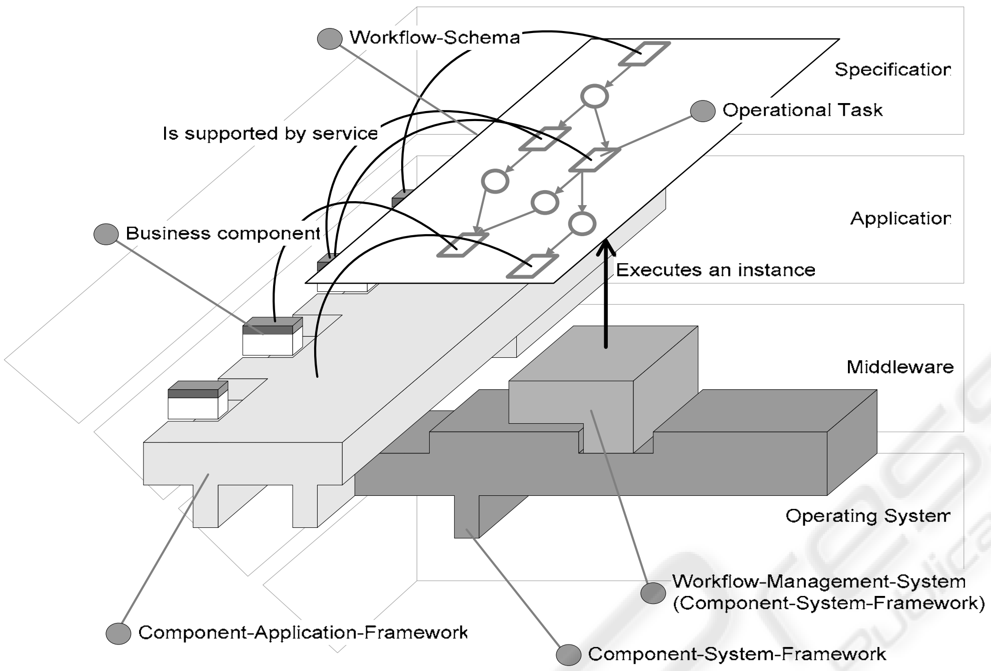

Figure 2: Embedding Workflows into BCArch

a web-service which web-service offers which functionality (see Krammer/Turowski, 2001, p. 3f.). Another difference is the explication of the run-time environment of web-services. However, business components and web-services have in common that

- Services are offered via well-defined interfaces,
- The implementation is hidden from the user,
- And Services can be arbitrarily combined.

Therefore, web-services are a special case or an implementation of business components from the point of view of software-architecture.

## 4 ALDI-ERP-SYSTEM ARCHITECTURE

To solve the ERP-SME-dilemma an architecture based on BCArch is proposed as illustrated in figure 3 (according to Marx Gómez et. al., 2004). Workflowmanagement- and databasemanagement-systems (WFMS und DBMS) are the system-framework. The application-frameworks consist of services for the user-interface and the basic ERP functionality. Both frameworks will be delivered as frontend-systems and are pre-installed.

Further functionality can be attached via web-services, which are offering the implementation for the corresponding business components. The web-services themselves are distributed over a P2P-Network. Depending on availability and performance criteria defined in Service Level Agreements (SLAs) between the users and the web-service-providers, web-services can be redundantly offered by many peers. These peers can be installed within the company as well as be installed off-site.

By using a P2P-network with redundantly distributed web-service, high availability can be achieved. If one peer fails, another peer can take over its role and offer the same functionality. The hardware-requirements are minimal because of the resource-sharing within the network. In connection with an out-of-the-box-solution the need for highly-skilled technical personal for administering and maintaining the system is low.

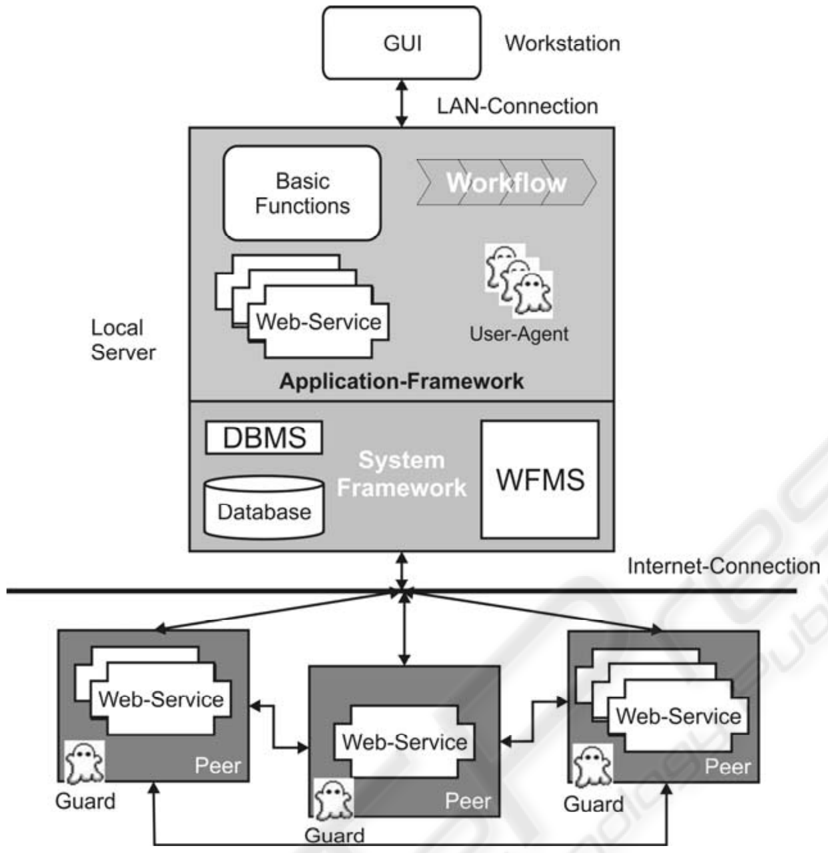

Figure 3: Architecture "Aldi-ERP-System"

The communication between the different peers will be done via SOAP-messages, the web-services will be described with WSDL and can be retrieved via UDDI By using these standards, and other offers of web-services can be easily integrated in the system. The system can be easily extended with new functionality by using new web-services which means that companies can very flexible use and extend their system. Subscription, execution ordering and parameterisation of the web-services will be done via the workflow-component. The enactment-service of the workflow-component within the frontend takes over the request for web-services during run-time.

Access rights to the functionality of the ERP-system will be managed in the front-end as well. Because access-rights are relevant when accessing web-services, a mobile user agent will be used which has its user's access rights. The agent will be sent to the web-service where a static guard agent will check the permissions and open the service accordingly. This way, it is assured that the same security-policies are effective when using local components or remote web-services.

Another security problem exists, if data are exchanged between front-end and web-service. Because the web-service is a shared resource accessible by different companies, data can principally be read by third parties. A solution to this is the usage of XML documents which are encrypted using the XML Encryption Standard. XML Encryption was standardized by the W3C in the XML Encryption Working Group and is a W3C recommendation since $10^{th}$ of December 2002.

The coupling of ERP-systems of many companies in the context of a supply chain is relatively simple, because it does not differ from extending the system via new web-services. This implies, that by constructing ERP-systems out of web-services the difference between inter- and intra-enterprise integration vanishes from the technical point of view.

With conventional ERP-systems the number of installed systems is roughly the number of the using enterprises. However, with the proposed concept, all components exist only once (apart from redundant installations for high availability). If a company needs a specific functionality, this can be subscribed to as web-services or a software-developer can write a new web-service. This way, an agile ERP-system is created, which is improved by a development community all the time similar to the open-sourcemovement. Another side-effect is that the market-barriers for small software companies are lower. Innovative solutions for single business requirements can be directly offered without building a complete application around the solution. This improves the competition in the oligopolistic ERP-market and will finally help the ERP-users.

The proposed architecture fulfils the requirements for service-oriented architectures (SOA). According to (Webopedia, 2004) SOAs are defined as: "Abbreviated *SOA*, an application architecture in which all functions, or services, are defined using a description language and have invokable interfaces that are called to perform business processes. Each interaction is independent of each and every other interaction and the interconnect protocols of the communicating devices (i.e., the infrastructure components that determine the communication system do not affect the interfaces). Because interfaces are platform-independent, a client from any device using any operating_system in any language can use the ser-vice."

The main requirement of a SOA is, that

- Functionality is only accessed via standardizes and platform-independent interfaces,
- the interfaces are defined in a standardized specification language, and
- the interaction between services must be independent of the underlying technologies.

These are fulfilled by web-services because only standards like SOAP, WSDL and XML documents are used. Because of this, the presented system architecture is an implementation of a SOA.

Comprising, the presented architecture provides a flexible and cost-effective alternative, to the classical, proprietary ERP-systems and are suited to fulfil the special requirements of SMEs.

## 5 CONCLUSIONS AND OUTLOOK

The usage of ERP-systems by SMEs has been a compromise because of the ERP-SME-dilemma: The usage meant abandonment of functionality (i.e. the usage of "Mini-ERP-Systems"), the sharing of critical business knowledge with 3$^{rd}$ parties (ASP) or waiving of individuality (EDP branch solutions). With the proposed concept, this compromise should be made unnecessary.

With the Aldi-ERP-System, it is possible to build and maintain fully functional, high available, inter- and intra-enterprise integrateable ERP-systems for SMEs with justifiable costs. The realization of department- and company-wide business processes will keep up the competitive position of SMEs in the world-wide market. Furthermore, ERP-Systems are the basis for optimizing internal business processes, for implementing a company-wide controlling and for participating in supply-chains – advantages which are realized by larger enterprises for a while.

The architecture also not only affects SMEs but also influences small and medium software companies. These are able to offer their innovative software in form of web-services to the ERP-market without risking an expensive development of ERP-systems from scratch themselves. This increases the number of suppliers in the ERP-market and finally results in increased competition between software companies which is attractive for the ERP-users.

The next step will be the development of an Aldi-ERP-prototype to test the presented concepts. The BCLifeCycle seems to be suited for this as it is the proposed method in the CoBCoM model. The aim is to validate and perhaps adapts the CoBCoM model to service-oriented architectures.

## REFERENCES

Aalst, W. v .d., 2003. Don't go with the Flow: Web Services compositions standards exposed, IEEE Intelligent Systems, January/February.

Arkin et. al., 2001. Web Service Choreography Interface 1.0, http://www.sun.com/software/xml/developers/wsci/wsci-10.pdf.

Beimborn, D., Mintert, S., Weitzel, T., 200. Web Services und ebXML, Wirtschaftsinformatik, 3, p. 277-280.

Berner Fachhochschule, 2003. Web Services im eGovernment, http://webservice.iwv.ch/definitionen.htm, August.

Bettag, U., 2001. Web-Services, Informatik Spektrum, 5, p. 304.

Curbera et. al., 2003. Business Process Execution Language for Web Services Version 1.1, http://www-106.ibm.com/developerworks/webservices/library/ws-bpel, September.

Symposium auf der 33. Jahrestagung der GI, 2003. Entwicklung Web-Services-basierter Anwendungen, http://www.winf.tu-darmstadt.de/arbeitskreis/symposium.htm, August.

Günther, O., Tamm, G., Hansen, L., Meseg, T., 2001. Application Service Providers: Angebot, Nachfrage und langfristige Perspektiven, Wirtschaftsinformatik 43, p. 555-568.

Krammer, A., Turowski, K., 2001. Spezifikationsbedarf von Web-Services. In *Ortner, E., Overhage, S., 1.*

*Workshop "Entwicklung von Anwendungen auf der Basis der XML Web-Service Technologie*, p. 1-14, Darmstadt.

Leymann, F., 2001. Web Services Flow Language (WFSL 1.0), http://www-3.ibm.com/software/solutions/ webservices/pdf/WSFL.pdf.

Milojicic et. al., 2002. Peer-to-Peer Computing, HP Labs (HPL-2002-57), Palo Alto.

Marx Gómez, J., Krüger, O., Kühne, C., Lübke, D., Rautenstrauch, C., 2004. Developing a Distributed ERP System based on Peer-to-Peer-Networks and Web Services. In *ICSC, Engineering of Intelligent Systems – EIS 2004 (CD-ROM)*, Funchal.

Organization for the Advancement of Structured Information Standards (OASIS), 2002. UDDI Version 2.04 API Specification, http://uddi.org/pubs/ ProgrammersAPI-V2.04-Published-20020719.htm.

Rautenstrauch, C., Schulze, T., 2003. *Informatik für Wirtschaftswissenschaftler und Wirtschaftsinformatiker*. Springer, Berlin, Heidelberg.

Rautenstrauch, C., Turowski, K., 2001. Common Business Component Model (CoBCoM): Generelles Modell komponentenbasierter Anwendungssysteme. In *Buhl, H.-U., Huther, A., Reitwiesner, B., Information Age Economy*. Heidelberg, p. 681-695.

Schoder, D., Fischbach, K., 2003. Peer-to-Peer-Netzwerke für das Ressourcenmanagement, Wirtschaftsinformatik 2003(3), p. 313-323.

World Wide Web Consortium (W3C), 2003. Simple Object Access Protocol (SOAP) 1.1, http://www.w3.org/TR/SOAP/, Requested in August 2003.

World Wide Web Consortium (W3C), 2003. Web Services Description Language (WSDL) 1.1.

Webopedia, 2004. Service-oriented Architecture. http://www.webopedia.com/TERM/S/Service_Oriented_Architecture.html, Requested on May 11, 2004.

Thatte, S., 2001. XLANF: Web Services for Business Process Design, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm, Microsoft, Redmont, Washington.

Turowski, K., 2003. *Fachkomponenten*, Shaker. Aachen.