Agent-Supported Collaboration and Interoperability for Networked Enterprises

Ingo Zinnikus, Xiaoqi Cao, Klaus Fischer

HAL Id: hal-01572108

https://hal.inria.fr/hal-01572108

Submitted on 4 Aug 2017

Agent-Supported Collaboration and Interoperability for Networked Enterprises

Ingo Zinnikus, Xiaoqi Cao, and Klaus Fischer [*]

German Research Center for Artificial Intelligence, Saarbrücken, Germany,
`Ingo.Zinnikus/Xiaoqi.Cao/Klaus.Fischer@dfki.de`,
WWW home page: `http://www.dfki.de/~kuf`

Abstract. The paper presents an agent-supported framework for improving solutions for enterprise interoperability and enterprise collaboration. We present the context of COIN in the European research area and explain the basic approach and system architecture COIN is aiming at. Special emphasis is put on how agents can support enterprise interoperability as well as enterprise collaboration services. The framework adopts a modeling approach for the description and execution of business processes. With this a system engineer can describe the interaction protocols that should be used at an intuitive level and transform the model down to executable code. Private partner processes can be integrated using a mapping approach to bridge technical interoperability gaps.
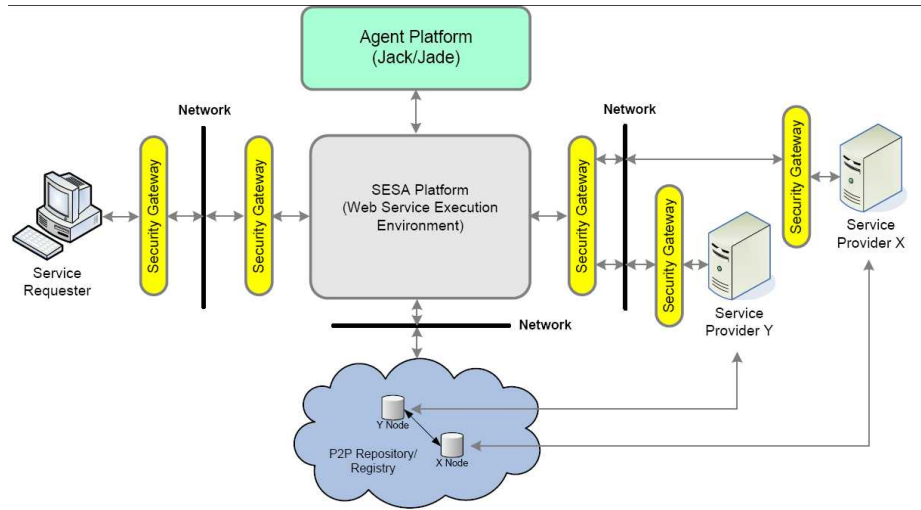
1 Motivation

According to the COIN vision, by 2020 enterprise collaboration and interoperability services will become an invisible, pervasive and self-adaptive knowledge and business utility at disposal of networked enterprises from any industrial sector and domain in order to rapidly set-up, efficiently manage and effectively operate different forms of business collaborations, from the most traditionally supply chains to the most advanced and dynamic business ecosystems [1].

In the area of business process modeling and enactment, progress has been made to bridge the notorious gap between business and IT. Still, the business perspective itself is split into a *value perspective* with regards to strategical considerations and a *process perspective* which concentrates on conceptual modeling of business activities. The latter perspective has received exceptional attention in research and commercial tool development. Recent empirical studies among researchers and practitioners confirmed critical areas of concern: *standardization of modeling approaches* and *model-driven process execution* are considered important [2]. Model-driven process design and execution could improve 'plug-and-play' functionality and rapid prototyping for testing in order to decrease

Fig. 1. The COIN Service Provisioning Platform

time to market. These tendencies are valid not only for intra-business process execution but also for modeling inter-organizational processes.

The gap between business and IT perpective was already a focus in the ATHENA project funded by the European Commission (IP, FP6). A core idea in the ATHENA project was to bring together different approaches and to combine them into a new framework: a modelling approach for designing collaborative processes, a model-driven development framework for SOAs and, among others, an agent-based approach for flexible execution.

The COIN Project continues this line of research. The main objectives of COIN are to design and develop a pervasive, adaptive service platform to host baseline and innovative COIN services for enterprise interoperability (EI) and enterprise collaboration (EC) and make them available under innovative on-demand, utility-oriented business models to enterprises (and SMEs in particular) for running their business in a secure, reliable and efficient way.

One of the main ideas of COIN is to unify EC and EI services in a unifying architecture. COIN adds a service provisioning platform for registering and executing (semantically annotated) services as baseline to the collaboration platform. The core of this service provisioning platform is a generic service platform (GSP) which is built on top of the basic infrastructure of the Internet and the Web. Innovative services to support EC and EI are designed as an extension to the GSP. These innovative services can access Web services registered in the GSP or can be built directly with traditional Internet technology. Proof-of-concept tools to support end users in the execution of collaborative processes in networked enterprises are developed on top of the COIN platform.

The COIN service provisioning platform (see Figure 1) is based on the WSMO [3] framework and the corresponding set of languages and specifications for de-

scribing ontologies, services, goals and mediators. Agent technologies are used in the GSP for intelligent service coordination and negotiation.

This paper describes an agent-based modeling framework with embedded support for solving interoperability issues. The rest of the paper is organized as follows: we describe the approach for modeling collaborative processes in Section 2 and describe the agent-based modeling environment in Section 3. Related work is discussed in Section 4. Finally, Section 5 concludes.
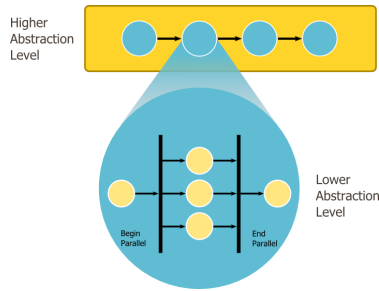
2 Collaborative Processes

Following the main objectives of COIN as described in the last section, the design and execution of collaborative business processes is in the main focus of COIN. Figure 2 presents COIN's approach to collaborative business process design. The situation in the figure is simplified to the collaboration of two partners but of course COIN aims at the collaboration of any number of partners. In such a setting each partner has its private process from which so-called *view* (or *public*) processes are derived. On the other hand, a collaborative process is defined where the collaborative process can take advantage of the processes and services specified in the view processes. The approach seems to be centralized, however, this is only a conceptual description. It is actually not prescribed how the collaborative process is developed nor is it prescribed how the resulting collaborative process is eventually executed (cf. [4]). However, to assume that the collaborative process is executed by any of the participating parties or by a third party is a straightforward solution once the collaborative process has been defined.
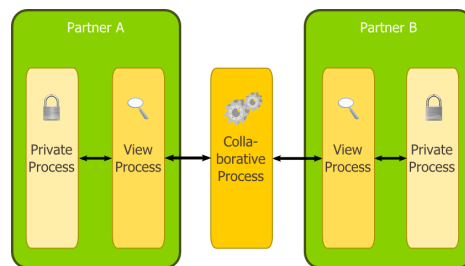
Fig. 2. COIN's Approach to Collaborative Process Design and Execution

The view processes mediate between the private processes that are already available in the local environment of the individual partners and the collaborative process. In the design of collaborative processes already existing view processes might be used or new view processes might be derived by the requirements that are specified in the collaborative process.

Fig. 3. Reducing Complexity in Collaborative Processes

Complex interactions between the partners in collaborative environments imply a number of interoperability problems occurring in cross-organizational scenarios which have to be solved:

– changing the protocol and integration of a new partner should be possible in a rapid manner (scalability)
– the execution of the message exchange should be flexible, i.e. in case a partner is unavailable or busy, the protocol should nevertheless proceed
– the different partners (may) expect different atomic protocol steps (service granularity)
– the partners expect and/or provide differing data structures

The collaborative processes are specified on a technology-independent level (using the Business Process Modeling Notation, BPMN) and transformed to a platform-independent level so that refinements and modifications in the interaction of the partners can be made on the respective levels and code generated automatically. Flexibility is achieved by applying a BDI (belief-desire-intention) agent-based approach for the actual execution layer. BDI agents provide flexible behavior for exception-handling in a natural way (compared to e.g. BPEL where specifying code for faults often leads to complicated code). The problem of different service granularities is envisaged by specifying a collaborative protocol which allows adaptation to different service granularities. Finally, the mediation of the data is tackled with using transformations which are specified at design-time and executed at run-time by transforming the exchanged messages based on the design-time transformations.

In the top-down approach a business process engineer starts of with the definition of a high-level process in BPMN. When she is more or less satisfied with the high-level definition she starts to work on the grounding of the collaborative process. The process model is transformed into an agent model which captures the structure of the high-level process. On the agent model, *refinement* of processes and interactions is done, i.e. adding concrete data structures for storing instance data and fleshing out the messaging within the public process. For the integration of the private processes, agents invoke Web services described by WSDL interfaces.
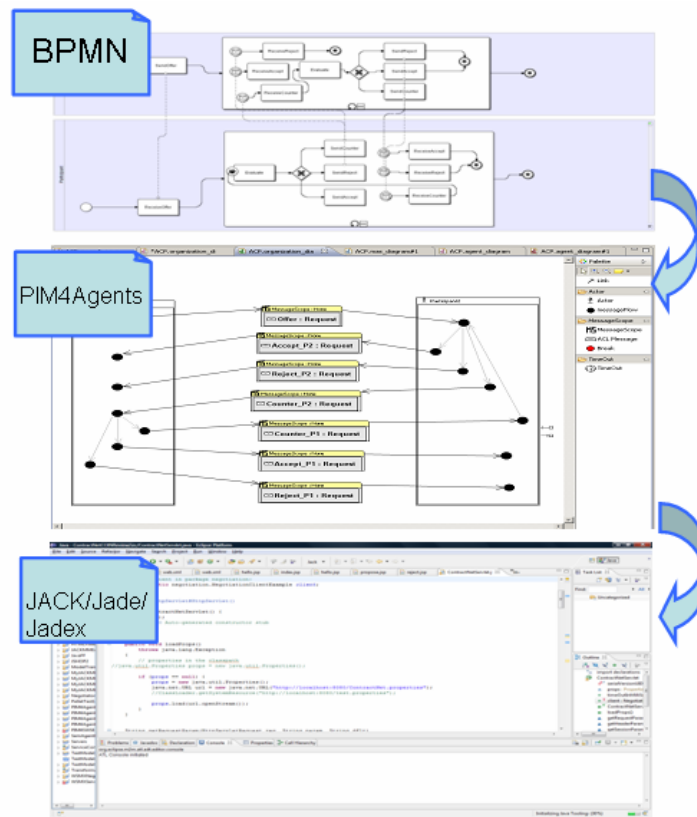
Fig. 4. Transformations for executable collaborative processes

3   Agent-based Modeling of Collaborative Processes

The graphical agent modeling framework developed at DFKI [5] directly supports service interaction and complex workflow patterns (for a more complete discussion of supported workflow and service interaction patterns see [6]). Enriched with concepts for supporting Web service invocation, it allows modeling and executing collaborative processes. The framework is expressive enough to capture the most important workflow and service interaction patterns identified in research on workflow modeling [7, 8].

The overall procedure for specifying collaborations can be summarized as follows (see Figure 4):

- In a first step, the domain roles and participants are specified, using BPMN models for collaborative processes.
- Next, the interaction protocol(s) between the roles are derived by transformation and refinement on PIM level.
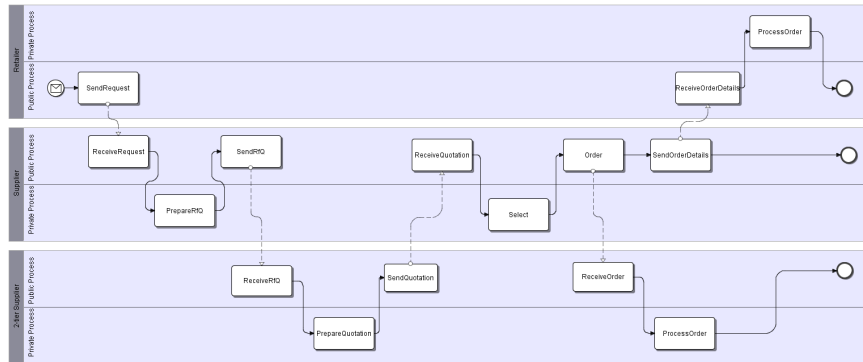
Fig. 5. Abstract collaborative process with lanes representing public and private processes

- The local behavior of the participants complementing the protocol description is defined in plans. The interoperability gap between the global data and the private processes (services) is bridged with a mapping on the PIM level.
- Executable code is generated from the model and deployed in a Web-based COIN environment to allow humans to use the modeled process.

With this approach, it is possible to model complex processes and generate executable code for agent-based platforms.

As an application example for collaborative processes, we consider negotiations between business partners as scenario to illustrate the approach. Following the approach presented in the previous section, we conceive negotiations in COIN as examples and instances of interactions between organizations which try to reach an agreement on some issues. A concrete example is the selection of a part provider in a 3-tier manufacturing supply-chain where the acceptance of a bid is the result of a possibly multi-lateral interaction between e.g. retailer, manufacturers/suppliers and e.g. raw material vendors. Note that supply chains are only used as illustration (concerning agents in supply chains, see e.g [9]). The approach is not restricted to supply chains; a different scenario could even involve n-to-n interaction between collaborating partners. In the following, we demonstrate the application of the model-driven methodology with an example which is based on a supply-chain scenario.

3.1 Deriving the roles and participants

As a first step, the collaborative process is specified with a graphical editor for BPMN (see Figure 5). Parties involved in the collaboration are specified (derived from pools). A domain role is designed for each lane in the collaboration. These roles are performed by agents which are abstract types that can be instantiated by concrete agent instances. Each agent in the scenario uses plans which describe the procedural knowledge the agent has about the domain.
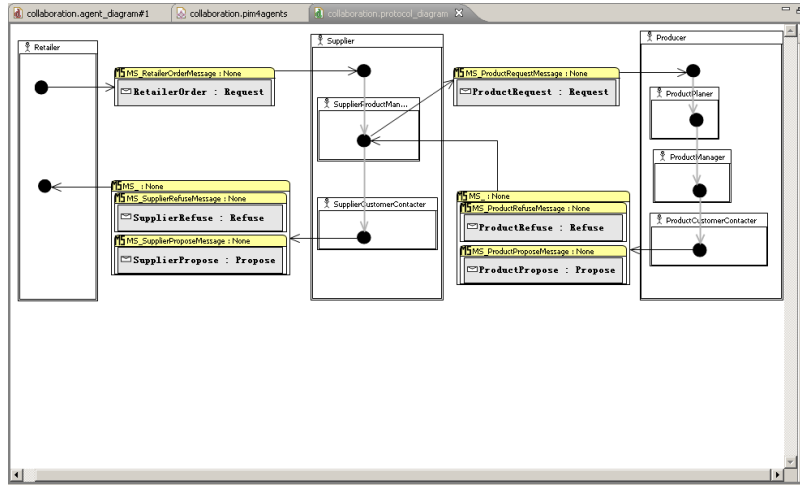
Fig. 6. Specifying the collaboration

3.2 Modeling the interaction

The next step in the methodology is to derive the interaction between the roles in the domain by transforming the abstract process into an agent interaction model. The interaction view provides the constructs necessary to specify complex protocols which abstract from the concrete behavior of the participants.

In Figure 6, the interaction protocol for the collaboration is depicted. Pool-like boxes for Retailer and n-tier suppliers indicate the activities which are related to participants. In-between the pools, the messages exchanged are defined. MessageFlows (indicated by filled circles/nodes in a pool) allow defining the control flow of the protocol at each participant site. E.g., after receiving the starting request, the retailer contacts the suppliers which reply by sending a quotation or refusing the request. Note that the criteria for the decision are not defined in the interaction view but locally in the behavior model of the participants. Furthermore, the number of suppliers is left open at design-time. This reduces the complexity of the interaction model.

3.3 Modeling the local behavior of the participants

The next step in the methodology consists in refining the local behavior of the participants corresponding to the interaction protocol. The behavior extends and complements the protocols by specifying additional information for generating executable code. In order to be consistent, the behavior has to comply with the interaction protocol.

In fact, an ongoing line of research is the automatic projection of the interaction protocol into local behavior models for the participating agents [10]. The results of the projection are plan skeletons which can then be further refined to specify e.g. decision conditions in more detail.
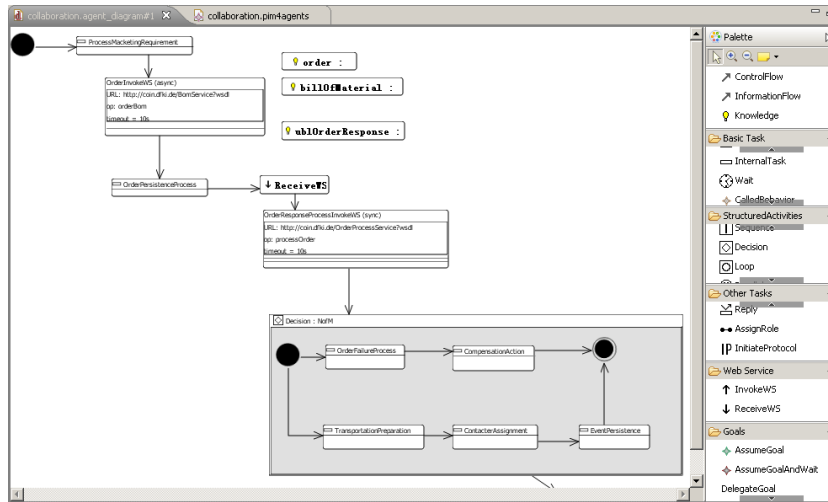
Fig. 7. Local behavior for participant

The plan for the supplier (see Figure 7) contains the detailed specification of the steps necessary for executing the supplier parts of the interaction protocol.

After receiving a request for quotation, the private process for generating a quotation is invoked. The service is exposed as a web service described in WSDL. Both synchronous and asynchronous invocation are supported. After receiving the response of the private process, the participant prepares the reply to the 2-tier supplier.

The plan for integrating the private process (service) also allows abstracting from peculiarities of the service interface (service granularity). The private process might require a complex interaction with a specific order of invocations. These local protocol issues can be encapsulated in the local behaviour of a participant. From a global perspective, the interaction appears as an atomic step.

The plan model is obviously more complicated than the interaction protocol description since the model contains the control flow as well as the information flow (data such as *order* or *billOfMaterial*). The control flow describes the sequence of tasks to be executed, whereas the information flow concerns the way data is transferred from one task to the other.

3.4 Mediating data heterogeneity

Concerning the integration of private processes, two cases can be distinguished:

(i) Integrating consortial partners: Partners define the shared process together. The common information/data model may also be defined together. Roughly speaking, two alternatives are possible, analogous to the local-as-view (LAV) vs. global-as-view (GAV) distinction in the fields of data integration (cf. [11]) and Enterprise Information Integration (cf. [12]).

In a LAV approach, the common data structure is defined independently from the local data model of each partner. Each partner then defines a (local) mapping from the common information model to the local model. The mapping in turn can be executed (at run-time) either by the consumer of the service or the partner service itself. The first solution is the one preferred by Semantic Web service descriptions, where the service provider describes the grounding to e.g. WSDL. The grounding is used by a service consumer who invokes the service. The second solution means that the service consumer always sends the same message (e.g. a SOAP message) to a partner service and does not care about the local data model. This is reasonable if specifying as well as testing the mapping is tedious and the mapping underlies many changes.

In the GAV approach, each element in the global model is defined in terms of local sources. Adding a new source (i.e. partner) affects the global model. The LAV approach is preferable for cases where the global model is stable and new partners are frequently joining (or leaving) the collaboration space.

(ii) Integrating partners external to the collaboration: For the integration of external partners service discovery as well as process and data mediation have to be realized. Service discovery using the COIN GSP can be based on the service requirements which are specified implicitly or explicitly for a service invocation task. Integrating the discovered services requires data transformations which are either provided by the service descriptions or based on a mapping to the common information model.

The data structure for the global (public) process is agreed among the partners. This means that, as long as the interfaces of the private processes are not aligned, mismatches between public and private data structure are to be expected. In order to bridge this gap, we developed a mapping tool (Figure 8) which allows to import the data structure of the private service and align it with the data of the public process. In more detail, the data structure of the service is described in XML schema which is imported and 'lifted' to Ecore format supported by the Eclipse developing environment. The mapping tool allows specifying the mediation between the Ecore format of the private process and the Ecore format of the public process. Currently, apart from one-to-one mappings, also one-to-many mappings are supported, as well as operations such as concatenation etc. For executing this mapping, an XSLT transformation is derived and used at runtime. The approach is open in the sense that other executable transformations are supported.

3.5   Generating executable agent-code

The modeling framework provides code generation for several agent-platforms. Currently, the agent platforms Jade and JACK are supported. Code generation for Jadex is under development. The specified models can be transformed e.g. into Java-based Jade code. The transformation leads to code which can be further refined if necessary, e.g. to invoke an evaluation service or to deploy the collaboration protocol in a Web-based environment to involve human users.
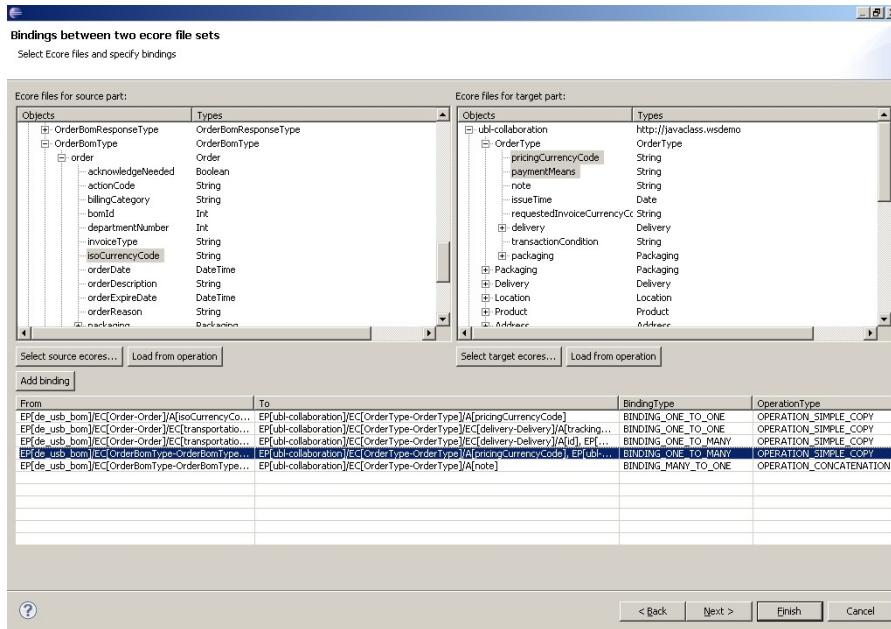
Fig. 8. Integrated tool support for defining maping on PIM level

4 Related Work

Extensions for modeling choreographies in BPMN have been proposed by [13]. The new version BPMN 2.0[1] includes notation diagrams for choreographies involving complex interactions patterns. Tool support for version 2.0 is slowly improving. However, recent empirical studies suggest that the 'core' subset of the notation used in realistic scenarios will remain rather stable and small [14]. It remains to be seen whether the 'new' elements will be used in a large scale in higher-level process models.

Besides the huge amount of literature on business process modeling, enterprise application integration and SOAs, the relation between agents and SOAs has already been investigated. [15] cover several important aspects, [16] propose the application of agents for workflows in general. [17] provide an overview of agent-based modeling approaches for enterprises. [18] describe the TROPOS methodology for a model-driven design of agent-based software systems. However, the problems related to integration of agent platforms and service-oriented architectures are out of scope for their approach. [19] map BPMN models to BDI agents but do not consider an integration of agents and Web services. [20] and [21] present a technical and conceptual integration of an agent platform and Web services. [22] integrate Web services into agent-based workflows, [23] integrate BDI agents and Web services. However, the model-driven approach

---

[1] http://www.bpmn.org/

and the strong consideration of problems related to cross-organizational settings have not been investigated in this context. Furthermore, our focus on a tight and lightweight integration of BDI-style agents fits much better to a model-driven, process-centric setting than the Web service gateway to a JADE agent platform considered by e.g. [20]. A good starting point for details on the semantic web is [24].

5   Conclusion

In this article we presented an agent-supported modeling framework for collaborative processes that is integrated with a service-oriented environment. We presented the context of COIN in the European research area and explained the basic approach and system architecture COIN is aiming at. Special emphasis was put on how a model-driven, agent-based framework can support enterprise interoperability as well as enterprise collaboration services.

References

1. COIN Consortium: COIN Integrated Project: Collaboration and Interoperability for Networked Enterprises http://www.coin-ip.eu/.
2. Indulska, M., Recker, J., Rosemann, M., Green, P.: Business Process Modeling: Current Issues and Future Challenges. In van Eck, P., Gordijn, J., Wieringa, R., eds.: CAiSE. Volume 5565 of Lecture Notes in Computer Science., Springer (2009) 501–514
3. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. Applied Ontology **1**(1) (2005) 77–106
4. Müller, J.P., Roser, S., Bauer, B.: Architectures for cross-enterprise business integration. In Sherif, H.M., ed.: Handbook of Enterprise Integration. Auerbach Publications (Taylor&Francis) (2009) 337–354
5. Warwas, S., Hahn, C.: The DSML4MAS development environment. In Sierra, C., Castelfranchi, C., Decker, K.S., Sichman, J.S., eds.: AAMAS (2), IFAAMAS (2009) 1379–1380
6. Hahn, C., Zinnikus, I.: Modeling and executing service interactions using an agent-oriented modeling language. In: Proceedings of the Forum at the CAiSE'08 Conference, Montpellier, France (June 2008) 37–40
7. Barros, A., Dumas, M., ter Hofstede, A.: Service interaction patterns. In: Proceedings of the 3rd International Conference on Business Process Management, Nancy, France, Springer Verlag (September 2005) 302–318
8. van der Aalst, W.M.P., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. Distributed and Parallel Databases **14**(3) (July 2003) 5–51
9. Chaib-draa, B., Müller, J.: Multiagent based Supply Chain Management (Studies in Computational Intelligence). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
10. Hahn, C., Zinnikus, I., Warwas, S., Fischer, K.: From agent interaction protocols to executable code: a model-driven approach. In: Proceedings of the 8th international Conference on Autonomous Agents and Multiagent Systems. Volume 2., Budapest, Hungary, Richland, SC (May 10–15 2009) 1199–1200

11. Lenzerini, M.: Data integration: a theoretical perspective. In: PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, New York, NY, USA, ACM Press (2002) 233–246

12. Halevy, A.Y., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., Rosenthal, A., Sikka, V.: Enterprise information integration: successes, challenges and controversies. In: SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (2005) 778–787

13. Decker, G., Puhlmann, F.: Extending BPMN for Modeling Complex Choreographies. In Meersman, R., Tari, Z., eds.: OTM Conferences (1). Volume 4803 of Lecture Notes in Computer Science., Springer (2007) 24–40

14. Muehlen, M.Z., Recker, J.: How much language is enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: Proceedings of the 20th international conference on Advanced Information Systems Engineering. CAiSE '08, Berlin, Heidelberg, Springer-Verlag (2008) 465–479

15. Singh, M.P., Huhns, M.N.: Service-oriented Computing — Semantic, Processes, Agents. John Wiley & Sons, Ltd. (2005)

16. Vidal, J.M., Buhler, P., Stahl, C.: Multiagent systems with workflows. Internet Computing, IEEE **8**(1) (2004) 76–82

17. Cabri, G., Leonardi, L., Puviani, M.: Service-oriented agent methodologies. In: 5th IEEE International Workshop on Agent-Based Computing for Enterprise Collaboration (ACEC-07). (2007)

18. Penserini, L., Perini, A., Susi, A., Mylopoulos, J.: From stakeholder intentions to software agent implementations. In: Proc. of the 18th Conference on Advanced Information Systems Engineerng (CAiSE'06). (June 2006) 465–479

19. Endert, H., Küster, T., Hirsch, B., Albayrak, S.: Mapping BPMN to agents: An analysis. In: First international Workshop on Agents, Web-Services, and Ontologies Integrated Methodologies (AWESOME07). (2007) 164

20. Greenwood, D., Calisti, M.: Engineering web service — agent integration. In: IEEE International Conference on Systems, Man and Cybernetics. Volume 2. (2004) 1918–1925

21. Dickinson, I., Wooldridge, M.: Agents are not (just) web services: Considering BDI agents and web services. In: AAMAS 2005 Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE). (2005)

22. Savarimuthu, B.T.R., Purvis, M., Purvis, M., Cranefield, S.: Agent-based integration of web services with workflow management systems. In: Fourth international joint conference on Autonomous agents and multiagent systems (AAMAS 05). (2005) 1345–1346

23. Bozzo, L., Mascardi, V., Ancona, D., Busetta, P.: Coows: Adaptive BDI agents meet service-oriented computing extended abstract. In: Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS05). (2005)

24. Lausen, H., Ding, Y., Stollberg, M., Fensel, D., Hernandez, R.L., Han, S.K.: Semantic web portals: state-of-the-art survey. Journal of Knowledge Management **9**(5) (2005) 40–49