A Framework for Automated Service Composition in Collaborative Networks

Hamideh Afsarmanesh, Mahdi Sargolzaei, Mahdieh Shadi

University of Amsterdam, Informatics Institute,FCN group,
Science Park 107, 1098 XG Amsterdam, The Netherlands
{H.afsarmanesh, M.sargolzaei1, M.shadi}@uva.nl

Abstract. This paper proposes a novel framework for automated software service composition that can significantly support and enhance collaboration among enterprises in service provision industry, such as in tourism insurance and e-commerce collaborative networks (CNs). Our proposed framework is founded on service oriented architecture (SOA) paradigm, in which software services implementing on-line business services that are provided by different enterprises, will be formally defined, using an extended BPMN notation to capture their *semantics and behavior*, as well as the WSDL notation to capture their *syntax*. Furthermore, with registering the *syntax, semantics and behavior* of these software services in a service repository at the CN, the task of service discovery in this framework can go far beyond the current practice, which comprise of service search by name, to the possibility of discovering by service behavior. The paper addresses enhancement of automated software service integration in CNs, through the application of the Reo coordination language, which is used to formalize interaction among the composed services. The main reason for using Reo in this context is that it supports separating the computations needed by software components in an integrated system from their interactions. The suggested framework provides more flexibility, adaptability, as well as cost-effectiveness in service composition, when supported in collaborative networks.

Keywords: Software service composition, service discovery, business process modeling, business service, collaborative network, coordination language

1 Introduction

Nowadays, in a number of application areas such as tourism and e-commerce, an increasing number of SME organizations are formalizing the definition of their provided *business services*. This formal definition is usually provided as business processes (BPs), e.g. a BP is defined to represent the reservation of a hotel room or an airline ticket, and they usually apply certain standards such as BPMN or UML for this definition. The resulted sets of BPs at the SME are then used by software developers

that implement them as software services. For a large number of service industries such as the tourism, insurance, banking, etc. most of their business services provided to the customers are now accessible on line (through Internet), i.e. developed as *web services,* e.g. the on-line hotel reservation web service. These web services are typically accessible by customers through user-friendly interfaces, which automate the execution of their requested transactions e.g. making a hotel or airline reservation, at the local SME sites.

But SME organizations are increasingly interested in working together and joining their knowledge, skills, resources, capabilities, and capacities, and in establishing collaborative networks (CNs) [1]. One form of CNs, the so called Virtual Organization (VO), is usually established for one of the following two purposes. One purpose is to target one specific emerged opportunity in the market or society, for which a number of organizations would be selected by the VO broker and invited to accept the joint responsibility of fulfilling tasks needed to achieve the common goal of the VO. A second purpose is for innovation, when usually a VO broker identities a potential opportunity that can be fulfilled through merging abilities, resources, capacities, etc. of a number of SMEs who get invited to the VO [2].

In order for VOs to compete in the market and society with real large existing organizations, the base for collaboration among its partner SMEs must also be pre-established before the VOs can operate. For this purpose, usually another form of CN, the so called Virtual organizations Breeding Environment (VBE) [3] is established as an association/community of SMEs in a sector, who have general common and/or compatible interests, who are interested in involvement in VOs, and who are also willing to work and share their potentials and competencies with other organizations. Since VBEs on one hand gather and organize large amount of information/knowledge about their member organizations, and on the other hand establish the base for common/uniform interoperation among their member organizations, they represent valuable communities, and therefore usually they do not dissolve [4]. But sometimes VBEs metamorphose either through expansion with another related area of activity, or narrowing down and focusing on a subset of the original activities.

Formation of a VBE is typically aimed at bringing the following set of advantages to its member organizations [5]:

(i) Capturing more opportunities and bigger market, and thus making more profits through sharing their customers
(ii) Reducing individual costs through focusing/applying only their core competence, while benefiting from the existence of complementary expertise in the network
(iii) Increasing the ability to take risks, through sharing and distribution of potential losses, while agreeing to share their profits

Focusing on the service industry, which is the main emphasis of this paper, in order to join others and collaborate effectively, SMEs in a VO must together act as a single entity, and therefore they must share their business services and software systems with each other as if they all belong to and co-work within one single real organization. But to share software services, they must be formally and uniformly defined. Such unification in definition format can occur at the VBE level. As such, once in a VO, other SME partners will be able to share them, e.g. to integrate them

with each other and/or together create new value-added services and innovate in this industry.

For example, consider the case of one SME (SME-S1) in a VBE (V1) planning to simply create an integrated tourism package, including the reservation of flight-ticket, hotel-room, day-trip and dinner at restaurants, and assume that these business services are all already implemented by several different SMEs at the VBE as web-services by different other SMEs in this VBE. In fact once SME-S1 identifies the most-fit SMEs to work with for creating this package as an integrated service, SME-S1 (as the VO broker) can start forming a VO together with these other SMEs. However, today due to the lack of uniformity in full and formal definition of implemented business services at SMEs, the creation of such an integrated single tourism package and providing it as an on-line service to the customer is quite challenging. In this paper we aim to address this challenge and define an approach and architecture that can in fact support the semi-automation of this challenging task.

Today, even without any automation, for developing such an integrated service, the following steps are needed to be taken:

(a) SME-S1 must first identify any and all potentially relevant web-services shared by other SMEs in the V1. But clearly to identify relevant web-services for this purpose, it is not enough for SME-S1 to only have access to the syntax information of these web-services, e.g. their names, and the name/type of their input and output. This is due to the fact that "names" for web-services and their input/output are not even necessarily mnemonic and/or using any common standards. On the contrary, these names are usually selected and assigned by different software developers and usually represent only their personal preferences.

(b) Beside the naming challenges, also the semantics of the web-services and each of their input and output elements are not uniformly provided. There is a need for common ontological definitions to clarify the actual intention of the software developer behind these elements.

(c) Furthermore, for the purpose of developing a new integrated web-service, through composition of other existing web-services, although SME-S1 (as the developer of this new integrated service) does not need to know and fully understand the exact code and how each of the existing component services are implemented, SME-S1 must in fact have a very good understanding of the functionality and interaction (namely the behavior) of each component service, in order to integrate it with the others. In some cases in fact SME-S1 needs to adapt the behavior of some of these services in order to make them match the others and become compatible.

But considering the real practice, it should be first noted that the SMEs in the VBE are fully independent and autonomous, therefore unlike the case of software service integration within one large enterprise, or through outsourcing, in the case of SMEs in service industry, there are no common base for definition of service syntax and semantics, and no standards is observed when developing their software services. Therefore, as a first step to support collaboration among such service providing SMEs in a VBE, it is necessary to establish a framework for their service interoperability. This interoperability framework shall create the common understanding about the

existing services in the VBE and thus requires a common frame and meta-data to formally express the syntax, semantics, and behavior of software services.
The structure of this paper is as follows. Section 2, address the application of SOA to the collaborative networks. In Section 4 we address the implemented architecture for the approach and introduce its 4 aspects of service modeling, service registering, service discovery, and service integration. Finally in Section 4, some conclusive remarks are drawn.

2 Applying SOA to Collaborative Networks

From the ICT perspective of a CN, supporting its agility, cost-effectiveness, and adaptability need designing more advanced infrastructures. The SOA and service utility paradigms [6] as prominent ICT approaches provide the strong required architecture for a system in which there are components interacting with each other, while their functionalities are viewed as software modules called services [7]. In service industry, software service accessibility by each enterprise in the collaborative network, in support of service interoperability, results in reinforcing its collaboration with others and decreasing its service development and hosting costs. In the SOA context, SaaS (Software-as- a-Service) [8] is a powerful model that can benefit the CNs. As clients, remote access to services can be supported for CN members, who can for instance pay per use, based on the contractual rules specified in SLAs (Service Level Agreement) [9]. As such, it does not matter where the services providers are located, and how the services have been deployed [10].

Using SaaS, CN members (as clients) can search among existing services, build and register their own new services, compose existing services, or adapt the accessed services according to their needs. In this sense, both common representation of software services as well as formal modeling and coordination of their interactions are fundamental issues which assist flexible and better service discovery and service composition. We focus and discuss these two aspects in the following paragraphs.

Currently there are several notations and graphical modeling tools used for modeling of business processes, such as the Business Process Modeling Notation (BPMN) and the Unified Modeling Language (UML) Activity Diagram, the Event-driven Process Chains (EPC), and the Role Activity Diagrams (RADs) [11]. BPMN and UML Activity Diagrams are currently considered as the two most suitable and popular ones. While these standards are largely overlapping, BPMN is slightly stronger and more expressive [12], so we have choose BPMN as the basic notation for modeling business processes in the proposed platform. Despite its popularity, the BPMN notation is too abstract to convey sufficient details required to represent the syntax, semantics, and operational behavior properties of software services, necessary for service composition/integration. Interface, data types and structures of each service are represented by its syntax. Web Service Definition Language (WSDL) is the most popular language for syntactical representation of software services [13].

Most web service models and languages such as Simple Object Access Protocol (SOAP) [14], and Universal Description Discovery Integration (UDDI) [15] are also used at the syntactic level. These technologies support interoperability between

different services through common standards, but human interaction is still needed to search for appropriate services and for composing them in a reasonable manner. The intervention of human is however in contrast to the strategic goals of SOA, such as the scalability, agility, etc. [16]. Therefore, it is necessary to automate main tasks of SOA such as service discovery, composition and adaptation to the extent possible. To achieve this goal, besides the syntax, we should also consider more conceptual properties of services that are known as *semantics*. The semantics represent conceptual aspects of services, using explicit and machine-interpretable descriptions.

Further to syntax and semantics, *behavioral properties* of a service must also be formally defined. These properties address the set of operations involved in a service and indicate the order of invocations of these operations. Composition and integration of web services lie at the heart of our framework and they cannot be successful without behavioral specification of software services. BPMN may also be used to define the behavior of a service, but, there are some drawbacks and ambiguities in this aspect of BPMN [17]. Therefore, developing service ontology is required to specify service semantics and disambiguate current BPMN process specification of services.

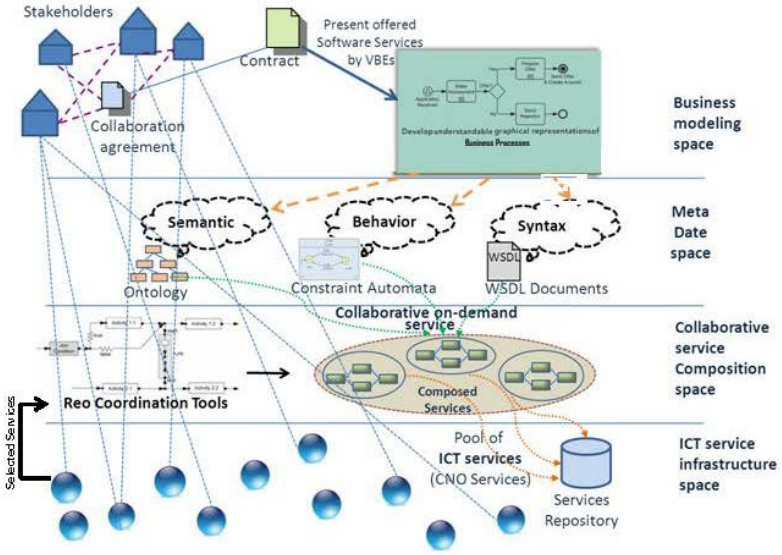

Fig. 1: A new SOA-based platform for a CN

We propose a new SOA based platform for a CN, as shown in Figure 1. This architecture is composed of four layers including *business modeling space, meta data space, service composition space* and *IT infrastructure space*. As Figure 1 illustrates, members/stakeholders of a CN share their software services according to their contracts in *business modeling space*. These services shall be represented as extended BPMN diagrams, from which Meta Data (as represented in *meta data space*) is extracted, to derive their syntactical, semantics and behavioral descriptions. Software Services are designed to support interoperable machine-to-machine interaction over a network using two basic standards including WSDL which describes the interface in a machine-interpretable format, and SOAP [13] which is used to format the exchanged messages. For documenting syntactical information of software services, we need to parse the XML-based documents generated by WSDL and Soap, to extract the syntax

of operations and messages of web services. We also use the formal notation of constraint automaton [18] for specification of the (external) behavior of a service. In this formalism, a web service is represented as a sequence of operations in which there are some constraints defined on operations' invocation order [19]. In the next section, we discuss requirements and the needed architecture to implement this platform, considering the mentioned Meta Data.

3 Implementation Architecture and Requirements

Our proposed framework for implementation of composition of software services within collaborative networks consists of four components, which are implemented as modules, to address: (i) modeling business processes and obtaining their formal meta-data (on syntax, semantics, and behavior), (ii) registering the extracted metadata in corresponding repositories, (iii) discovering, matching and adapting appropriate services (needed for their composition into a new integrated service), and (iv) coordinating, integrating, and executing composed services. Thus, the complete implementation architecture includes four modules, labeled as: modeling, registering, discovery, and integration.

3.1 Modeling of Services

This module supports the formal definition of the software services to be shared with the community of enterprises at the CN.

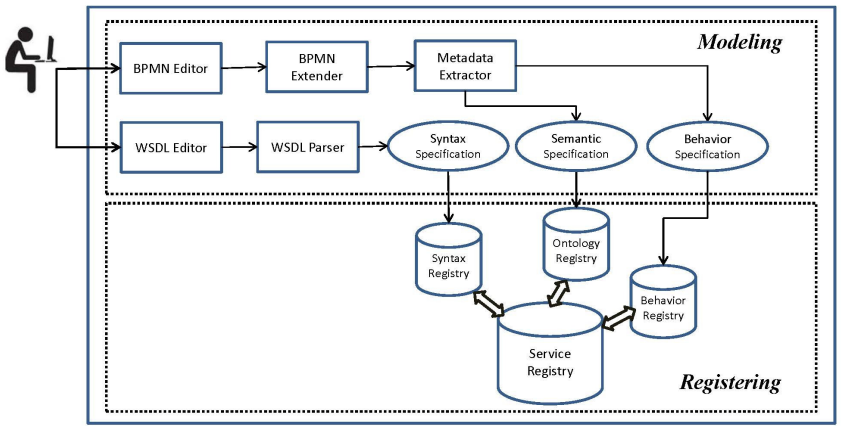

Fig. 2: Architecture of *Service Provision* in the platform

See Figure 2 that represents the general architecture for *service provision* with two of its modules, modeling and registering. From the end users point of view at each enterprise, their shared business services are required be defined. At first, user presents the complete flow of his every business process as a BPMN diagram or BPD (Business Process Diagram) using the BPMN editor. Each diagram should be further

extended to define the syntax, semantic and behavior of its corresponding software service. Such enrichments, which is represented in the box labeled as “BPMN Extender”, consist of three parts: adding new descriptions, refining ambiguities, and annotating to facilitate extraction of these metadata. The task of extracting elements needed to register the behavior and semantics metadata of the service in appropriate repositories has been presented with the box labeled “Metadata Extractor”. To obtain Syntactical metadata, besides some basic syntactical meta-data derived from the primary BPMN diagram, we need to load and parse WSDL documents of web services, which are represented in the boxes labeled as "WSDL Loader" and "WSDL Parser". These steps are shown within the *Business Modeling Space* and *Meta Data Space* in Figure 1.

3.2 Registering of Services

As mentioned in Section 3.1, we formalize the externally observable behavior of the software services in terms of constraint automata. There is therefore, a constraint automaton generated for each service which should be captured within the behavior registry. The data structure of this registry is formed of state tables which store current state, next state, ports (operation names) and data constraints of constraint automaton [19]. Behavior registry of services is needed to generate final executable code for integrated services, as described in the next subsections. Also the syntactical metadata for services need to be captured in syntax registry. Most activities in the framework e.g. discovery, integration and execution of services, refer to this registry.

The services meta-data is improved by using OWL-S, a rich description language for representing semantics. Service semantics are then captured within the ontology registry, which describes "individual" services, together with the set of their "property assertions" relating individual services to each other [20].

Obviously beside the three metadata registries addressed above, the framework also needs a service registry, such as UDDI which contains all shared services with their syntactic metadata. But in addition to syntactic information, we also keep the ontological and behavioral metadata associated with each published service synchronized in the service registry. Figure 2 illustrates the required operations for registering new services by the service provider in the platform. This task uses the BPMN for modeling the behavior and semantic of the services and the WSDL for specification of their syntax. We will describe in the next section how we use these registries to compose existing services into an integrated service. All mentioned registries are considered as *Service Repository* within the *ICT Service Infrastructure Space* in Figure 1.

3.3 Discovery of Services

Figure 3 represents the general architecture for *service composition* in our framework, where its four different aspects of modeling, registering, discovery, and integration are also indicated. We will of course in this subsection focus on describing the Discovery aspects of this architecture. Let us consider the example provided in

Section 1, where the user in SME-S1 intends to create a new integrated tourism service composed of a number of other services (i.e. reservation of flight ticket, reservation of accommodation, etc.). This user will then need to first specify this integrated business process and its components through the BPMN editor. The task of decomposing the integrated service into its component services is represented in the box labeled as "Decomposer". The step after this decomposition is the performed by the service search engine, which starts to match as much as possible the description of the component services defined by the user against the services existing at the service registry, to be offered as alternatives to the user. As such, the role of Service search engine is the discovery of potential required services among the existing shared services. The Service search engine has to search the syntax and ontology service registries simultaneously, for matching the syntactic and semantic aspects of the registered services against the decomposed components. The report of all the partially matched services, as well as those that are not found will be sent to the user, where he can apply his/her preferences in selecting among the alternatives.

Nevertheless standardizing the service definitions does not remove the need for service adaptation. Usually the discovered services, do not fully, but only partially match the requirements and context of the desired service component (decomposed tasks). In this case, in case such a service is selected by the user, the service adapter modifies it in order to make it compatible with the user's desired service. This adaptation provides semi-automated support for identification and resolution of interface-level mismatch between the intended and the discovered services[21]. As Figure 1 shows, the dashed lines connecting the *Composed Services* to the *Service Repository* indicate the process of service discovery.

3.4 Integration of Services

We apply the Reo [18] circuits to model the coordination aspects related to the interactions among software services. As such, the user at SME-S1 will use the Extensible Coordination Tools (ECT) [22], which is a collection of Eclipse plug-ins, to design these interaction circuits, using a drag-and-drop graphical interface. But, some essential activities like communication between web services and the Reo circuits, and generating executable code from a circuit diagram cannot be done by ECT. Therefore, we need a wrapper to connect web services to the boundary nodes of Reo circuits as Reo components, to communicate with web services and exchange data via SOAP message, and also translate Reo circuits to java code [19]. Using wrapper provides interaction among the running software services in a way coordinated by the Reo circuits and also dynamic and automatic service invocation. For these purposes, the Reo compiler produces the glue-code among the individual services involved in the integrated service, and the Proxy generator produces the proxies for each involved service. Furthermore, to generate final executable code for a composed service, we need its syntactic metadata and behavioral properties of selected services. We have described more details about the design and the current implementation of this wrapper in another paper [19]. As shown in Figure 1, the *Selected Services* would be integrated by *Reo Coordination Tools* within the *Collaborative Service Composition Space*.

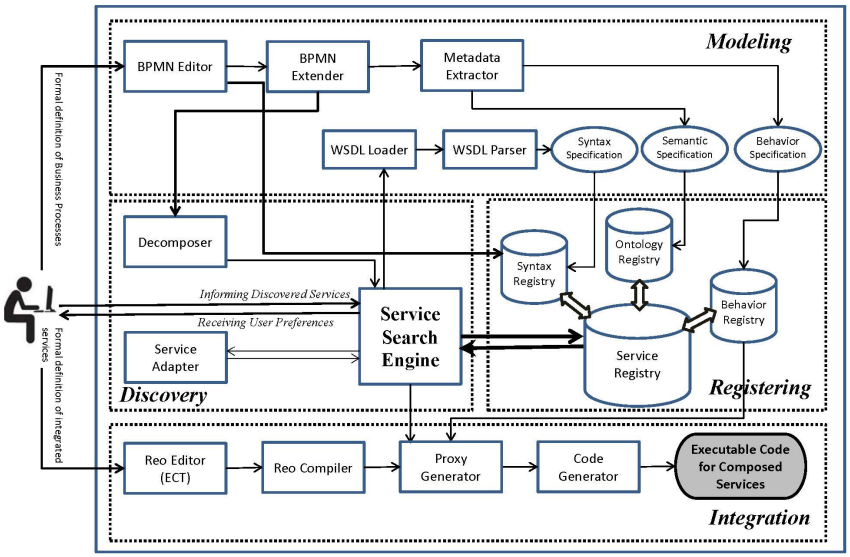

Fig. 3. Architecture of *Service Composition* in the platform

4 Conclusion

This paper addresses the design of a framework to assist service providers in a collaborative network, e.g. the SMEs involved in a tourism or insurance VBE, to uniformly create the formal definition of their services, so that they can be understood and shared with others. For this purpose, the paper proposes a novel SOA-based Collaborative Business ICT framework to support CNs in service provision industry with their effective collaboration through sharing and composition of their web services. This approach supports the organizations involved in a service industry CN with a platform-independent, easy-to-use, on-demand, and pay-per-use ICT service interoperability infrastructure. From the technical point of view, this is a web-based integrated platform devoted to the CNs, for providing value-added business services, via service composition. The platform improves software service discovery and service integration using variant metadata, as well as concrete formal machine readable definitions. Moreover, it uses Reo circuits as coordination models, used to coordinate the complex interaction protocols among enterprise business processes, which is one of the most challenging issues in modeling and executing business processes in CNs.

Acknowledgements

The work on this paper is partially supported by the FP7 project GLONET, funded by the European Commission.

References

1. Camarinha-Matos, L.M., Afsarmanesh, H.: A comprehensive modeling framework for collaborative networked organizations. In the Journal of Intelligent Manufacturing, Springer publisher. 18(5), pp. 527-615 (2007)
2. Kaletas, E. C., Afsarmanesh, H., Anastasiou, M., Camarinha-Matos, L.M.: Emerging technologies and standards, in Virtual Organizations: Systems and Practices. Springer, ISBN 0-387-23755-0, chap. 2.2, pp. 105-132 (2005)
3. Afsarmanesh, H., Camarinha-Matos, L.M.: On the classification and management of Virtual organisation Breeding Environments. IJITM .8(3), pp. 234-259 (2009)
4. Camarinha-Matos, L.M., Afsarmanesh, H.: A framework for virtual organization creation in a breeding environment. Annual Reviews in Control 31(1), pp. 119-135 (2007)
5. Afsarmanesh, H., Camarinha-Matos, L.M., Msanjila, S.S.: On Management of 2nd Generation Virtual organizations Breeding Environments. In the Journal of Annual Reviews in Control. Elsevier. 33(2) , pp. 209-219 (2009)
6. Framing the future of the Service Oriented Economy, http://www.nessi-europe.com/documents/NESSI_SRA_VOL_1_20060213.pdf (2006)
7. Singh, M., Huhns, M.: Service Oriented Computing. Wiley, Chichester (2005).
8. Software as a Service: Strategic Backgrounder, http://www.siia.net/estore/ssb-01.pdf (2001)
9. Web Service Level Agreement (WSLA) Language Specification. V 1.0, http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf(2003)
10. Ma, D.: The Business Model of Software-As-A-Service. IEEE International Conference on Services Computing, SCC 2007, pp. 701–706 (2007)
11. K.L. Ko, R., S.G. Lee, S., Wah Lee, E.: Business process management (BPM) standards: a survey". Business Process Management Journal. 15(5), pp.744 -791(2009)
12. Wohed, P., Aslst, W. M. P., Dumas, M., Hofstede, A.H.M, Russel, N.,: On the Suitability of BPMN for Business Process Modelling. Business Process Management Lecture Notes in Computer Science. 4102, pp. 161-176 (2006)
13. Web Service Description Language, http://www.w3.org/TR/wsdl.
14. Simple Object Access Protocol, http://www.w3.org/2000/xp/Group/soap
15. UDDI Version 3.0.2, http://uddi.org/pubs/uddi_v3.htm
16. Thomas, E.: SOA Principles of Service Design. Prentice Hall. pp. 608 (2008).
17. Kokash, N., Arbab, F.: Formal Behavioral Modeling and Compliance Analysis for Service-Oriented Systems, International Symposium on Formal Methods for Components and Objects (FMCO'08), Revised Papers, Sophia Antipolis, Vol. 5751 of LNCS, Springer, pp. 21-41 (2008)
18. Christel, B., Sirjani, M., Arbab, F., Rutten, J.: Modeling Component Connectors in Reo by Constraint Automata. Science of Computer Programming, Elsevier, 61(2), pp. 75-113(2006)
19. Jongmans1, S.T.Q., Santini1, F., Sargolzaei, M., Arbab, F., Afsarmanesh, H.: Automatic Code Generation for the Orchestration of Web Services with Reo, submitted to European Conference on Service-Oriented and Cloud Computing, Italy (2012)
20. Sirin, E., Parsia, B.: Bringing Semantics to Web Services with OWL-S (2007)
21. Motahari Nezhad, H., Benatallah, B., Martens, A., Curbera, F., Casati, F.: Semi-automated adaptation of service interactions. In Proceedings of the 16th international conference on World Wide Web (2007)
22. Arbab, F., Krause, C., Maraikar, Z., Moon, Y., Proenca, J.: Modeling, Testing and Executing Reo Connectors with the Eclipse Coordination Tools. in proceedings of the International Workshop on Formal Aspects of Component Software (FACS), pp. 10-12, Malaga(2008)