View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by SZTAKI Publication Repository

Published in: Alt, R., Fähnrich, K.-P., Franczyk, B. (Eds.), Proceedings First International Symposium on Services Science ISSS'09, Logos, Berlin, 2009 (www.logos-verlag.de)

Adaptive SLA Management along Value Chains for Service Individualization

Paul Karaenke[1], András Micsik[2], Stefan Kirn[1]

[1]University of Hohenheim, Schwerzstr. 35, D-70599 Stuttgart, Germany
[2]MTA SZTAKI, Budapest Lágymányosi u. 11. H-1111, Hungary

{karaenke|kirn}@uni-hohenheim.de, micsik@sztaki.hu

Abstract: The object of our investigation is a software architecture for adaptive Service Level Agreement (SLA) management in value chains for service individualization. We address the problem that current SLA management is not capable to represent the full complexity of SLAs existing in real-world service industries. The problem is investigated from a functional-analytical supply chain perspective. The solution is developed from a software architecture modeling perspective according to the design science paradigm. The contribution of this paper is a software architecture that facilitates SLA negotiation and SLA-based resource management in complex agreement hierarchies. The architecture is validated in an application scenario from the airport logistics domain.

1 Introduction

The main objective of the service-oriented architecture (SOA) paradigm is to facilitate technical interoperability over organizational boundaries, representing real-world business relationships on the technical information system level. The vision of seamless SOA representation of real-world business relationships requires a technical representation of contractual agreements that exist along respective value chains. An explicit formal statement of the obligations and guarantees regarding services in a business relationship is referred to as a service level agreement (SLA) [Verma 1999, 1-5]. Thus, a SLA provides the operational definition of a service as part of a contract between a service provider and a service consumer. SOA-related SLA approaches aim at providing an abstraction of the service while facilitating measurement and monitoring of service properties agreed upon [Czajkowski et al. 2004, 6f]. Individualization of services denotes the definition and configuration of service attributes in a way in which they meet individual preferences of customers as exact as possible [Kirn et al. 2008, 4]. In real-world value chains, contractual agreements exist along the flows of goods and services. This is also true for individualized services for which individualized agreements exist. These agreements depend either directly or indirectly on other agreements along the supply chains of the value system participants (e.g. for procurement, outsourcing, etc.).

We address the problem that current SLA management is not capable to represent the full complexity of SLAs existing in real-world service industries. The reason is that it originates from high-performance computing (HPC) concerning purely technical attributes of limited expressivity which can be directly forwarded and broken-down to the lower level stages in a 1:1 manner. In this paper, we study SLA negotiation and SLA-based resource management with regards to dependencies between SLAs on different value chain levels. Therefore, we firstly investigate the production process; i.e., we place special emphasis on the process of service individualization. The problem is investigated from a functional-analytical supply chain perspective which allows for a formal analysis of factor combinations and value flows. Further, we formally analyze SLA relationships in value chains and how SLA dependencies can be addressed with SLA management, especially for determination and contracting of the dominant alternative of value creators and factor combinations for provision of individualized services.

The object of our investigation is a software architecture for adaptive SLA management. The proposed architecture is developed from a software architecture modeling perspective according to the design science paradigm [Hevner et al. 2004]. The contribution of this paper is a software architecture that facilitates SLA negotiation and SLA-based resource management in complex agreement hierarchies. The remainder of this paper is as follows: section 2 describes the basic assumptions of our research. In section 3, we propose the SLA management software architecture. In section 4, we provide a preliminary evaluation of this artifact in an application scenario of airport logistics. Section 5 discusses related work. Section 6 summarizes the result and gives an outlook on future work.

2 Basic Assumptions

2.1 Value Chain Model

This paper addresses the individualization of services; i.e., the value systems which produce the services are investigated themselves. In conformance to a supply chain perspective, value chains can be reduced to value flows between actors. Then, a value system is a directed graph $V = (A, F)$ consisting of the set of actors A and the set of value flows F. An actor is an abstraction that is delimitable from other actors which contribute to the creation of value. This contribution is carried out by production; i.e., the combination of production (input) factors and the transformation of these to products (output factors). F is a relation over the actors so that $F \subseteq A \times A$. A value flow $f \in F$ connects two actors $a_1 \in A$ and $a_2 \in A$ with $f = (a_1, a_2)$. Value flows are directed and primarily carried out from upstream actors down to the customer actor, which does not show primary value flows to other actors itself. Incoming edges are regarded as complementary input factors that are transformed into output factors [Kirn et al. 2008, 11]. Figure 3 shows a generic value chain model for service supply chains. The problem in the production of the individualized value (individualization problem) consists of both the non-determinateness of the individual customer requirements until the point in time of the demand, as well as the individuality of the requirements themselves.

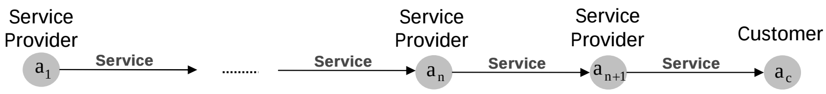

Figure 3: Generic Value Chain Model for Service Supply Chains

For the value creation it is possible to both (i) utilize actors from the own inventory and (ii) buy services from substitutional value creators as subcontractors, if the own capacities are insufficient or if the utilization of the own capacities is not economically favorable because of the cost function (e.g. step costs). Here, the provider (individualization actor) has to consider economically relevant values to determine the concrete actors for the value creation. For individualized services for which SLAs exist, this implies that the individual requirements of the customer determine the requirements to SLAs that have to be established on upstream value chain levels. Due to the non-determinateness of the requirements of needed resources and SLAs to be established on upstream value chain levels until the moment of value demand, the value flow has to be adapted. The capability of adaptation (i.e., the adaptation potential) is denoted as adaptivity. For software support of value chain adaptivity, the structure of the value chain has to be mapped to the level of technical information systems. This is done with SLAs, which describe the contractual agreements of actors involved in value creation.

2.2 SLA Model

Abstracted from technical details, a SOA-related (e.g. WS-Agreement [Open Grid Forum 2007]) SLA contains information about (i) the involved parties *A*, (ii) the service definitions *S*, and (iii) guarantees *G* assured in an agreement. According to the supply chain perspective, the parties involved in agreements are equivalent to the set of actors *A; i.e.*, $A \subset C$. As the set of SLAs *C* can also be mapped to the set of value flows *F*, *C* defines the value chain, $\forall C' \subseteq C \exists f : \mathrm{Pot}(C') \to \mathrm{Pot}(V)$. A set of observable service properties P_S is part of the service definitions *S* ($P_S \subset S \subset C$). These properties are mapped to assurances on service quality as part of the set of guarantees $G \subset C$, including qualifying conditions on external factors. These mappings are often referred to as service level objectives (SLOs). Further, *G* contains all economic values of the agreements. Currently, SLAs are mostly defined using XML (e.g. WS-Agreement). It is unfeasible to collect and standardize all possible service parameters, service metrics, and economic values, as they are assumed to be domain dependant [Open Grid Forum 2007, 5-9]. Hence, the dynamic interpretation of SLAs plays an important role. It can be achieved by using decoupled intelligence in SLA management, such as semantic reasoning techniques and intelligent agents. When SLA definitions are mapped into a knowledge base (as they are in our case), standard reasoning mechanisms can be used to interpret the SLA definitions properly, relate them to different service parameters and metrics, and finally make decisions based on a sound logical basis.

3 Adaptive SLA Management Architecture

In this section, we propose a software architecture that addresses SLA negotiation and SLA-based resource management with regards to dependencies between SLAs of different value creation levels. We suggest a goal-driven approach to represent the different goals of the actors on different levels of the value chain. This allows extending the solution space for actions to reach the desired goals. The solution space is represented and supported by a knowledge base, containing semantic descriptions of capabilities, goals, tasks, and offers. Decisions needed to manage the value chains such as matching, ranking and selecting offers are supported by reasoning in this knowledge base. Further, we propose an agentification of WS resources; i.e., WS resources are represented by agents. This provides increased flexibility of resource management and the utilization of multiagent coordination mechanism, especially market-based coordination and multiagent negotiation, without central control.

3.1 Goal-driven System Behavior

The belief-desire-intention (BDI) architecture approach [Wooldridge 2001] is a model for describing rational software agents - agents that reason, based on beliefs, which action to perform to reach given goals. That is, the BDI architecture facilitates goal-driven system behavior. The model consists of the following concepts: *beliefs* capture informational attitudes realized as a data structure containing current *facts* about the world. *Desires* capture the motivational attitudes realized as *goals* that represent the concrete motivation; i.e., desires capture a set of goals to be realized. *Intentions* capture the deliberative attitudes realized by reasoning to select appropriate *actions* to achieve given goals or to react to particular situations. A BDI agent is equipped with sensors to assist it on its environmental awareness, and effectors to impact the environment by actions. A reasoning mechanism between the sensors' input and the effectors' output deduces the necessary actions for achieving the agent's goals. The agent acquires new beliefs in response to changes in the environment and through the actions that it performs as it carries out its intentions [Wooldridge 2001]. Thus, the BDI agents allow reasoning regarding decisions to determine which, possibly conflicting, business goals can be achieved and how the agent is going to achieve these goals. For example, for an agent representing a resource of our case, beliefs correspond to the state, capabilities, and SLAs of the resource; desires represent the business goals of the resource provider, while intentions result from a collection of possible decision mechanisms to select and execute requests to use the resource. In addition, the BDI concept has been integrated with semantics: the agent's beliefs, stored in the agent's beliefbase, are completely based on semantic data. Further, semantic reasoning is applied to derive new knowledge - especially required actions to reach goals - based on the semantic beliefs. Conceptual definitions of SLA parameters, metrics, and economic values as well as resource characteristics are given in an OWL DL ontology [W3C 2004]. New data arriving to the agents are inserted into the knowledge base, which is automatically enriched using DL reasoning. Agents can then retrieve the results of reasoning via the beliefs. This provides essential support towards the targeted technical interoperability over organizational boundaries, representing real-world business relationships.

3.2 Adaptive Web Service Coordination

To allow an adaptive coordination of Web services (WSs), we propose an approach for coupling of WS and agent technology in a head body architecture [Haugeneder/Steiner 1994]. The head body paradigm implies a conceptual separation of a software agent into two parts - head and body. The agent's head is used for interactions with other agents being member of the agent society. Interacting agents form multiagent systems. This includes reasoning about interactions such as participating in cooperative processes for problem solving. The body is encapsulating any other (domain) functionality of an agent [Haugeneder/Steiner 1994]. The head body paradigm is used in the approach shown in Figure 4. WS resources that are represented by agents are part of the body. The agent's core capabilities are implemented in the head; i.e., interactions and especially coordination with other agents in the agent society. The agent has full control over the WS and communicates with it via WS sensor and WS effector. On the conceptual level, agent-to-agent communication takes place using FIPA communication standards. On the technical level, agent-to-agent communication is based on WS technologies and standards. Therefore, the agents can be used in existing WS infrastructures and systems. The presented approach of coupling WS and agents allows the utilization of multiagent coordination protocols for WS coordination in existing infrastructures. A WS which is represented by an agent can transparently be invoked by other WSs, respectively clients. The agent can evaluate the invocation requests and can reason if an invocation of the encapsulated WS is in accordance to its own goals. If the invocation request is opposed to the goals, the agent can intercept the invocation and the encapsulated WS is not invoked. Further, the agents can pro-actively work towards the goals; e.g. maximizing revenue for encapsulated resources by establishing SLAs.

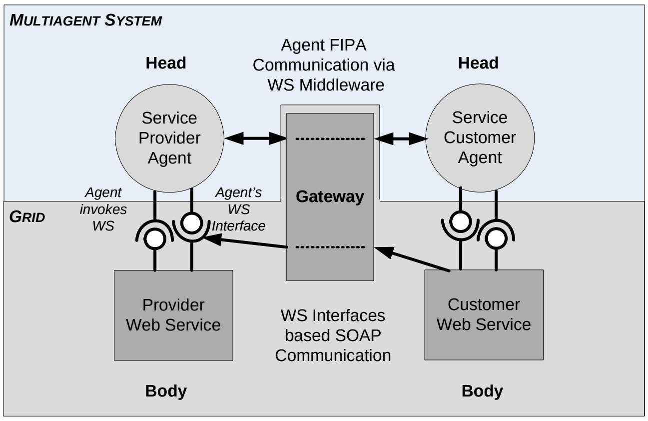

Figure 4: Head Body Architecture

3.3 SLA-Management Architecture

With regard to the targeted cross-organizational interoperability and adaptivity in heterogeneous distributed environments, we propose a combination of WS and goal-driven multiagent system (MAS). WS resources are represented by agents in accordance to section 3.2; i.e., the coordination of the resources takes place in the MAS. Thus, the system is able to perform processes which have not been defined a priori without central orchestration or control, as the agents can react goal-driven and autonomously in unforeseen situations. Thus, the advantages of both multi-agent as well as the WS technologies can be utilized. The overall architecture is shown in Figure 5: resources are encapsulated by resource agents which locally manage the execution of contracted tasks. The resource agents also offer their services to the dispatcher agents of their service provider. The dispatcher agents respond to search requests by customers and make offers accordingly; i.e., they provide resource management capabilities on a higher abstraction level than the resource agents and provide an interface for the customer. A service provider can obviously have several resource agents. In addition, several dispatcher agents can be used to avoid bottlenecks and single points of failure. Received offers from both resource agents as well as customers are stored in the business partner candidate set generator (CSG).

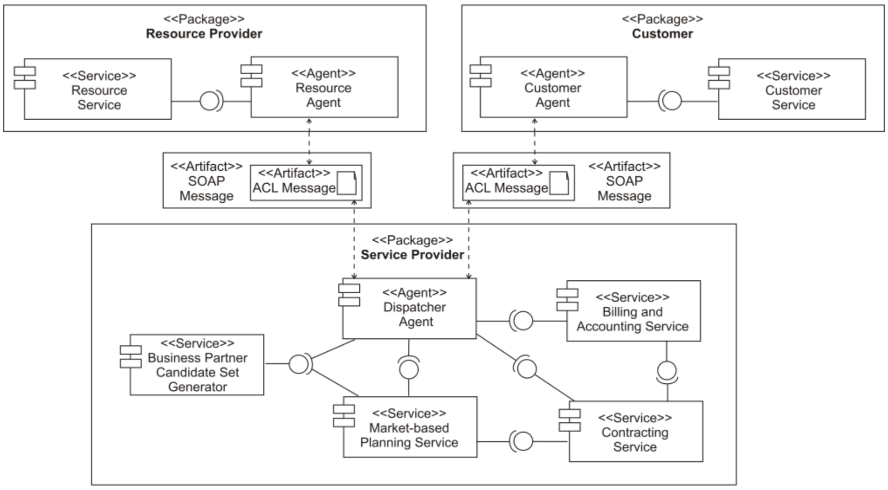

Figure 5: SLA-Management Architecture

The dispatcher agents can also request offers (bids) from other service providers, e.g. for subcontracting. Reverse (procurement) multi-attribute combinatorial auctions are applied for resource allocation by the dispatcher agents: once the bids have been collected and rated by the dispatcher agent in accordance to its goals, the market-based planning service is invoked to generate schedules based on economic values. The collected bids will be retrieved from the CSG. The offers that have been collected constitute binding bids by the participants. Thus, an allocation results in binding agreements. The technical representations of these agreements are generated by the contracting services, which are informed about allocations along with the participants. Qualitative and quantitative information about the delivery of services, respectively service quality, are collected by the dispatcher agent. It can then reason about the collected information to detect and anticipate SLA violations and react accordingly; e.g. by re-allocating certain tasks to other resources. The contracting services inform the billing and accounting service about resulting payment obligations which will inform the contracting services about fulfillments of contracted payment obligations accordingly.

4 Preliminary Evaluation

In this section, we present a scenario-based validation of the proposed software artifact as a first step towards a rigorous evaluation. The evaluation of artifacts by means of application scenarios in accordance to the design science paradigm belongs to the descriptive evaluation methods. Detailed scenarios are constructed around artifacts to demonstrate their utility [Hevner 2004, 18]. This work validates the presented artifact by means of an application scenario from the airport logistics domain. The airport management value chain produces individual services for the dispatching of aircrafts at airports.

4.1 Scenario

The physical services of ground handling at airports are represented in a WS-based system. The contracts of the organizations are described with technical SLAs. The airport management value chain includes airlines as customers as well as actors from luggage, passenger and aircraft related service providers. Our investigations are exemplarily limited to passenger transports on the ground from and to aircrafts (bus transports). The individual requirements of the customer (airline) include the number of passengers, the parking position of the aircraft, planned and actual arrival and departure time, etc. The temporal dependencies of aircraft dispatching services are mapped to dependencies between SLAs. These are considered during determination and contracting of the concrete actors for value creation. Figure 6 shows the actors involved in the value creation process: airlines constitute the customers in the investigated value chain section. They determine the individual requirements for the dispatching of aircrafts. The air traffic control provides information about concrete arrival and departure times. The dispatchers coordinate the service ‘passenger transportation on the ground’ within the airline service provider (ASP) organization. In the same organization, busses provide ‘bus trip’ services.

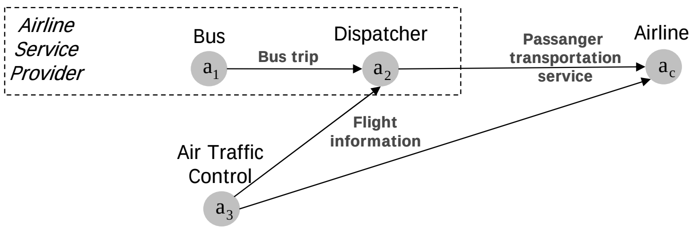

Figure 6: Value Chain Model Passenger Transport

4.2 Experiment

We assume that a dispatcher $a_{2,1}$ has established a SLA c_1 with an airline $a_{c,1}$ for the dispatching of a concrete aircraft. Internally, dispatcher $a_{2,1}$ has planned to provide the service contracted in c_1 with the bus $a_{1,1,1.}$ Therefore, it has established an internal SLA c_2. c_1 contains a guarantee $g_{1,1}$ that the dispatching of the aircraft has to be finished within ten minutes, or a penalty has to be paid. c_2 contains a service property $p_{2,1}$, stating that the number of passengers will be below 50 (capacity of bus $a_{1,1,1}$). In addition, the ASP of dispatcher $a_{2,1}$ owns a second bus resource $a_{1,1,2}$. There are also two other dispatchers $a_{2,2}$ and $a_{2,3}$ with a single bus resource each, $a_{1,2,1}$ and $a_{1,3,1}$. The resource agent's goals are to maximize the revenues by establishing SLAs for resource utilization. The dispatcher agent's goals primarily include the fulfillment of established SLAs and maximization of the revenues for their ASP by establishing SLAs for available resources. When the aircraft arrives, there is a mismatch of the internally contracted service property $p_{2,1}$ and the actual number of passengers which is assumed to be 60. To reach the goal of fulfilling the SLA c_1, dispatcher $a_{2,1}$ has to contract additional resources. Therefore, it will send a call for proposals (CFP) to its resources, including a description of the task to be executed. The bus agent representing $a_{1,1,2}$ responds with a proposal to perform the task in accordance to its goal of maximizing revenues for resource utilization, which we assume to be accepted by $a_{2,1}$. Before the execution takes place, bus $a_{1,1,1}$ is malfunctioning and thus unable to perform the task contracted in c_2. To reach the goal of fulfilling the SLA, dispatcher $a_{2,1}$ has to outsource the task to another dispatcher, as local resources are insufficient to provide the requested service and service level, as a sequential execution of both tasks by the remaining bus would violate the guarantee $g_{1,1}$. Thus, it will send a CFP to $a_{2,2}$ and $a_{2,3}$. $a_{2,2}$ offers to perform the task for 100 monetary units within 15 minutes while $a_{2,3}$ proposes the execution within 10 minutes for 150 monetary units. The internal interactions are performed in accordance to the internal re-planning of $a_{2,1}$, taking already established SLAs into account. For these SLAs, a direct dependency to the SLA with the airline c_1 exists, and there now exist conflicting goals for allocating the task to a dispatcher, as the selection of $a_{2,2}$ costs less but would imply a penalty, while the selection of $a_{2,3}$ is more expensive. The ability to determine the appropriate action via its reasoning mechanism in accordance to its goals allows $a_{2,1}$ to select the dominant alternative. The test case has been implemented using Jade as the agent platform, Jadex as the core BDI implementation on top of Jade, Jena as knowledge base and reasoning engine, and Apache CXF as WS access toolkit.

5 Related Work

Mass customization (MC) is a competition strategy which targets the production of individual products under conditions of mass production [Piller 2006]. A discussion of the effects of this strategy on information systems is given in [Dietrich et al. 2006]. Research works that focus on MC information systems (e.g. [Dietrich et al. 2007]) explicitly take product data models for product descriptions into account. However, these models mainly target material goods and do not allow a representation of contracted service levels and economic values. Approaches that focus on economic resource allocation in SOA (e.g. [Buyya et al. 2002], [Schnizler et al. 2008]) mainly consider standardized resources only. [Eymann et al. 2006] consider both a centralized market for standardized resources as well as a decentralized mechanism for an application service market. Requirements and abstract design considerations for a market-based resource allocation infrastructure for WS resources are given in [Karaenke et al. 2008]. However, these approaches do not include details about distributed lower level resource management or SLA management. An evaluation of technical SLA representation approaches and standards from an e-business transaction perspective can be found in [Karaenke/Kirn 2007]. An approach for SLA re-negotiation of parts of an established agreement is presented in [Modica et al. 2007]. The coupling of agents and WS resources in a similar approach is investigated in [Negri et al. 2005], though the authors remain on a very high level of abstraction.

6 Conclusion

This paper proposes a software architecture that facilitates SLA negotiation and SLA-based resource management in complex agreement hierarchies. The WS basis enables cross-organizational interoperability in heterogeneous distributed environments on the technical access level as well as integration and interoperability with existing service-oriented systems. A key aspect of the proposal is the mapping of real-world service contracts to the technical representation in terms of semantically annotated SLA definitions. This enables software agents to infer knowledge about necessary actions that have to be performed to provide individually contracted services; i.e. SLA-based resource management. The utilization of explicit semantics further facilitates the interoperability on business level by incorporation of domain knowledge in all phases of the service life cycle. The strong coupling of WS resources and multiagent technology allows a seamless utilization of multiagent coordination and negotiation approaches for SLA management in SOA. Further, the adoption of the BDI multiagent paradigm fosters a business goal-driven behavior of the system, breaking down strategic goals to operational tactics via semantic reasoning, enabling automatic support for decisions.

We have shown first evidence that the architecture provides delivery of services in an open, dynamic, uncertain environment without centralized control. It allows the required ad hoc re-planning and contracting of the required resources, honoring the dependencies of SLAs over multiple value chain levels, forming individual virtual organizations at run-time. Future research has to further underpin the utility of the artifact in simulations of advanced scenarios and for different multiagent coordination and negotiation mechanisms. The contribution of this work is a flexible and scalable architecture which facilitates interoperability in complex dynamic heterogeneous distributed environments; i.e., it enables the required economic adaptivity of the value chain for service individualization.

Acknowledgements

This work has been supported by the BREIN project (http://www. gridsforbusiness.eu) and has been partly funded by the European Commission under contract FP6-034556.

References

Buyya, R., Abramson, D., Giddy, J., Stockinger, H., Economic Models for Resource Management and Scheduling in Grid Computing, in: Concurrency and Computation: Practice and Experience, 14 (2002), pp. 1507-1542.

Czajkowski, K., Foster, I., Kesselman, C., Grids in Context, in: Foster, I., Kesselman, C. (Eds.), The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, San Francisco (CA) 2004, pp. 259-283.

Dietrich, A.J., Kirn, S., Sugumaran, V., A Service-oriented Architecture for Mass Customization - A Shoe Industry Case Study, in: IEEE Transactions on Engineering Management, 54 (2007) 1, pp. 190-204.

Dietrich, A.J., Kirn, S., Timm, I.J., Implications of Mass Customization on Business Information Systems. in: International Journal of Mass Customization, 1 (2006) 2-3, pp. 218-236.

Eymann, T., Neumann, D., Reinicke, M., Schnizler, B., Streitberger, W., Veit, D., On the Design of a Two-Tiered Grid Market Structure, in: Proceedings Multikonferenz Wirtschaftsinformatik (MKWI'06), Passau 2006.

Haugeneder, H., Steiner, D., Ein Mehragentenansatz zur Unterstützung kooperativer Arbeit, in: Hasenkamp, U., Kirn, S., Syring, M. (Eds.), CSCW - Computer Supported Cooperative Work, Addison Wesley, Bonn etc. 1994, pp. 203-229.

Hevner A.R., March, S.T., Park, J., Ram, S., Design Science in Information Systems Research, in: MIS Quarterly, 28 (2004) 1, pp. 75-105.

Karaenke, P., Bieser, T., Schuele, M., Kirn, S., Towards a Market-Centric OGSA-Compliant Architecture Model, in: Proceedings Multikonferenz Wirtschaftsinformatik (MKWI'08), Garching, 2008.

Karaenke, P., Kirn, S., Service Level Agreements: An Evaluation from a Business Application Perspective, in: Proceedings eChallenges e-2007 Conference, 24-26 October, The Hague 2007.

Kirn, S., Anhalt, C., Bieser, T., Jacob, A., Klein, A., Leukel, J., Individualisierung von Sachgütern und Dienstleistungen durch Adaptivität von Wertschöpfungssystem in Raum, Zeit und Ökonomie, in: Kirn, S. (Ed.), Individualization Engineering, Cuvillier, Göttingen 2008, pp. 3-60.

Modica, G.D., Regalbuto, V., Tomarchio, O., Vita, L., Dynamic Re-negotiations of SLA in Service Composition Scenarios, in: Proceedings 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007, pp. 359-366.

Negri, A., Poggi, A., Tomaiuolo, M., Intelligent Task Composition and Allocation through Agents, in: Proceedings 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05), IEEE 2005, pp. 255-260.

Open Grid Forum (OGF), Web Services Agreement Specification (WS-Agreement), OGF Proposed Recommendation GFD.107, 2007.

Piller, F.T., Mass Customization: Ein wettbewerbsstrategisches Konzept im Informationszeitalter, 4th ed., Deutscher Universitätsverlag, Wiesbaden 2006.

Schnizler, B., Neumann, D., Veit, D., Weinhardt, C., Trading Grid Services - A Multi-Attribute Combinatorial Approach, in: European Journal of Operational Research, 187 (2008) 3, pp. 943-961.

Verma, D., Supporting Service Level Agreements on IP Networks, Macmillan Technical Publishing, New York 1999.

W3C, Web Ontology Language (OWL), 2004, http://www.w3.org/2004/OWL/, 31.10.2008.

Wooldridge, M., Intelligent Agents, in: Weiss, G. (Ed.), Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, 3rd edition, MIT Press, Cambridge (MA) 2001, pp. 27-77.