VirtualEMF: a Model Virtualization Tool

Cauê Clasen, Frédéric Jouault, Jordi Cabot

VirtualEMF: a Model Virtualization Tool

Cauê Clasen, Frédéric Jouault, and Jordi Cabot

AtlanMod Team (INRIA, École des Mines de Nantes, LINA) – France
{caue.avila_clasen, frederic.jouault, jordi.cabot}@inria.fr

Abstract. Specification of complex systems involves several heterogeneous and interrelated models. Model composition is a crucial (and complex) modeling activity that allows combining different system perspectives into a single cross-domain view. Current composition solutions fail to fully address the problem, presenting important limitations concerning efficiency, interoperability, and/or synchronization. To cope with these issues, in this demo we introduce VirtualEMF: a model composition tool based on the concept of a virtual model, i.e., a model that do not hold concrete data, but that redirects all its model manipulation operations to the set of base models from which it was generated.

1 Introduction

Complex systems are usually described by means of a large number of interrelated models, each representing a given aspect of the system at a certain abstraction level. Often, the system view a user needs does not correspond to a single model, but is a cross-domain view in which the necessary information is scattered in several models. This integrated view is provided by the means of model composition which is, in its simplest form, a *modeling process that combines two or more input (contributing) models into a single output (composed) model*. Model composition can be very challenging, due to the heterogeneous nature of models and the complex relationships that can exist between them.

Composition has been extensively studied from various perspectives: its formal semantics [2], composition languages [3], or also targeting different families of models (UML [1], Statecharts [4], database models [5], . . .). A commonality of the vast majority of approaches is the fact that the composed model is generated by copying/cloning information from its contributing models, what poses some important limitations in terms of synchronization (updates in the composed model are not propagated to the base ones, or the other way round), creation time (copying many elements is time consuming, and composition must be re-executed every time contributing models are modified), and memory usage (data duplication can be a serious bottleneck when composing large models).

In this demo we present VirtualEMF: a model composition tool that allows overcoming these limitations by applying the concept of *virtual models*, i.e., models that do not hold concrete data (as opposed to *concrete models*), but that access and manipulate the original contributing data contained in other models. The tool was built on top of Eclipse/EMF[1].

[1] *Eclipse Modeling Framework*: http://www.eclipse.org/modeling/emf/

2 Virtual (Composed) Models

In short, a virtual model is a model whose (virtual) elements are proxies to elements contained in other models. It provides to tools/users the *illusion* of working with a regular model while, in fact, all access and manipulation requests are transparently redirected to the set of models from which it was generated (completely integrating contributing model elements into the composed model). Model virtualization brings the following properties to model composition:

- **Interoperability**: virtual models are handled as normal models and therefore they can be exchanged between standard modeling tools;
- **Synchronization**: virtual and contributing models share the same element instances. Thus, updates are automatically propagated in both directions;
- **Faster creation time**: no information cloning is required, and as elements are synchronized, generation is executed only once;
- **Less memory usage**: as virtual models do not contain concrete data, no extra memory is used.

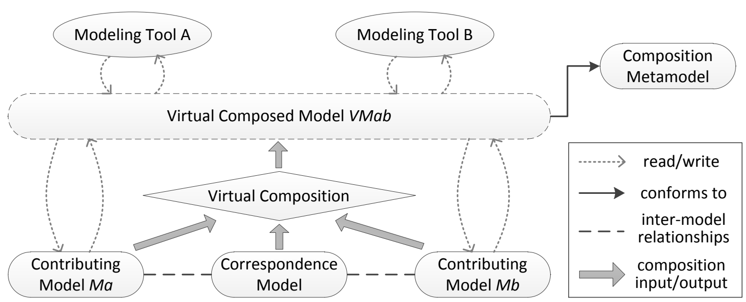

Fig. 1. Model virtualization artefacts.

Fig. 1 introduces the main idea of model virtualization and the involved artefacts. Tools (editors, analysis and transformation tools,. . .) see and use the *virtual model* as a normal model. The virtual model delegates modeling operations to its set of *contributing models*, locating referenced element(s), and translating them into virtual elements to be used by the tool. Contributing elements are composed at runtime, on an on-demand basis.

Contributing elements (and their properties) can be composed/translated into virtual elements in different manners. Some may be filtered; others, simply reproduced. Another possibility is when contributing elements are related to each other and the virtual element is a combination of them (e.g. as in merge or override relationships). A *correspondence model* links contributing elements and identifies which *translation rule* should be used for composing each element.

A virtual composed model conforms to the same *composition metamodel* a concrete composed model would. This composition metamodel states the core concepts that can populate the composed model.

3 The VirtualEMF Tool

The VirtualEMF tool is an Eclipse plug-in built on top of EMF in order to support the transparent usage of virtual models by EMF-based tools. Here we provide a brief explanation on how to create and use virtual models.

To create a virtual model, users must provide, besides the contributing models to be virtualized (and their metamodels), three elements:

1. A **composition metamodel** that specifies the virtual model concepts. It can be manually defined or be the result of a separate composition process;
2. A **correspondence model** (defined with the AMW[2] tool) containing *virtual links* that relate contributing elements and specify how they should be composed (i.e. which translation rule is to be applied to them);
3. A `.virtualmodel` **file** specifying the physical location of the resources involved in the virtual composition process (see Fig. 2).

```
VMab.virtualmodel
compositionMetamodel = {\MMab.ecore}
contributingMetamodels = {\MMa.ecore, \MMb.ecore}
contributingModels = {\Ma.xmi, \Mb.xmi}
correspondenceModel = {\MatoMb.amw}
```

Fig. 2. A sample virtual model file.

The `.virtualmodel` file extension is automatically associated in Eclipse with our specific virtual model handler. Therefore, the loading of a virtual model involves only providing this file as input to any EMF-based tool (e.g., double-clicking a `.virtualmodel` file will automatically trigger VirtualEMF, that loads the virtual model in the standard model viewer). No extra information is required, as the actual data used by the virtual model is retrieved from contributing models and the correspondence model defines how to combine them.

VirtualEMF refines the full set of operations available for a regular model. Thus, usage is also completely transparent (as in Fig. 3). When virtual elements are accessed, VirtualEMF checks which are the referenced contributing element(s), if they are virtually linked, and then translates it(them) accordingly (e.g. with *filter*, *merge*, *override*, *inherit*, or *associate* rules). If no virtual link is specified for a contributing element, it is simply reproduced.

During the demo it will be presented how virtual models are created and used with VirtualEMF, which are its main elements, and how users can define them. This will be shown through several step-by-step examples, with different types of models and specifying different kinds of relationships between them. Finally we will present a set of experiments used to prove that our approach fulfils the desired properties mentioned in section 2.

[2] *AtlanMod Model Weaver*: `http://www.eclipse.org/gmt/amw/`

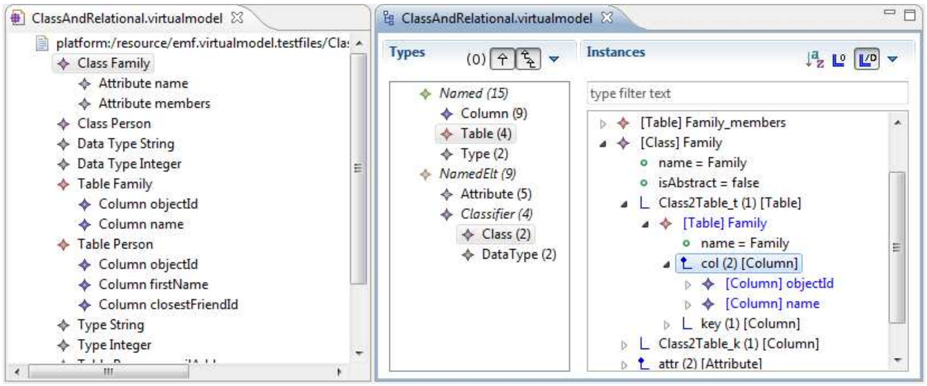

Fig. 3. A virtual model (composition of a UML class model with a relational database model, where the latter derives from the former and virtual associations are used to display traceability links between them) handled in two different EMF tools: Sample Ecore Editor (left) and MoDisco Model Browser (right).

4 Conclusion

Model virtualization is a powerful mechanism that provides a more efficient model composition process, while maintaining perfect synchronization between composition resources. This demo presents VirtualEMF[3], our model virtualization tool. The tool is extensible and supports different types of virtual links and new semantics for them. As further work we intend to explore new types of inter-model relationships, and to use state-of-the-art matching techniques to automatically identify relationships and generate the correspondence model. Several experiments have been conducted to prove the scalability of our solution.

References

1. Anwar, A., Ebersold, S., Coulette, B., Nassar, M., Kriouile, A.: A Rule-Driven Approach for composing Viewpoint-oriented Models. Journal of Object Technology 9(2), 89–114 (2010)
2. Herrmann, C., Krahn, H., Rumpe, B., Schindler, M., Völkel, S.: An Algebraic View on the Semantics of Model Composition. In: ECMDA-FA 2007, LNCS, vol. 4530, pp. 99–113. Springer, Heidelberg (2007)
3. Kolovos, D., Paige, R., Polack, F.: Merging Models with the Epsilon Merging Language (EML). In: MoDELS 2006, LNCS, vol. 4199, pp. 215–229. Springer, Heidelberg (2006)
4. Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S., Zave, P.: Matching and Merging of Statecharts Specifications. In: ICSE 2007. pp. 54–64. IEEE Computer Society (2007)
5. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10, 334–350 (2001)

[3] VirtualEMF website: `www.emn.fr/z-info/atlanmod/index.php/VirtualEMF`.