MARS: Metamodel Recovery from Multi-Tiered Models Using Grammar Inference*

Qichao Liu[1], Faizan Javed[2], Marjan Mernik[1, 3], Barrett R. Bryant[1],
Jeff Gray[1], Alan Sprague[1] and Dejan Hrnčič[3]
[1]University of Alabama at Birmingham, [2]Regions Financial Corp., [3]University of Maribor
qichao@cis.uab.edu, faizan.javed@regions.com, bryant@cis.uab.edu,
marjan.mernik@uni-mb.si, gray@cis.uab.edu, sprague@cis.uab.edu, dejan.hrncic@uni-mb.si

Abstract

In model-driven engineering, metamodels may get lost over time resulting in the inability to load and view existing model instances. MARS is a system that recovers metamodels from model instances using grammar inference. This paper discusses advances in MARS that improve accuracy and scalability.

Keywords

Metamodel, Grammar Inference, Multi-tiered domain

1. Introduction

The rapid development of model-driven engineering [1] has made modeling become a widely used software development technique. Metamodels define the syntax of models and are needed to load model instances into a modeling tool. As a metamodel undergoes frequent evolution, previous model instances may become orphaned from the new definition. The metamodel may also be lost, resulting in the inability to load and view existing model instances. MARS [2], a metamodel recovery system using grammar inference, was developed to solve this problem.

An overview of MARS is illustrated in Fig. 1. The metamodel inference process begins with the translation of a set of instance models in XML (most modeling tools provide a capability to export a model as an XML file) into a Domain-Specific Language (DSL) called the MRL (Model Representation Language) that filters the accidental complexities of the XML representation of a model in order to capture the essence of the instance models (step 1 in Fig. 1).

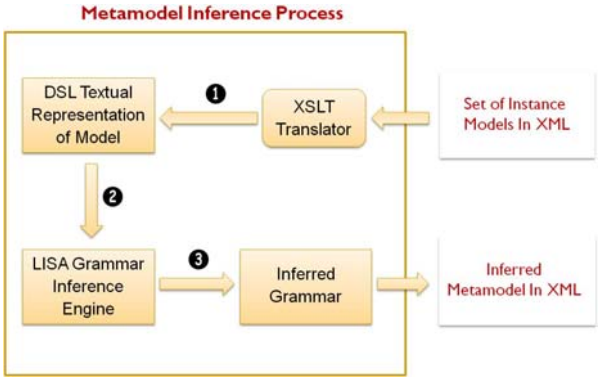

Figure 1 Overview of MARS

The inference is performed within the LISA [3] language description environment (step 2 in Fig. 1) which is an interactive environment for programming language development and was chosen for this project because of its benefits in designing DSLs [4]. Moreover, the LISA system has been used successfully in our evolutionary-based context-free grammar (CFG) inference engine [5]. The result of the inference process is a context-free grammar that is generated and concurrently transformed using formal transformation rules into the inferred metamodel in XML file, which can then be used to load the instance models into the modeling tool (step 3 in Fig. 1).

2. Application of MARS

MARS has been applied successfully to various domains (e.g., Finite State Machines, Petri Nets [6]), and the inferred metamodels are almost exactly the same as the originals. However, the metamodels in our first phase of evaluation are all simple with just a small number of elements, because they are from single-

* This work was supported in part by NSF awards CCF-0643725 and CCF-0811630.

tiered domains. Compared with these domains, multi-tiered domains are more widely used in model-driven engineering, due to being able to represent larger models and enabling users to capture multiple viewpoints instead of a single one; as a result we call these domains multi-tiered. Multi-tiered metamodels allow users to create models with different viewpoints in the domain. In GME (a modeling tool used in the project), each viewpoint is established as a separate folder (a model organization concept in GME) under the "Root Folder" to store its own elements. But for single-tiered domains all elements are placed in the "Root Folder". In this paper, we use ESML (Embedded Systems Modeling Language) [7], a typical multi-tiered domain as our example. The original ESML metamodel has 7 different folders, "Configuration", "Component", "Event", "Interaction", "Interface", "Parameters" and "TNA" to capture 7 viewpoints of the domain.

In order to test the effectiveness of MARS in ESML, we created an instance model using the original ESML metamodel, which captured 4 out of 7 viewpoints. After applying MARS on this instance model we inferred a metamodel, which we found to be different from the original. The inferred metamodel appears simpler (only one instance model was used, which does not exhibit all properties of the ESML domain), but all elements contained in the instance model were inferred accurately. Nevertheless, the inferred metamodel puts all inferred elements directly into the "Root Folder". As mentioned previously, the instance model captured 4 viewpoints of ESML, what we expected to see was a metamodel having inferred elements arranged to accommodate 4 folders.

3. MARS with Extensions

For a more accurate ESML metamodel inference, or multi-tiered domain metamodel inference, MARS was extended in several ways. First, MARS was modified to handle C++-style class scoping rules. Specifically, the scoping rule for the XSLT translator was changed. As a result, the multi-tiered folder information contained in the model instance has been incorporated into the MRL generated by the translator at the beginning of the inference process. A comparison between the DSL generated before and after modification is illustrated in Table 1. For the new MRL, the viewpoint a modeling element belongs to is indicated. In other words, we now know which folder an element came from when it was created in GME. Next, the grammar inference engine was changed correspondingly so the output of the inferred grammar

Table 1 Comparison of old and new MRL

old MRL	new MRL
model model$_1$ { **fields f$_1$, …, f$_n$;** **connections** **con$_1$: src$_1$→dst$_1$;** **…** **con$_n$: src$_n$→dst$_n$;** **}** **atom atom$_1${ fields; }** **…** **atom atom$_n${ fields; }** **model model$_2${** **…** **}** **…**	**model Viewpoint$_m$ :: model$_1$ {** **fields f$_1$, …, f$_n$;** **connections** **Viewpoint$_m$ :: con$_1$: src$_1$→dst$_1$;** **…** **Viewpoint$_m$ :: con$_n$: src$_n$→dst$_n$;** **}** **atom Viewpoint$_m$ :: atom$_1${ fields; }** **…** **atom Viewpoint$_m$ :: atom$_n${ fields; }** **model Viewpoint$_m$ :: model$_2${** **…** **}** **…**

captures all viewpoints contained in the new MRL. Finally, the modified process establishes folders that are related to those viewpoints when the grammar is translated into the inferred metamodel in XML. Elements are placed under the correct folder where they came from instead of in the "Root Folder".

4. Conclusion

This paper describes the limitation of MARS to infer metamodels with multi-tiered domains. The ESML is used as an example modeling language. As a result MARS with extensions is presented here to solve this problem and infer multi-tiered domain models (i.e., ESML models). More details about this project are at http://www.cis.uab.edu/softcom/GrammarInference.

5. References

[1] D. C. Schmidt, "Model-driven engineering", *IEEE Computer Society*, 39(2): February 2006, pp. 25-31.

[2] F. Javed, M. Mernik, J. Gray, B. R. Bryant, "MARS: A metamodel recovery system using grammar inference", *Information and Software Technology*, 50(9-10): August 2008, pp. 948-968.

[3] M. Mernik, M. Lenič, E. Avdičaušević, V. Žumer, "LISA: An interactive environment for programming language development", *Compiler Construction*, Springer, Heidelberg, Germany, January 2002.

[4] M. Mernik, V. Žumer, "Incremental programming language development", *Computer Languages, Systems and Structures*, 31(1): April 2005, pp. 1-16.

[5] M. Črepinšek, M. Mernik, F. Javed, B. R. Bryant, A. Sprague, "Extracting grammar from programs: An evolutionary approach", *ACM SIGPLAN Notices*, 40(4): 2005, pp. 39-46.

[6] J. Peterson, "Petri nets", *ACM Computing Surveys,* 9(3): September 1977, pp. 223-252.

[7] G. Karsai, S. Neema, D. Sharp, "Model-driven architecture for embedded software: A synopsis and an example", *Science of Computer Programming*, 73(1): September 2008, pp. 26-38.