Towards a Framework for the Integration of Modeling Languages

Svetlana Arifulina

Department of Computer Science, University of Paderborn, Germany**
svetlana.arifulina@uni-paderborn.de
http://is.uni-paderborn.de

Abstract. In software markets of the future, customer-specific software will be developed on demand from distributed software and hardware services available on world-wide markets. Having a request, services have to be automatically discovered and composed. For that purpose, services have to be matched based on their specifications. For the accurate matching, services have to be described comprehensively that requires the integration of different domain-specific languages (DSLs) used for functional, non-functional, and infrastructural properties. Since different service providers use plenty of language dialects to model the same service property, their integration is needed for the matching. In this paper, we propose a framework for integration of DSLs. It is based on a parameterized abstract core language that integrates key concepts needed to describe a service. Parts of the core language can be substituted with concrete DSLs. Thus, the framework serves as a basis for the comprehensive specification and automatic matching of services.

Keywords: Integration of DSLs, Metamodeling, Service Specifications

1 On-The-Fly Computing

In the Collaborative Research Centre 901 "On-The-Fly Computing" (OTF Computing), we develop techniques to *automatically* configure IT services distributed on world-wide markets in an ad hoc manner to fulfill customer-specific requests[1]. On receiving a request, suitable software and hardware services have to be automatically discovered and composed. For that purpose, services descriptions have to be compared with search requests (*matching*), in order to determine whether they fit. After a service composition has been created, its quality has to be analyzed, in order to determine the most suitable solution to the request.

In the remainder of this paper, research questions in OTF Computing are presented in Section 2. Then, preliminary results on the framework and its contributions are described in Section 3. Finally, Section 4 summarizes the proposed method and contains further steps.

** This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901)

[1] For more information refer to http://sfb901.uni-paderborn.de

2 Research Challenges and Issues

One of the research challenges of OTF Computing is the automatic configuration of services. It is based on an appropriate matching mechanism and a quality analysis of composed solutions. The appropriate matching mechanism is based on service specifications that, on the one hand, contain all service properties necessary for their accurate comparison and, on the other hand, not too complicated so that they would hamper the efficiency of this comparison. The quality analysis requires service specifications that can be used to derive properties of a composed service, in order to check whether it functions properly. So, a research issue is a service description that enables the automatic and efficient configuration and quality analysis.

One broadly used language to describe web services is Web Service Description Language (WSDL) [CCMW01]. WSDL allows the description of the service in the form of its operation signatures, but does not provide any expressive means to describe the behavior of the service. Another challenge is that in addition to structure and behavior, non-functional properties and infrastructure information have to be specified as well, in order to enable an accurate matching of service level agreements and execution environments. Approaches like WS-* specifications[2] propose solutions for single properties of services but these is still a lack of an integrated language. So, a research issue is an integration of DSLs for different service properties in a consistent comprehensive specification.

One more research challenge of OTF Computing is the fact that service providers use different DSLs to describe the same service properties. Since it is infeasible to enforce the usage of only one language for all service providers and requesters, the flexibility to use their own DSLs must be preserved. However, these heterogeneous service descriptions still have to be matched. Therefore, different mutual exclusive DSLs used for the same service property have to be integrated as well. This kind of integration is often solved in the literature by creating a syntactical mapping between two DSLs that results in merely syntactical matching. So, a research issue is a semantical integration of DSLs that will also yield a matching mechanism considering semantics.

3 Preliminary Results and Contributions

In this section, we introduce a framework for the integration of modeling languages that addresses the research issues introduced in Section 2. An overview of the framework is shown in Fig. 1. The framework consists of an abstract *core language* that serves as a common basis for the integration of DSLs for a comprehensive specification and for matching. The core language accumulates different service properties necessary for the automatic configuration and quality analysis. It is represented by a metamodel, which in turn comprises further metamodels each modeling a certain service property. Furthermore, the core language is *parameterized*, i.e., its parts can be substituted by concrete DSLs.

[2] For more information refer to `http://www.w3.org/2002/ws/`

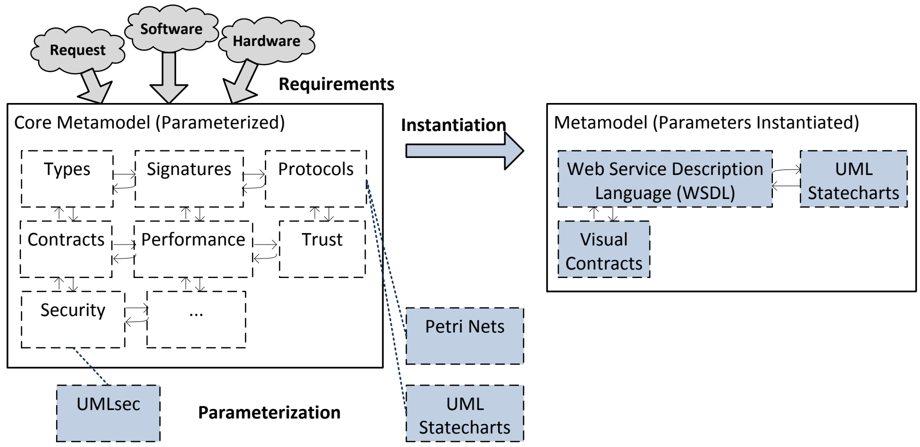

Fig. 1. Framework for the Integration of Modeling Languages

Core Language The core language contains abstract DSL-independent key concepts for each modeled service property. To determine them, requirements from the content of requests and service specifications have to be investigated. Based on DSLs used by domain experts, appropriate language constructs to represent these key concepts have to be identified. Depending on the separation of concerns, these language constructs have to be grouped into packages with explicit relations defined in the form of a metamodel algebra. These relations have to be checked for consistency throughout the core language that shall enable the correct integration of concepts from different part of the core metamodel.

Parameterization The parameterization is a possibility to substitute abstract parts of the core language with concrete syntaxes of DSLs. In Fig. 1, it is illustrated that either UML Statecharts [Obj10] or Petri Nets can be used for the specification of interaction protocols. The parameterization is enabled by defining a mapping between the core and a DSL that shall consider not only their syntax but also their semantics. So, a challenge of this work is to define the formal semantics of the core and develop a systematic approach to create semantical mappings from DSLs onto the core. This research will be based on the preliminary work on the formal definition of behavioral semantics by Soltenborn [SE] and a semantics-preserving mapping of DSLs by Semenyak [EKR$^+$].

Example The process of binding concrete DSLs onto the core is called instantiation. One instance of the core language that can be used for the comprehensive service specification is shown in the right box in Fig. 1. This is a language for rich service descriptions described by Huma in [HGEJ] that allows for service descriptions based on types, signatures, protocols, and contracts.

4 Conclusions and Further Steps

In this paper, a framework for the integration of modeling languages in the context of service-oriented computing is proposed. The framework represents key concepts of service aspects in the form of an abstract parameterized core language that enables to bind concrete DSLs onto it. The main advantages of the framework is the possibility to consistently use different DSLs in one comprehensive service specification and to match comprehensive service specifications written in different languages, both with the consideration of their syntax and semantics. This serves as a basis for the automatic discovery and composition of services in OTF Computing.

My work plan for the PhD is the following:

1. Develop the syntax of the core language that would fit the best for the binding of DSLs and matching of specifications.
2. Develop a formal semantics definition for the core language.
3. Identify packages of the core language and develop a metamodel algebra for relations between them. Check them for consistency.
4. Develop a systematic approach to map concrete DSLs on to the core syntactically and semantically. On the side of semantics, consider mathematical formalisms as well as some pragmatic approaches.
5. Evaluate the results.

The framework will be evaluated during the development of a service specification language in the context of the CRC 901 and by a student group who will build a framework for the composition of mobile applications.

References

[CCMW01] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Service Definition Language (WSDL). Technical report, March 2001.

[EKR+] G. Engels, A. Kleppe, A. Rensink, M. Semenyak, C. Soltenborn, and H. Wehrheim. From UML Activities to TAAL - Towards Behaviour-Preserving Model Transformations. In *Proceedings of the 4th European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA 2008)*, volume 5095 of *LNCS*, pages 95–109. Springer.

[HGEJ] Z. Huma, C. Gerth, G. Engels, and O. Juwig. A UML-based Rich Service Description for Automatic Service Discovery. In *Proceedings of the Forum at the CAiSE'12 Conference on Advanced Information Systems Engineering*, CEUR Workshop Proceedings, pages 90–97. CEUR-WS.org.

[Obj10] Object Management Group, Inc. (OMG). OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.3. Technical report, May 2010.

[SE] C. Soltenborn and G. Engels. Towards Test-Driven Semantics Specification. In *Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems (MODELS 2009)*, volume 5795 of *LNCS*, pages 378–392. Springer.