A Specification and Validation Approach for Business Process Integration Based on Web Services and Agents

Djamel Benmerzoug[1], Mahmoud Boufaida[1], and Fabrice Kordon[2]

[1] LIRE Laboratory, Computer Science Department,
Mentouri University of Constantine 25000, Algeria,
{benmerzougdj,boufaida_mahmoud}@yahoo.fr

[2] LIP6 Laboratory, Pierre et Marie Curie University,
4, place Jussieu, 75252 Paris Cedex 05 France
fabrice.kordon@lip6.fr

Abstract.. In this paper, we present a new approach for business processes integration. Our approach is based on interaction protocols that enable autonomous, distributed business process modules to integrate and collaborate. In our case, the business processes integration is modelled using AUML and specified using BPEL4WS. Furthermore and to increase the reliability of interaction protocols at design time, our approach presented in this paper can validate the BPEL4WS specification with business constraints (specified by means of OCL). The validated BPEL4WS specification is considered as a specification language for expressing the interaction protocols of the multi-agents system, which can then intelligently adapt to changing environmental conditions.

1 Introduction

Business Processes Integration (BPI) is a key technology for business to business collaborations. Nowadays many enterprises have automated their internal business processes with workflow technologies. They have now a new challenge: the automation of their collaborations with partner enterprises, in open and very dynamic environments, to accelerate their business in a cost-effective manner. Web Services (WS) are a promising technology to support these type of collaborations [1]. WS are an XML-based middleware technology that provides RPC-like remote communication, using in most cases SOAP over HTTP.

Heterogeneity, distribution, openness, highly dynamic interaction, are some among the key characteristics of another emerging technology, that of intelligent agents and Multi-Agent Systems (MASs). WS and intelligent software agents share many common features, and this suggests that some relationship between the two technologies should exist. Actually, the most recent literature in the agents' field devotes much space to these relationships [2].

This paper present a new approach based on WS and agents for integrating business process. The used approach is based on interaction protocols that enable

autonomous, distributed business process management modules to integrate and collaborate. In order to reach an implicit consensus about the possible states and actions in an interaction protocol, it is necessary for the protocol itself to be correct. Indeed and to increase the reliability of interactions protocol at design time, we have developed an approach for the specification and the validation of BPI. In our case, the BPI is modelled using AUML (Agent UML) [7] and specified using BPEL4WS (Business Process Execution Language for Web Services) [8]. We then use this precise specification to generate a validation tool that can check that a BPEL4WS document is well-formed (that is the BPEL4WS preserving the business constraints).

In the next section we briefly introduce our approach. Sections 3 and 4 discuss our approach to engineering interaction protocols exploiting AUML/OCL for the modelling stage, and BPEL4WS for the specification stage. Section 5 gives a short description on how the BPEL4WS specifications can be validated. Our conclusion and future work are described in the final section.

2 An Overview of the Proposed Approach

BPI modelling and reengineering have been longstanding activities in many companies in recent years. Most internal processes have been streamlined and optimized, whereas the external processes have only recently become the focus of business analysts and IT middleware providers. The static integration of inter-enterprise processes as common in past years can no longer meet the new requirements of customer orientation, flexibility and dynamics of cooperation [4].

In this paper, we consider two type of business processes, the private business processes and the public business ones. The first type is considered as the set of processes of the company itself and they are managed in an autonomous way. Private processes are supported within companies using traditional Workflow Management Systems, Enterprise Resources Planning systems or proprietary systems. These systems were intended to serve local needs. In other hand, public business processes span organizational boundaries. They belong to the companies involved in a B2B relationship and have to be agreed and jointly managed by the partners.

The B2B integration scenarios typically involve distributed business processes that are autonomous to some degree. Companies participating in this scenario publishes and implements a public process. The applications integration based on public process is not a new approach. The current models for BPI are based on process flow graphs [4], [5], [6]. A process flow graph is used to represent the public process. This approach lacks the flexibility to support dynamic B2B integration. In contrast, our approach (figure 1) presents an incremental, open-ended, dynamic, and personalizable model for B2B integration.

We have previously developed an agent-based method for developing cooperative enterprises information systems [10]. This method permits to explicitly mapping the business process into software agents. In this paper, we describe the use of interaction protocols to define and manage public processes in B2B relationships. This process is modelled using AUML [7] and specified using BPEL4WS [8]. The use of interaction protocols to define public processes enable a greater autonomy of companies because

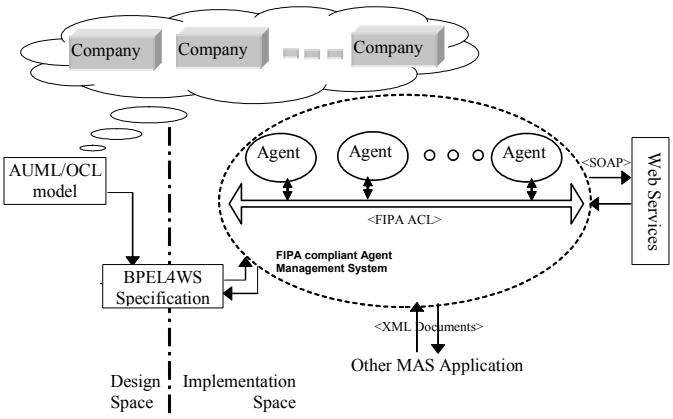

Fig. 1. The Proposed Approach

each company hides its internal activities, services and decisions required to support public processes. In this way, the interaction protocols provide a high abstraction level in the modelling of public processes. The AUML model is mapped to a BPEL4WS specification, which represents the initial social order upon a collection of agents (figure 1). Since BPEL4WS describes the relationships between the WS in the public process, agents representing the WS would know their relationships a priori. Notably, the relationships between the WS in the public process are embedded in the process logic of the BPEL4WS specification. This relationships entails consistency problems, which can best be solved at the level of models. We then use this precise specification to generate a validation tool that can check that a BPEL4WS document is well-formed (the BPEL4WS preserves the business constraints).

3 Modelling Public Business Processes with Interaction Protocols

Interaction Protocols have been used in the area of multi-agent systems to represent interactions among agents [7]. In the context of B2B relationships, an interaction protocol allows to model and manage the interactions among the enterprises involved in a B2B relationship. These interactions represent public business processes that the enterprises agreed on the collaboration. The main objective of interaction protocols is to abstract public processes from the services to be invoked within each enterprise for executing the internal activities supporting public processes [2]. In AUML, we can define an interaction protocol through a protocol diagram, which is an extension of the sequence diagram of UML. Roles are represented by a rectangular box indicating the company that performs the role. Role lifeline defines the time period during which the company participates in the protocol. The lifeline may split up into two or more lifelines, using logical connectors, to show AND and OR parallelism and decisions, corresponding to the branches on the incoming message flow.

In our approach, initial requirements permit to capture dependencies between different roles in different private business processes (possibly other public business process). This dependencies are detailed using protocol diagrams.

The only use of modelling languages (like AUML) is not sufficient, since it missing adequate means to specify constraints over the dynamic behavior of an AUML model. However, it is essential to support the definition of business rules constraints already in the early phases of development in order to specify correct system behavior over time. Facing this problem, the Object Constraint Language (OCL) [9] provides the chance to explicitly and automatically deal with business constraints when building agent-based applications. The reason for using OCL is that offers a textual means to enhance AUML diagrams, offering formal precision in combination with high expressiveness.

The definition of a business process constraints is nothing else then a constraint on the interaction diagrams of the AUML-based public business process. In our case, we used OCL to define the preconditions and postconditions for our interaction protocols defined previously. The precondition captures necessary and sufficient conditions that determine when a constraint is applicable. The postcondition describes the intended update to the model, that is, the effect of the messages exchange. In the case of defining the business process constraints, we specify the corresponding business interaction protocol after the OCL keyword `context` and followed by the keyword `inv` for invariants.

4 From an AUML Model to a BPEL4WS Specification

BPEL4WS represents the merger of two process description language, IBM's Web Services Flow Language (WSFL) and Microsoft's XLANG. It provides both graph-based and block-based control structures, making it capable of representing a wide range of control flows. This merger has created the market consolidation necessary to make BPEL4WS the de facto standard for expressing BPI consisting of WS. BPEL4WS can be used to describe executable business processes and abstract processes. Abstract processes are used to create behavioral specifications consisting of the mutually visible messages exchanged between transacting parties executing a business protocol.

In our case, a BPEL4WS specification describes a public business process by stating whom the participants are, what services they must implement in order to belong to the public business process, and the control flow of the public process. for example, the <partners> section defines the different parts that participate in the public process. Each partner is given a service link type and the role it will perform as part of the service link. The <variables> section defines the variables used by the process. The process definition of the public process occurs after the fault handlers section and before the close process tag. A public business process is defined using BPEL4WS activity constructs (sequence, flow, while, switch,…etc).

Many of the features that characterize BPEL4WS abstract processes, make it very suitable to represent interaction protocols and correspond in a prettyprecise way to

those that characterize AUML. Table 1 describes the ideas behind the automatic translator from AUML to BPEL4WS.

Table 1. The AUML/BPEL4WS Mapping Overview

AUML	BPEL4WS
Roles	<partners>
sequence	<sequence>
AND-split	<flow>
OR-split	<switch>
XOR-split	<if>
Iteration	<while>
Message	<invoke>, <receive> and <reply>

5 Generating a BPEL4WS Validator

As we already have said before, the BPEL4WS process specification is considered as a specifying language to express the interaction protocol of the multi-agents system. In order to reach an implicit consensus about the possible states and actions in an interaction protocol, it is necessary for the protocol itself to be correct. We believe that from the point of view of efficient integration of business process, there are two key issues: (1) a precise and intuitive way to integration conversation partner into an interaction protocol (the combination of AUML/OCL model); (2) the ability to simulate the interaction protocol is a helpful addition to formal validation and verification (the verification stage is not described in this paper).

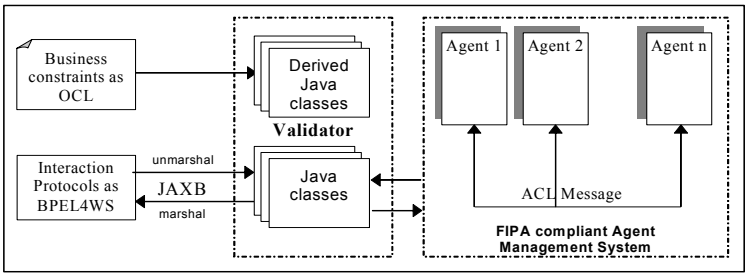

Fig. 2. Validation of the BPEL4WS Specification

A set of AUML interaction diagrams are created to model the public business process, where business constraints are specified in OCL. We then use this precise specification to generate a Validation tool that can check that a BPEL4WS document is well-formed (the BPEL4WS preserves the business constraints). A model constraint that fails would indicate an invalid combination of BPEL constructs.

In this research, we exploit the Sun Microsystem Web Services Developer Pack [11]. In particular, we use the JAXB (Java Architecture for XML Binding) library to

build Java classes from an BPEL4WS specification. Our validation tools permits the combination of code generation from BPEL4WS specification and code generation from constraints; with the resulting code facilitating the validation of the constraints against instances of the specification. In particular, the BPEL4WS code and the constraint code are generated separately (see figure 2).

6 Conclusion

In this paper, we presented an approach to modelling, specification and validation of BPI. Our approach is based on interaction protocols where the autonomy of the participants must be preserved. Indeed, the BPI is modelled using AUML and specified using BPEL4WS. Our approach presented in this paper can validate the BPEL4WS with the business constraints through the BPEL4WS validator. The validated BPEL4WS specification is considered as a specification language for expressing the interaction protocol of the multi-agents system.

Our primary future work direction is certainly the exploitation of BPEL4WS features to publishing the protocols specification on the Web and we will describe how the MAS can use the verified and the validated BPEL4WS specification to establish the BPI.

References

1. Jae-yoon Jung, W.H., Kang, S.H.: Business Process Choreography for B2B Collaboration. IEEE Internet Computing (2004) 37-45
2. Luck, M., McBurney, P., Shehory, O., Willmott, S.: Agent Technology: Computing as Interaction - A Roadmap for Agent-Based Computing. AgentLink III, (2005)
3. Koehler, J., Tirenni, G., Kumaran, S.: From Business Process Model to Consistent Implementation: A Case for Formal Verification Methods. 6th International Enterprise Distributed Object Computing Conference. IEEE Computer Society (2002)
4. Peregrine B2B Integration Platform, www.peregrine.com
5. WebMethods B2Bi www.webmethods.com.
6. Vitria Business Ware. www.vitria.com.
7. Huget, M., Odell, J.: Representing agent interaction protocols with agent UML. In 3rd International Joint Conference on Autonomous Agents and Multiagent Systems. IEEE Computer Society (2004) 1244-1245
8. Business Process Execution Language for Web Services Version 1.1, 05 May 2003. http://www-106.ibm.com/developerworks/
9. OMG; Object Constraint Language Specification; http://www.omg.org/cgi-bin/doc?formal/03-03-13
10. Benmerzoug, D., Boufaida, Z., Boufaida, M.: From the Analysis of Cooperation Within Organizational Environments to the Design of Cooperative Information Systems: An Agent-Based Approach. R. Meersman et al. (Eds.): OTM Workshops 2004, LNCS Springer-Verlag, (2004) 496-506
11. Sun Microsystems. Java Web Services Development Pack 1.1. http://java.sun.com/webservices/webservicespack.html/, (2006).