AGENTS FOR MANAGING BUSINESS-TO-BUSINESS INTERACTIONS

Software Agents for Managing Business-to-Business Collaborations

Edgar Tello-Leal
Universidad Autónoma de Tamaulipas, Matamoros entre J.B. Tijerina y C. Colón, 87000, Victoria, Tamaulipas, Mexico

Omar Chiotti
INGAR-CONICET, Avellaneda 3657, S3002GJC, Santa Fe, Argentina

Pablo D. Villarreal
CIDISI, Universidad Tecnológica Nacional-FRSF, Lavaise 610, S3004EWB, Santa Fe, Argentina

Keywords: Software agents, Business-to-Business, Collaborative business process, Model-driven architecture.

Abstract: Current market opportunities and the growth of new Internet technologies encourage organizations to dynamically establish Business-to-Business (B2B) collaborations. B2B interactions are carried out by executing collaborative business processes among the parties. In this work we propose B2B collaboration agents for managing B2B interactions that allow organizations to dynamically establish collaborations and execute collaborative processes with their partners. The planning and execution of the actions of the agents that execute collaborative processes are driven by a Petri Net engine embedded in these agents. The role an organization fulfills in a collaborative process is represented by a high-level Petri Net model which is used to drive the behavior of the B2B collaboration agents representing the organization. Moreover, interaction protocols representing collaborative processes are executed by these agents without the need for protocols defined at design-time. Finally, an implementation of the B2B agents is presented.

1 INTRODUCTION

The growth of new Internet technologies has led to the opportunity of much greater connectivity among organizations. Then, organizations are focusing on the setting up of Business-to-Business collaborations with their business partners in order to manage inter-organizational collaborations and improve their performance and competitiveness. A B2B collaboration entails a process-oriented integration among heterogeneous and autonomous organizations which must be achieved at a business level and at a technological level (Villarreal et al., 2007b). Inter-organizational collaborations are carried out through the execution of collaborative business processes. A collaborative business process defines the global view of the interactions among enterprises to achieve common business goals (Villarreal et al., 2007b; Roser and Bauer, 2005). The design and implementation of collaborative business processes (CBPs) implies new challenges, such as participants autonomy, decentralized management, peer-to-peer interactions, negotiation, and alignment between the business solution and the technological solution (Villarreal et al., 2006b; Weske, 2007). Therefore, the software applications must be developed that can interoperate effectively in this new distributed, heterogeneous, and sometimes unreliable environment.

Software agent technology is seen as a potentially robust and scalable approach to meet this challenge. Since the features of software agents such as autonomy, heterogeneity, decentralization, coordination and social interactions are also desirable for organizations involved in B2B collaborations (Villarreal et al., 2007b), for which the use of this technology can be considered as appropriate to be used in this domain. The use of software agents to execute CBPs helps to improve process integration, interoperability,

Tello-Leal E., Chiotti O. and D. Villarreal P..
AGENTS FOR MANAGING BUSINESS-TO-BUSINESS INTERACTIONS - Software Agents for Managing Business-to-Business Collaborations.
DOI: 10.5220/0003180002380244
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 238-244
ISBN: 978-989-8425-41-6

reusability and adaptability (Trappey et al., 2009; Zinnikus et al., 2008; Guo et al., 2005).

Therefore, in this work we propose B2B collaboration agents for managing B2B interactions that allow organizations to establish dynamic collaborations and execute collaborative processes with their partners. Interaction protocols are better intuitive models of how agents will interact through a message-based communication. The interaction protocols representing collaborative processes are executed by these agents without the need for protocols defined at design-time. The planning and execution of the actions of these agents are driven by a Petri Net engine embedded in them. The role an organization fulfills in a collaborative process is represented by a high-level Petri Net model which is used to drive the behavior of the B2B collaboration agent representing the organization. Our proposal is consistent and integrated with the model-driven development approach for CBPs and B2B information systems proposed in (Villarreal et al., 2006a; Villarreal et al., 2007b). The Petri Net models of the agents' behaviors are derived from collaborative process models which describe business interaction protocols defined with the UML Profile for Collaborative Business Processes based on Interaction Protocols (UPColBPIP) language (Villarreal et al., 2006a; Villarreal et al., 2007b).

The paper is organized as follows. Section 2 describes the architecture and agents that compose the proposed B2B collaboration approach. Section 3 describes the MDA-based method for generate Petri Net models. Section 4 describes the implementation of B2B collaboration agents. Section 5 presents conclusions and future work.

2 B2B COLLABORATION BASED ON SOFTWARE AGENTS

In this section we propose a B2B Collaboration Agents architecture in order to enable organizations to establish collaborations in a dynamic way and execute CBPs. Since the communication among software agents is based on interaction protocols, a straightforward implementation of UPColBPIP collaborative process models based on interaction protocols is provided by this approach. Two main functionalities are supported in the B2B collaboration agents approach. The first functionality is the management of dynamic agreements on collaborations among organizations to execute one or more CBPs. This implies that mechanisms for organizations can instantiate and execute CBPs whose models are provided by an organization or stored in a public repository. The second functionality refers to the capabilities to execute and monitor CBPs that organizations agreed to carry out.

The B2B collaboration agents architecture consists of two types of agents to support the above functionalities:

- A *Collaboration Administrator Agent* (CAAgent), which represents an organization or participant of a B2B collaboration. It is responsible for establishing communications with other CAAgents in order to agree, in a dynamic way, on the CBPs to be executed, and responsible for instantiating CPAgents, which execute CBPs (see Fig. 1).
- A *Collaborative Process Agent* (CPAgent), which executes the role an organization fulfills in a CBP, and thus, it is the responsible for the jointly execution of a CBP along with the other CPAgents from the partners involved in the CBP. Thus, at a specific moment or time, an organization can have as much CPAgents as CBPs being executed on the B2B collaboration (see Fig. 1).

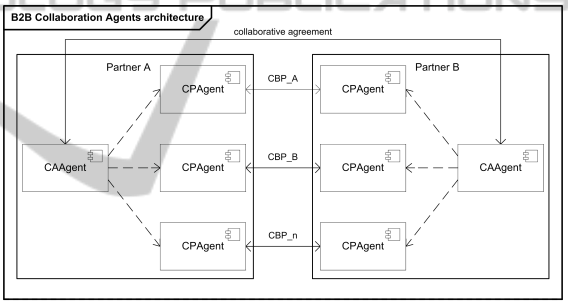

Figure 1: B2B Collaboration Agents architecture.

2.1 Collaboration Administrator Agent

CAAgents are used to create dynamic collaboration agreements and instantiate the CPAgents, which execute CBPs. When an organization wants to establish a dynamic agreement with another organization to collaborate and execute a CBP, its CAAgent is used to initiate a conversation with the CAAgent of the other organization. This conversation among agents is carried out by executing a predefined CBP *Request for Collaboration*. Figure 2 shows the UPColBPIP model that represents the interaction protocol of this CBP.

The *initiator* role is performed by the CAAgent of the organization that initiates the conversation. It sends a *request* for collaboration to *responder* role, which is performed by the CAAgent of the other organization and receives the request. This means that a CAAgent can perform the *initiator* or *responder* role according to the way that the organization engages in a collaboration agreement.

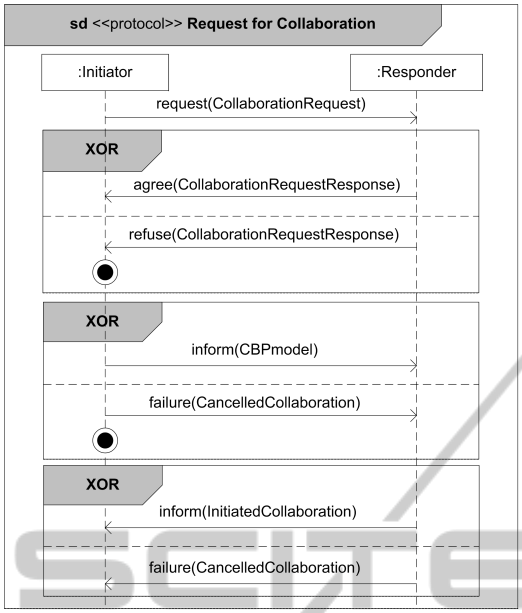

Figure 2: Request for Collaboration Interaction Protocol.

The request sent by the *initiator* conveys a *CollaborationRequest* document which contains the UPColBPIP model of the CBP that the *initiator* wants to execute with a *responder*. The *responder* evaluates the request and responds with a successful *agree* message or a *refuse* message. In the last case, it implies that the *responder* does not want to establish a dynamic collaboration agreement to execute the CBP, and the protocol finishes. If the *responder* sends an *agree* message, it waits for the process model that contains the activities that the role is going to execute in the CBP. When the *initiator* receives the *agree* message, it creates an instance of a CPAgent, and if it is successful, it sends an *inform* message with the responder's process model. Otherwise, the *initiator* sends a *failure* message indicating that it could not instantiate its CPAgent.

Once the *responder* receives the process model, it immediately instantiates a CPAgent, and if it is successful, it sends an *inform* message indicating that the collaboration was established, and the protocol finishes. Otherwise, if an error occurs in the instantiation of the CPAgent, the *responder* sends a *failure* message indicating that the collaboration was not established, and the protocol finishes. Thus, by executing this protocol CAAgents establish a dynamic agreement to execute a CBP. In case the CBP involves several organizations, the *initiator* executes the *Request for Collaboration* protocol for each organization that can fulfill the roles of the CBP. The CAAgents can also remove their CPAgents in case of the execution of a CBP needs to be interrupted.

2.2 Collaboration Process Agent

CPAgents carry out the jointly and decentralized execution of CBPs from an established dynamic agreement (which was achieved by CAAgents) among the organizations involved in the CBP. A CPAgent is responsible of performing the role an organization fulfills in a CBP. To perform the role of an organization, a CPAgent takes a process model containing the behavior of the organization's role, which is known as integration process or abstract process model (Villarreal et al., 2006a; Villarreal et al., 2007b). An integration process contains the public and private activities and logic that support the role the organization performs in the CBP. Thus, a CBP is executed by the enactment of the integration process of each organization by means of the CPAgents of the organizations.

In order to interpret and execute an integration process model by a CPAgent, this model is defined in terms of a High-Level Petri Net (PN) model, which is derived from the UPColBPIP model of the CBP to be executed. A CPAgent has an embedded *process machine* that interprets the PN model and executes the organization's role in a CBP (see Fig. 3). The activities or transitions defined in a PN model represent the actions (i.e. send a message to another agent, wait for the reception of a message from another agent, execute an internal action) that a CPAgent has to perform to support the message exchange defined in the interaction protocol of the CBP that is executed, according to the role that the agent performs in the CBP.

Therefore, the behavior of this agent is driven by the *process machine* according to a PN model and it invokes the *behavior action manager* component which is responsible for planning and executing the agent's actions derived from the transitions of the PN model. The sending or receiving of a message from another agent is carried out by the *communication manager* component of the CPAgent when the *behavior action manager* executes a message sending or receiving action. Tokens in the execution of a PN model contain the *business documents* that are exchanged between the CPAgents. Thus, *business documents* move through the organizations' CPAgents when the actions for the sending of messages are executed.

In this way, CPAgents only have generic actions (such as send or receive) defined at design-time and they do not require to have the behavior of predefined interaction protocols implemented. The transitions of the PN models execute actions. Such actions are used to activate the behavior of the CPAgent according to the type of the transition to execute. Different types of transitions (such as send message, receive message, invoke an internal process) were defined to plan and assign the behavior to be executed by the

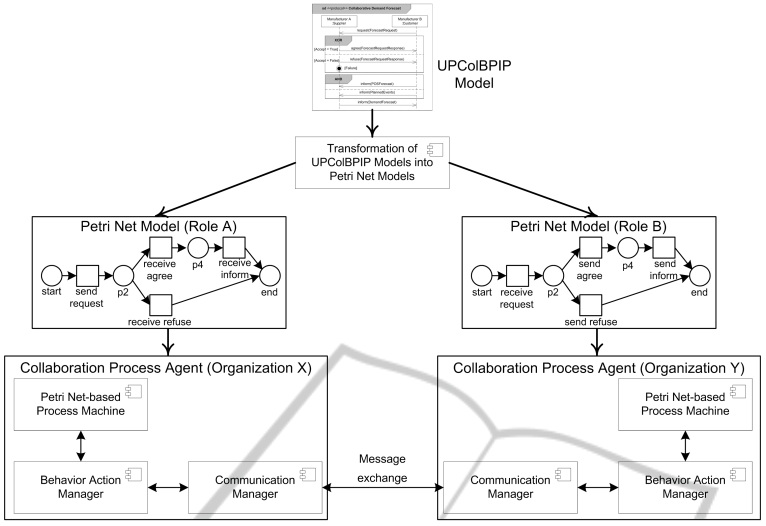

Figure 3: Collaboration Process Agents executing a CBP.

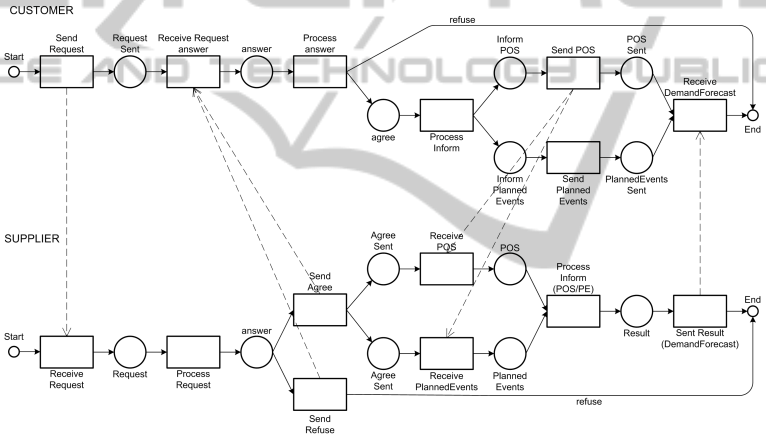

Figure 4: Interaction protocol for Collaborative Demand Forecast scenario.

agent. Tokens maintain information about the process in which they were generated (process ID) and business documents which are sent, received, or generated internally. The transitions are also allowed to have a *guard*, which is a boolean expression. When a guard is present, it must evaluate to *true* to enable the transition, otherwise the transition is disabled and cannot occur (Jensen and Kristensen, 2009). Hence, a guard puts an extra constraint on the enabling of a transition. Therefore, the concrete actions (speech acts and business documents) of these agents are obtained at run-time according to the logic defined in PN models. This enables CPAgents to execute any interaction protocol representing the CBP that organizations want to execute as part of a dynamic collaboration agreement.

For each individual role there is a separate Petri net. The collection of individual Petri nets associated with all the relevant roles represents the entire interaction protocol. Figure 4 shows the PN that represents the interaction protocol that the CPAgents have to execute the *Collaborative Demand Forecast* collaborative process (Villarreal et al., 2010). The CPAgents involved in this collaborative process are the *customer CPAgent* and the *supplier CPAgent*. The interaction protocol of the process supports a negotiation process between a customer and a supplier to agree on a demand forecast. The protocol begins with the customer, who requests a demand forecast. The supplier processes the request and may respond by accepting or rejecting it, as it is indicated by the *Xor* control flow segment. If it is accepted, the supplier undertakes to realize the required forecast; otherwise, the process finishes with a business failure. If the supplier accepts the request, the customer informs, in parallel, a sale forecast of its points of sales (POS) and its planned sales, as it is indicated by the *And* control

flow segment. Finally, with this information, the supplier generates a demand forecast and sends it to the customer. Then, the process ends. Each agent in the scenario described above is an instance of a CPAgent. The CPAgent is capable of executing CBPs models. In this case, we have two CPAgents that are capable of executing interaction protocols. Figure 4 shows the Petri Nets representing the customer and supplier roles in a Collaborative Demand Forecast. The dotted arrows that emanate from the transitions of individual Petri nets show the exchange of messages and business documents.

3 MDD-BASED METHOD TO GENERATE PN MODELS

A Model-Driven development approach for generating a technological solution for B2B Collaborations should support the derivation of the software artifacts from technology-independent models. In previous work, we proposed a model-driven architecture (MDA) (OMG, 2003) methodology that consists of three phases: analysis and design of collaborative processes, verification of collaborative processes and generation of B2B specifications (Villarreal et al., 2006a). The first phase consists in the modeling of CBPs from a business perspective, i.e. using concepts that are less bound to the implementation technology and are closer to the B2B collaboration domain. The UPColBPIP language (Villarreal et al., 2007b; Villarreal et al., 2010) is used for modeling technology-independent CBPs. The behavior of CBPs is represented through the definition of interaction protocols from a business point of view.

The second phase consists in verifying the correctness of CBPs defined in a UPColBPIP model. To support this, the MDA-based method for generating Petri Net specifications from a UPColBPIP model is applied (Villarreal et al., 2007a). Interaction protocols are formalized, transformed and mapped into Colored Petri Net (CPN)(Jensen and Kristensen, 2009) specifications, which are then verified with CPN Tools. Thus, the verification of CBPs is carried out at an early stage of the development, when most of the fundamental decisions of a B2B collaboration are carried out, and previous to the generation of the technological solution.

The third phase consists in selecting the target implementation technology to generate the technological solution and final software artifacts from CBP models.

In this work, we propose a MDD-based framework to generate technological solutions based on software agents in which process models are interpreted by software agents. To generate the PN model that each CPAgent require to execute a CBP, a process transformation approach is applied to derive the PN models of the organizations from a UPColBPIP model (see Fig. 3). Thus, the behavior required for the agents to execute a CBP is generated in an agile way and represented in the PN models, which are the software artifacts that interprets the agents.

4 IMPLEMENTATION OF B2B COLLABORATION AGENTS

In this section we describe the implementation of B2B collaboration agents architecture (see Fig. 5). The software agents were built by using the Java Agent DEvelopment Framework (JADE) (Bellifemine et al., 2007), which is a physical multi-agent development framework which complies with FIPA specifications and aims at simplifying the development and implementation of multi-agent systems (Bellifemine et al., 2007). The proposed B2B collaboration agent is composed of a message transport system (MTS) used for communicating with other agents or agent platforms, an agent management system (AMS) used for managing the agent life cycles such as starting and stopping, and a directory facilitator (DF) used for recording the services provided by an agent. Therefore, the CAAgent and CPAgents are executed on the JADE platform and they use ACL (FIPA, 2002) messages to communicate among them.

The *process machine* component of the CPAgent was implemented by using the Java-based Petri Net framework (JFern). JFern provides an object-oriented Petri Net simulator (Nowostawski, 2003). It consists of a lightweight Petri Net kernel, providing methods to store and execute Petri Nets in realtime, and for simulation. Furthermore, JFern supports XML based persistent storage of Petri nets and markings. The tokens of PNs contain *business documents*, which are in XML format and they are transported as parts of the content of ACL messages that agents exchange.

The B2B collaboration agents architecture includes an administration tool based on Eclipse Rich Client Platform. This tool has to be deployed in each organization and provides the support to:

1. Instantiate the JADE platform,
2. Instantiate the CAAgent of the organization,
3. Search, select and retrieve the UPColBPIP and Petri Net models stored in the local repository of the organization; and

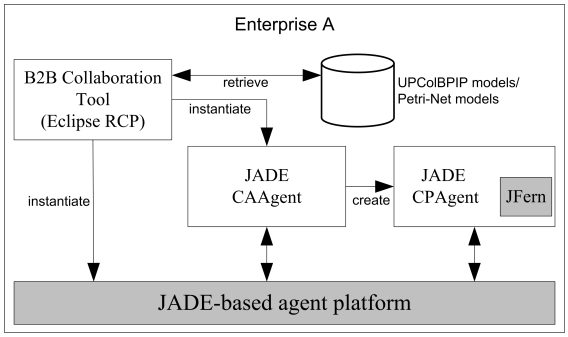

Figure 5: Agent-based architecture for B2B collaboration.

4. Administrate and monitor the CPAgents instantiated.

5 RELATED WORK

There are several approaches that exploit the benefits of software agents in the execution of the collaborative business processes. In (Kahl et al., 2007), a complete architecture for the design and agent-based implementation of cross-organizational business processes was defined. The principles of MDA are applied in three levels: business, technical, and execution. An advantage is shown by the generating of agents from a platform-independent model of a service oriented architecture (PIM-SOA). In addition, they consider the potentials of their execution offered by agents in comparison to other techniques like WS-BPEL engines.

Another approach (Zinnikus et al., 2008), supports a rapid prototyping by combining a model-driven framework for CBPs with an agent-based approach for flexible process execution, including a semantic service discovery for flexible service composition.

In these approaches the behavior and interaction protocols of the agents are defined in design and development time. However, they do not consider the generation of the agents' behaviors at run-time, which makes more flexible the architecture of the agents for executing CBPs and allows the establishing of dynamic B2B collaborations.

6 CONCLUSIONS

In this work we have proposed a B2B collaboration agents architecture for managing B2B interactions that allow organizations to dynamically establish collaborations, and carry out a decentralized execution of collaborative business processes (CBPs) with their partners. B2B collaboration agents architecture are composed by CAAgents and CPAgents.

We apply High-Level Petri Net models to represent the required behavior of a CPAgent to perform the role an organization fulfills in a CBP. Petri Net (PN) models are derived from UPColBPIP models of CBPs. Therefore, the decentralized execution of a CBP is achieved by the enactment of a PN model carried out by each CPAgent representing the role of an organization in the CBP.

In addition, due to the behavior of CPAgents is driven by the Petri Net-based process machine, their actions are generated in run-time to support the execution of any interaction protocol representing a CBP. These concrete actions are composed of speech acts and business documents, the concrete actions are obtained at run-time according to the logic defined in PN models. This makes more flexible the architecture of agents for executing CBPs. This is different from the traditional development of agents where actions and interaction protocols that can be executed are beforehand implemented and defined in design-time.

Finally, an implementation of the B2B collaboration agents architecture is based on the JADE agent platform. A Petri Net simulator was used to implement the process machine of the CPAgent. Also, a distributed B2B collaboration management tool based on the Eclipse platform was developed that enables organizations to manage their agents and their own repository of CBP models.

Future work is about how establish an algorithm and automatic mechanism for the CBP *Request for Collaboration*, in which the CAAgent will determine if *agree* or *refuse* to establish a dynamic agreement with another organization to collaborate and execute a CBP. Another future work is about the addition of mechanisms and tools to discover CBP models either in a public or organizations' repositories. In the current implementation of the CPAgent the private activities or processes of the organizations, which generate or processes the exchanged messages, are simulated. However, a support for a real integration and execution will be developed by integrating Web service technology with agents.

REFERENCES

Bellifemine, F., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.

FIPA (2002). FIPA Agent Communication specifications deal with Agent Communication Language (ACL). Available: http://www.fipa.org/repository/aclspecs.html.

Guo, L., Robertson, D., and Chen-Burger, Y. (2005). A novel approach for enacting the distributed business-workflows using BPEL4WS on the multi-agent platform. In *The 2005 IEEE International Conference on e-Business Engineering*.

Jensen, K. and Kristensen, L. (2009). *Coloured Petri Nets, Modelling and Validation of Concurrent Systems*. Springer Heidelberg, USA.

Kahl, T., Zinnikus, I., Roser, S., Hahn, C., Ziemann, J., Müller, J. P., and Fischer, K. (2007). Architecture for the design and agent-based implementation of cross-organizational business processes. In Gonalves, R. J., Müller, J. P., Mertins, K., and Zelm, M., editors, *Enterprise Interoperability II*, pages 207–218. Springer London.

Nowostawski, M. (2003). JFern - Java-based petri net framework.

OMG (2003). MDA guide v1.0.1, 03-06-01.pdf. Available: http://www.omg.org/cgi-bin/doc?omg/03-06-01.

Roser, S. and Bauer, B. (2005). A categorization of collaborative business process modeling techniques. In *7th IEEE International Conference on E-Commerce Technology Workshops*, pages 43–54.

Trappey, C. V., Trappey, A. J., Huang, C.-J., and Ku, C. (2009). The design of a JADE-based autonomous workflow management system for collaborative SoC design. *Expert Systems with Applications*, 36(2):2659–2669.

Villarreal, P., Roa, J., Salomone, H., and Chiotti, O. (2007a). Verification of models in a MDA approach for collaborative business processes. In *10th Ibero-American Workshop of Requirements Engineering and Software Environments*.

Villarreal, P., Salomone, E., and Chiotti, O. (2006a). A MDA-based development process for collaborative business processes. In *European Workshop on Milestone, Models and Mappings for Model-Driven Architecture, 3M4MDA 2006*.

Villarreal, P., Salomone, E., and Chiotti, O. (2006b). Transforming collaborative business process models into web services choreography specifications. In Lee, J., Shim, J., Lee, S., Bussler, C., and Shim, S., editors, *Data Engineering Issues in E-Commerce and Services, Second International Workshop, DEECS 2006*, LNCS. Springer, Heidelberg.

Villarreal, P., Salomone, E., and Chiotti, O. (2007b). Modeling and specifications of collaborative business processes using a MDA approach and a UML profile. In Rittgen, P., editor, *Enterprise Modeling and Computing with UML*, pages 13–45. Idea Group Inc, USA.

Villarreal, P. D., Lazarte, I., Roa, J., and Chiotti, O. (2010). A modeling approach for collaborative business processes based on the up-colbpip language. In Aalst, W., Mylopoulos, J., Sadeh, N. M., Shaw, M. J., Szyperski, C., Rinderle-Ma, S., Sadiq, S., and Leymann, F., editors, *Business Process Management Workshops*, volume 43 of *LNBIP*, pages 318–329. Springer Berlin Heidelberg.

Weske, M. (2007). *Business Process Management. Concepts, Languages, Architectures*. Springer, Heidelberg.

Zinnikus, I., Hahn, C., and Fischer, K. (2008). A model-driven, agent-based approach for the integration of services into a collaborative business process. In *7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 241–248.