Position Paper: Applying Model-Driven Engineering for Linking Web Service and Context Models

Georgia M. Kapitsaki
University of Cyprus
75 Kallipoleos Street
P.O. Box 20537, CY-1678, Nicosia,Cyprus
+357 2289 2700
gkapi@cs.ucy.ac.cy

Achilleas Achilleos
University of Cyprus
75 Kallipoleos Street
P.O. Box 20537, CY-1678, Nicosia,Cyprus
+357 2289 2700
aachila@cs.ucy.ac.cy

ABSTRACT

Context-awareness is gaining wide importance, especially when targeting pervasive environments, whereas Web Services are often exploited as building blocks of context-aware applications because of their interoperability properties. In this framework, Web Services can be used as business services requiring context data for their functionality, but also as context sources offering access to context information. Combining the development of context-aware Web Services with Model-Driven Engineering techniques requires the identification and matching of appropriate context models with Web Service models. In this paper, a new ongoing research direction is presented, which refers to model transformation for allowing the identification of appropriate context sources and ensuring their compatibility with Web Services. Matchmaking principles between Web Service and context models are examined in order to extract a reasoning mechanism on context adaptation compatibility. In specific, the principles of graph transformation are mainly exploited.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques – *State diagrams, Modules and interfaces*, D.2.11 [**Software Engineering**]: Software Architecture – *Domain-specific architectures*.

General Terms

Design, Languages.

Keywords

Model-Driven Engineering, Model transformation, Web services, Context-awareness.

1. INTRODUCTION

Model-Driven Engineering (MDE) has been adopted in different application areas in the latest decade. Its use can be found in the field of Service-Oriented Architecture (SOA) and specifically in the development of Web Services (WSs). Web Services constitute the most widely adopted implementation of SOA embraced extensively by the research community and the industry.

iiWAS2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam..

At the same time Web Services are often related with context-awareness, where "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*" [5]. In the context field, Web Services can be exploited as business components that depend on context data. In this case they offer specific functionality (e.g., room reservation, stock options information retrieval) and can be referred to as Business Web Services (BWSs). However, Web Services may also be used as context sources – or Context Web Services (CWSs) – exposing functionality related to the retrieval of context information via appropriate mechanisms, such as sensors and RFIDs spread in the environment. In this way BWSs consume the information provided by CWSs in order to be rendered context-aware. Since both BWSs and CWSs can be represented via models, it is interesting to study their potential combination on the modelling level. Such an examination forms part of the initial steps for the development of applications consisting of WSs through the principles of MDE.

Taking into account the above, the aim of the ongoing work presented in this paper is to use for the first time, to the best of our knowledge, model transformation techniques for linking models towards building context-aware services. In this way, developers of context-aware Web applications can reuse existing Web Services, available on online service registries[1]. In order to perform such a matchmaking procedure the WS and the context information models need to be analysed and combined in an adequate way. Existing research work (e.g., [8], [2]) refers to models that conform to the same semantics; i.e. to the same modelling language. In this work, model matching is examined in cases of models that conform to different semantics but depend on each other. This issue has not been addressed in the MDE literature before. Nevertheless, it constitutes a challenging issue, since it is possible to meet systems with modules representing different aspects of the system with a specific dependency degree among them. In the following sections the main identification principles are presented along with some initial results.

2. BACKGROUND AND MOTIVATION

Works that combine MDE directives with WSs, such as [13] and [15], concentrate on providing modelling specifications between Web Service Definition Language (WSDL) representations and Meta-Object Facility (MOF)-compliant models or ones that follow the Unified Modeling Language (UML) notation. Since

[1] http://webservices.seekda.com/

WSs can be also exploited as building blocks for larger applications or produce composite Web applications, the research interest has also been shifted towards this direction [6]. These works prove the interest of the research community for the combination of context-awareness, WSs and MDE. For more information on related work on context-awareness the reader can refer to this survey on context-aware service engineering [10].

In order to be able to apply MDE in different domains of use, transformations between models that adhere to different meta-models are necessary [1]. Model transformations and respective languages and tools are usually represented through graphs transformations (GTs). The theory of GTs is diverse and covers a wide area of applications (e.g., validation of model transformations, conformance to meta-models). Typed graphs, typed attributed graphs and related graph morphisms are used in graph transformations [4]. Typed attributed graphs $AG = (G,D)$ consist of an E-graph G together with a DSIG-algebra (data signature algebra) and are exploited for the model representation in the following sections. Implementations of model transformation languages are also available in the literature, such as the Model Transformation LAnguage (MOLA), which uses pattern matching and rules defining how the elements of the matched pattern should be transformed [9]. More generic graph transformation tools that can be applied in transformations are also available; e.g., Attributed Graph Grammar System (AGG[2]), Atom3 [3], Viatra [17]. AGG uses graph transformation systems consisting of a type graph and a set of transformation rules, whereas Atom3 is based on the theory of triple graph grammars that uses three separate graphs (source, target and correspondence) to model transformations.

Using as motivation the latest research area of model transformation, a novel application and extension for context-aware Web Services can be derived. Model transformation is usually exploited in order to produce mappings between models that contain the same kind of information (e.g., [8], [2]). In this case the individual models represent information used for the same purpose. This needs to be changed, when model transformations are addressed for purposes of context source identification, since the source and target model do not share the same semantics. Model transformation should and can be applied on models expressing different kind of information, e.g., business service models vs. context information models.

3. LINKING SERVICE AND CONTEXT MODELS

3.1 Compatibility Principles

Using model transformation context information, represented by a context model, is extracted from 1) the BWS models and 2) the CWS models. By performing such an extraction for both the BWS and the CWS, the two context models that follow the same notation can be matched against each other. The most important phase consists in the correct extraction of the context model representations. The transformation of the BWS to a context model represents the context data that the BWS requires for its functionality, whereas the corresponding transformation from the CWS illustrates the context information provided by the CWS. If these two generated target models are compatible, the CWS can be used as input source for the BWS. Please note that a specific Business Web Service may use more than one Context Web Services for context information retrieval. Therefore, more than one CWSs may be used for the extraction of a complete context model that could satisfy the needs of the BWS.

In the current stage of this work it is assumed that all models (i.e., BWS, CWS and context models) follow the UML notation. This is not restrictive; context source identification can be expanded to other MOF-compliant representations. Specifically, in the presentation of the paper the ContextUML [13] meta-model is assumed. More generic UML profiles for context modeling exist in the literature and their combination with model matching principles (e.g., when using context models that adhere to different profiles) consists an interesting study [14]. In previous work of the authors a Web Service profile and a context meta-model, initiating from the principles of ContextUML, have been proposed [11]. These profiles are suitable when context adaptation is applied on Web Services, since they incorporate context adaptation cases usually encountered in WSs, where request and response messages are encapsulated in XML format.

The three adaptation cases captured in ContextUML (presented in detail in [11]) are: 1) Parameter Injection, 2) Operation Selection and 3) Response Manipulation. However, the third case is neglected here since it is not possible to reason on this type of matching based on the information contained in the UML models. The first case is related with the input parameters of the operations of the BWS and the second one with the operations themselves. Model elements corresponding to these properties are matched against each other. The compared elements need to be compatible both in terms of names used and types assigned. The matchmaking procedure can also be performed on the service specification level, i.e. on the WSDL descriptors. The two levels can be seen as supplementary, since they constitute two different views of the same information. Model matchmaking can be performed using model transformation, whereas specification level matching is performed by comparing the respective BWS and CWS descriptions. This case is described in recent work [12].

3.2 Model Transformation Principles

In order to exploit the existing literature on graph transformations, the two parts of the ContextUML meta-model, i.e., a) Web Service part and b) context part, are represented by typed attributed graphs as depicted respectively on the left and on the right part of Figure 1. Node inheritance is used as described in [4] for class diagrams; properties and associations of the super-metaclasses are used directly on the sub-metaclasses. This appears for example on the *name* node attribute edge connecting the *InMessage* and *OutMessage* graph nodes with the *String* data node. Additionally, the *realization* graph edge is used to express the interface realization of UML class diagrams. In the represented graphs no associations between the WS and the context part have been added. These associations correspond to the rules that link the WS graph (source) with the context graph (target). An instance of the typed attributed graph for an example BWS (*AiportInformation*) and a context model expressing *Location* information can be found in the two parts of Figure 2.

Different rules are defined for the transformation a) of the BWS graph to a context graph and b) of the CWS graph to a context graph. The second transformation is performed by mapping each *OutMessage* of the CWS operations to a *Context* element (atomic or composite) along with the mapping of the corresponding Parts

[2] http://user.cs.tu-berlin.de/~gragra/agg

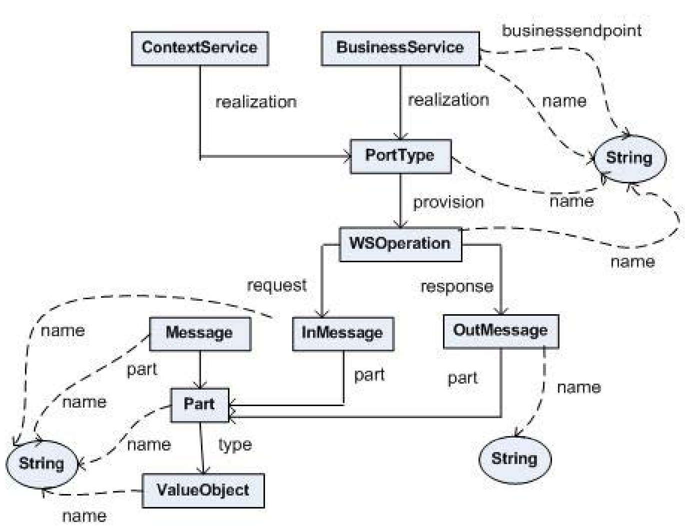

Figure 1. ContextUML meta-model as typed attributed graphs.

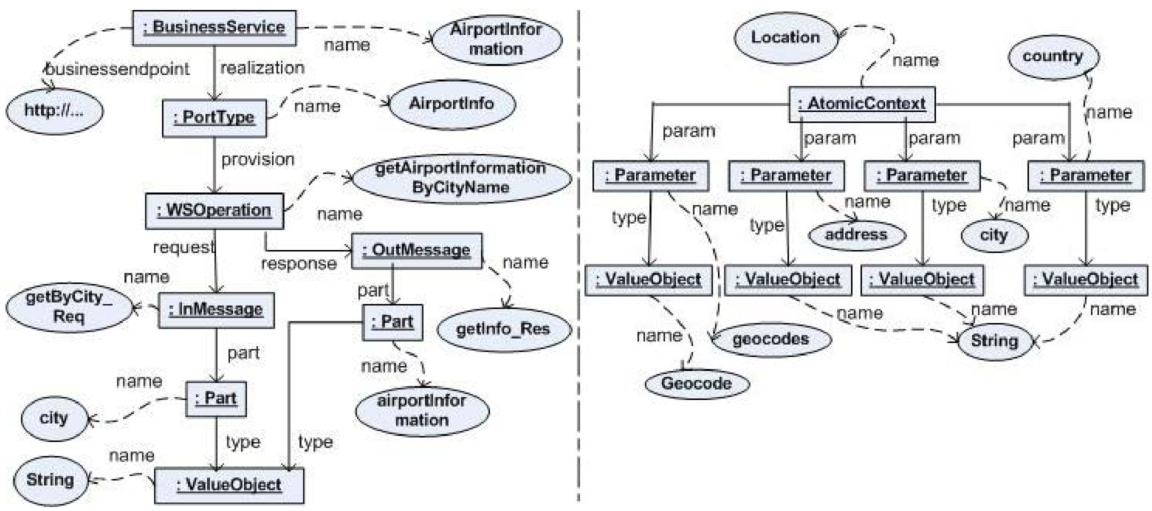

Figure 2. Instances of ContextUML typed attributed graphs.

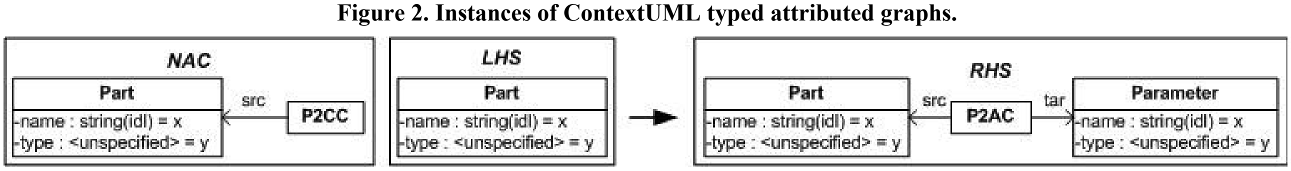

Figure 3. Production example with a NAC.

to context Parameters. The transformation of the BWS is more complicated and includes the 1) determination of context-related model elements and 2) analysis of these elements based on the first two adaptation cases.

For the graph transformation the existence of an adequate graph grammar consisting of a set of productions and a starting graph is necessary. The model transformations between the two models can be defined as a set of productions mapping elements of the Web Service graph to elements of the Context graph. In each production, the left hand side (LHS) corresponds to the source graph and leads to the target graph of the right hand side (RHS), when the correct rule is applied on the graph. Constraints and application conditions play a major role in these transformations [7]. Application conditions specify under which context each application rule is applied and can be divided into: positive application conditions (PAC), negative application conditions (NAC) and general application conditions (GAC) [16]. An example of a NAC is provided in Figure 3 stating that a WS part that has already been mapped to a *CompositeContext* parameter through a *P2CC* transformation rule cannot be mapped additionally to an *AtomicContext* parameter using a *P2AC* transformation (the existence of the reverse NAC in the production set is also necessary). More layers of transformation productions are needed in order to complete the transformation.

The above NAC forms part of the syntactic properties of the transformation. In the case of model linking for context source identification, the main issue is that such a production set is not restricted to a syntactical definition. PACs, NACs and general application conditions can be defined for expressing the transformation rules in a syntactical manner; semantic compatibility, however, requires a different handling. Although syntactic capabilities are related with the correctness requirements of model transformations, the correctness testing can be expanded to functional behavior and semantical correctness [7]. The latter is related with the semantic compatibility of the models, i.e., how to specify if a model element can be regarded as context-related. Semantic compatibility can be enabled through tools that examine the contextual properties of an element. Context ontologies can be useful in that respect.

4. CONCLUSIONS

In this paper the application of MDE principles for linking Web Service and context models for context adaptation purposes has been discussed. Some initial results towards this direction (regarding graph transformations) have been presented. The current stage of work exploits the ContextUML meta-model and respective adaptation cases for context-aware Web Services. The proposed approach is different than existing transformation approaches, which address models with the same semantics.

In order to examine model compatibility further the detailed specification of the necessary models transformations needs to be defined. Other interesting issues are related with applying the approach on real Web Services, which can be obtained from online service registries and repositories. Such an implementation could reveal the practical aspect of the presented work, since it would give the opportunity to software engineers to exploit and combine in their own implementations existing and already tested third party Web Services. The model transformation principles can also be applied on the Eclipse Modelling Framework (EMF), which is in many cases used as an aggregation point for MDE tools by the introduction of plugins for various purposes.

5. REFERENCES

[1] Ahmed Jilani, A. A., Usman, M. and Halim, Z. 2010. Model Transformations in Model Driven Architecture. *Universal J. of Computer Science and Engineering Technology*, 1 1 (Oct. 2010), 50-54, IEEE Computer Society Washington DC, USA.

[2] Cicchetti, A., Di Ruscio, D., Eramo, R. and Pierantonio, A.. 2008. Automating Co-evolution in Model-Driven Engineering. *In Proceedings of the 2008 12th International IEEE Enterprise Distributed Object Computing Conference* (EDOC '08), ACM, New York, NY. DOI=10.1109/EDOC.2008.44

[3] de Lara, J. and Vangheluwe, H. 2002. AToM3: A Tool for Multi-Formalism Modelling and Meta-Modelling. *In Proceedings of the ETAPS/FASE 2002*. LNCS, 2306, Springer, Berlin, 174-188.

[4] de Lara, J., Bardohl, R., Ehrig, H., Ehrig, K., Prange, U. and Taentzer, G. 2007. Attributed Graph Transformation with Node Type Inheritance. *Theor Comput Sci*, 376 3. DOI= 0.1016/j.tcs.2007.02.001

[5] Dey, K and Abowd, G. D. 1999. *Towards a Better Understanding of Context and Context-Awareness*. In GVU technical report GIT-GVU-99-22, 3-4 Georgia Institute of Technology.

[6] Dumez, C., Gaber, J. and Wack, M. 2008. Model-driven engineering of composite web services using UML-S. *In Proceedings of the 10th International Conference on Information Integration and Web-based Applications and Services* (iiWAS '08), ACM, New York, NY, 395-398. DOI= 10.1145/1497308.1497379

[7] Ehrig, H., Ehrig, K., Habel, A. and Pennemann, K. H. 2004. Constraints and Application Conditions: From Graphs to High-Level Structures. In ICGT 2004. LNCS, 3256, Springer, 287-303. DOI= 10.1007/978-3-540-30203-2_21

[8] Hearnden, D., Lawley, M. and Raymond, K. 2006. Incremental Model Transformation for the Evolution of Model-Driven Systems. In Proceedings of the 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS 06), Springer, 4199/2006, 321-335. DOI= 10.1007/11880240_23

[9] Kalnins, A., Barzdins, J. and Celms, E. 2004. Model Transformation Language MOLA. *In Proceedings of the MDAFA 2004 Model-Driven Architecture: Foundations and Applications*.

[10] Kapitsaki, G. M., Prezerakos, G. N., Tselikas N. D. and Venieris, I. S. 2009. Context-aware Service Engineering: A Survey. *J. Syst. Softw.* 82 8 (August 2009), 1285-1297. DOI= 10.1016/j.jss.2009.02.026

[11] Kapitsaki, G. M., Kateros, D. A., Prezerakos, G. N. and Venieris, I. S. 2009. Model-driven Development of Composite Context-aware Web Applications*. Elsevier J. of Information and Software Technology*, 51 8, 1244-1260. DOI= 10.1016/j.infsof.2009.03.002

[12] Kapitsaki, G.M. 2011. Identifying Context Sources Towards Context-aware Adapted Web Services*. In the Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST 11), (Noordwijkerhout, Netherlands, May 5-9, 2011).

[13] Sheng, Q. Z. and Benatallah, B. 2005. ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services. *In Proceedings of the International Conference on Mobile Business* (ICMB'05). IEEE Computer Society, 206-212. DOI=10.1109/ICMB.2005.33

[14] Simons, C. 2007. CMP: A UML Context Modeling Profile for Mobile Distributed Systems. *In Proceedings of the 40th Annual Hawaii International Conference on System Science*, IEEE Computer Society, 289b-289b. DOI= 10.1109/HICSS.2007.125

[15] Skogan, D., Gronmo, R., Solheim, I. and Oldevik, J. 2004. Model-driven web services development. *In Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service* (EEE 04), IEEE Computer Society, Press, Taipei, Taiwan, 42- 45. DOI= 10.1109/EEE.2004.1287288

[16] Tsiolakis, A. 2000. *Consistency Analysis of UML Class and Sequence Diagrams based on Attributed Typed Graphs and their Transformation*. In Technical Report No. 2000/3 Technische Universitaet Berlin.

[17] Varro, D. and Pataricza, A. 2004. Generic and meta-transformations for model transformation engineering. *In Proceedings of the 7th International Conference on the Unified Modeling Language* (UML 2004), LNCS, 3273, Springer, Berlin, 290-304.