HAL
archives-ouvertes.fr

Practical meta-model extension for modeling language profiles. An enterprise architecture modeling language extension for telecommunications service creation

Vanea Chiprianov, Yvon Kermarrec, Siegfried Rouvrais

To cite this version:

Vanea Chiprianov, Yvon Kermarrec, Siegfried Rouvrais. Practical meta-model extension for modeling language profiles. An enterprise architecture modeling language extension for telecommunications service creation. 7th Days of Model Driven Engineering, Jun 2011, Lille, France. pp.85 - 91. hal-00623182

HAL Id: hal-00623182

https://hal.archives-ouvertes.fr/hal-00623182

Submitted on 28 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Practical meta-model extension for modeling language profiles

An enterprise architecture modeling language extension for telecommunications service creation

Vanea Chiprianov* — **Yvon Kermarrec*** — **Siegfried Rouvrais**

Institut Telecom, Telecom Bretagne, Université Européenne de Bretagne, France
**UMR CNRS 3192 Lab-STICC*
firstname.lastname@telecom-bretagne.eu

RÉSUMÉ. *En génie logiciel, pour tirer un maximum de bénéfices de l'ingénierie dirigée par les modèles, il est essentiel de permettre les transformations entre ceux-ci. Parmi les nombreux types de transformation, l'extension de méta-model permet notamment de profiler des langages de modélisation afin de les rendre spécifiques à un domaine. Les méta-modèles décrivant la syntaxe de tels langages sont souvent peu complexes. Mais, du fait des imprécisions dues à l'abstraction, l'extension pilotée par un expert est à préférer à une autre approche entièrement automatique. Ainsi, nous proposons dans ce papier trois principes pour étendre, par transformation manuelle, des méta-modèles à l'aide de modèles spécifiques à un domaine. Les profils générés sont rétro-compatibles. L'approche est exemplifiée pour produire un profil d'architecture d'entreprise dédié à la création de services de télécommunications.*

ABSTRACT. *Model Driven Engineering aims at changing the focus from code to models. To achieve it, enabling model transformation is essential. A type of transformation is meta-model extension. It is particularly salient for the use of models in defining Domain Specific Modeling Languages, especially for profiling existing languages. Meta-models describing language syntax have a low number of components. Accordingly, an expert-driven approach to extending meta-models is both practicable and preferable to an automatic one, which has a higher level of inaccuracy. We propose in this paper three principles for aiding an expert in practically extending meta-models with domain specific concepts. The resulted language profiles are backwards-compatible. We apply these principles to defining an ArchiMate profile for telecommunications service creation.*

MOTS-CLÉS : *Transformation de modèles, extension de modèles, langages de modélisation dédiés.*

KEYWORDS: *Model transformation, model extension, domain specific modeling languages.*

1. Model transformation

Model-Driven Engineering (MDE) aims at changing the main focus from code to models. For this, model transformation (e.g., extension, inheritance) (Vallecillo, 2010) must be enabled. As meta-models (MM) are themselves models that are in a conformance relation with their respective meta-MMs, all types of transformation enabled on models can be enabled on MMs as well. Also, for modeling language engineering, MMs are used to describe their syntax. So, transformations on models can be used for performing transformations on modeling languages.

Among types of transformation, in this paper we focus on modeling language extension (i.e. a special form of model extension). Model extension is defined by (Barbero *et al.,* 2007) as the operation that produces the result model - M_r, by applying a mapping ϵ between two models : M_i - initial, and M_f - fragment. The mapping ϵ specifies that, if a node from M_f is equivalent (=) with a node from M_i, the node from M_i will be copied in M_r. If a node from M_f is not equivalent with any node from M_i, a new node will be created in M_r, not necessarily the initial node from M_f. Defined in this way, model extension is a particular case of model combination (also called model composition or weaving) (Vallecillo, 2010).

Model composition has been widely addressed. For example, (France *et al.,* 2007) propose a generic framework for model composition that is independent from a modeling language. The framework is based on an algorithm structured in two steps :

1) *Matching* : identifies model elements that describe the same concepts in the different models that have to be composed ;

2) *Merging* : matched model elements are merged to create new model elements that represent an integrated view of the concepts.

Model merging is kept flexible through pre and post-merge directives. This composition approach is intended for automatic composition of models belonging to separate aspect-oriented views, to build a global view of a system.

(Barbero *et al.,* 2007) note that the node comparison operators $=$ and $\neq$ must be defined in a MM specific way (e.g. if the MM has classes, class names can be used). This is part of the more general issue of model matching (Kolovos *et al.,* 2009), defined as a theoretically NP-hard graph isomorphism problem. Approaches to tackle it automatically, usually consider models as typed attribute graphs and calculate the similarity of nodes based on the combined weighted similarity of their features. However, these methods are intended for large models. Unlike models, MMs are relatively small and strongly influence all models that are conformant with them. This is why, a precise, MM-dedicated approach for comparing and extending them (describing the ϵ mapping) is both possible and preferable to an automatic, general one.

Next, we present a practical, expert-driven approach for MM-dedicated extension, in Sect. 2. It consists of three principles that produce a backwards-compatible profile. We apply it to the ArchiMate business layer MM, which we extend with a telecom MM, in Sect. 3. We discuss the relation with more theoretical approaches in Sect. 4.

2. A practical approach for meta-model extension

The notations introduced by (Barbero *et al.,* 2007) take into account for comparison only nodes. To obtain better matching, edges must be considered too, and even patterns - formations of multiple nodes and edges. Also, only two values (equal, not equal) for equivalence is not enough. A node (edge) from M_f may be equivalent with several nodes (edges) from M_i, but with different degrees. So, we propose using the concept of *similarity* $\sim$ to replace the dichotomic equivalence $=/\neq$. Based on their semantics (definitions/ontologies), the matching of any two nodes (edges) from the two models - M_i and M_f - can be calculated using a similarity measure (continuous - e.g. $\in [0, 1]$), chosen by an expert. If a node (edge) from M_i has a similarity degree, greater than a certain threshold (established by an expert), with another node (edge) from M_f, the two can be considered similar ($\sim$). Now we introduce the principles (p.) (without any claim on completion) that define our approach for MM extension :

1) *Generalization of similarity of nodes*. If a node from M_f is similar with a node from M_i, which is derived from another node (in general, is part of an inheritance hierarchy, at some level), the node from M_f is similar with the parent node of the hierarchy the node from M_i is part of. So, the parent node from M_i is copied to M_r.

2) *Transitivity of similarity for edges (nodes)*. Let's consider a node (edge) (e.g. N_{f1} in Fig. 1) from M_f, that is connected with two edges (nodes) (e.g. E_{f1} and E_{f2}). Let both edges (nodes) have a (high and) close degree of similarity ($\sim$) with an edge (node) from M_i (e.g. E_i). Both edges (nodes) from M_f are similar with the edge (node) from M_i. Only this edge (node) (e.g. E_i) from M_i is copied to M_r.

3) *Pattern matching*. Let's consider several nodes from two models that are similar. If they form a connected graph in M_f, the edges connecting the nodes in M_f have to be similar with the edges connecting the nodes in M_i.

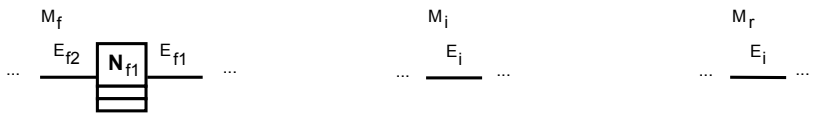

Figure 1. *Illustration of transitivity p.2 for edges ($E_{f1} \sim E_{f2} \sim E_i$).*

We now define the concept of *dissimilarity* $\not\sim$. Two individual nodes (edges) are dissimilar if their similarity degree is under the established threshold. A second possibility for having dissimilarity is when pattern matching p. 3 conditions are not fulfilled.

Priority of principles. The generalization p. 1 should be applied before the transitivity p. 2, as it provides looser node semantics and more possibilities of matching, albeit with a smaller similarity degree. The pattern p. 3 is the most difficult to fulfill (it involves multiple nodes and edges), so it should be applied the last. Semantics of groups (global semantics) is more important than semantics of individual nodes or edges. So (minor) changes may be beneficial in the overall similarity degree of the pattern. Changing the order of principle application may change the obtained results.

3. A telecommunications case study

We apply the proposed three principles to extend the MM of the ArchiMate (The Open Group, 2009) business layer - M_i (Fig. 3 without the nodes marked with a red **T** and the edges that connect them), with the telecommunications specific MM - M_f, Fig. 2, proposed by (Bertin, 2009) :

1) We start by determining the similarity degree of nodes ; *Stakeholder*, *Action*, *ServiceProcess* from Fig. 2 can be considered similar respectively with *BusinessRole*, *BusinessFunction*, *BusinessService* from Fig. 3. Moreover, through the generalization p. 1 between *BusinessFunction* and *BusinessBehaviourElement* from Fig. 3, *Action* is similar to *BusinessBehaviourElement*. The *is done by* edge between *Action* and *Stakeholder* can be considered similar with the edge *assigned to* between *BusinessBehaviourElement* and *BusinessRole* from Fig. 3.

2) We have seen that *ServiceProcess* is the most similar with *BusinessService*. We apply the pattern p. 3. However, the edges that connect it must be taken into account. The only edge is *aggregation* with *Action*. An edge similar with it must be found in the ArchiMate MM. If none is found, a new one will be introduced. The candidate aggregation edges from the ArchiMate MM, that relate to *BusinessBehaviourElement* are : the one with *Product* through the *realizes* edge with *Business Service*, and the one with *BusinessCollaboration* through the *assigned to* edge. The *BusinessColaboration* node has a higher similarity degree with *ServiceProcess* than that of *Product*. Consequently, through the pattern p. 3 and the priority of principles, *ServiceProcess* can be considered similar with *BusinessCollaboration*. However, due to the rather low similarity value, we decide to introduce *ServiceProcess* as a derived node (Fig. 3).

3) *ServiceEntityState* has the highest similarity with *BusinessObject*, but in a low degree. As the *produces* and *uses* edges between *Action* and *ServiceEntityState* from Fig. 2 have a strong similarity degree with the *accessed by* edge between *BusinessBehaviorElement* and *BusinessObject* from Fig. 3, we apply the pattern p. 3 and introduce *ServiceEntityState* as derived from *BusinessObject*. *ServiceEntity* has a low similarity degree with all other nodes from the ArchiMate MM, so it is introduced as a new node, with edges *refers to* and *in relation with*. The result is M_f (Fig. 3 with a red **T**).

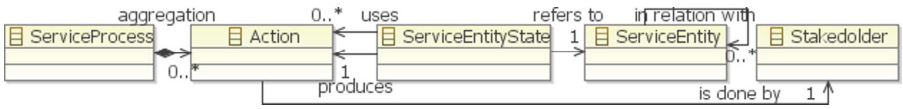

Figure 2. *Meta-model of the telecom reference business view.*

4. Discussion

A special case of model extension, inheritance, is discussed by (Kühne, 2010). It allows adding only nodes (classes) that have a derivation relation with existing nodes. The author identifies three forms of inheritance and investigates properties of forward

and backward compatibility. However, we have found the constraints imposed on these forms too strong in practice. Even if the generalization and transitivity p. 1 and 2 keep the impact of M_f on M_i at a minimum and to enable forward-compatibility, the inherent dissimilarity introduced by domain specificity breaks this type of compatibility. So new models specified with the profile cannot be used with former tools developed for the base language. As this is an important property for language engineers, an approach that guarantees it, while still allows adding new nodes (other than through derivation) is needed. However, as the ArchiMate MM is totally contained in the telecommunications profile MM, we do ensure backward-compatibility. So, former models written in ArchiMate can be used with new tools developed for the profile.

The approach proposed in this paper for meta-model extension is intended for manual application on small meta-models. It can be used of course on the model level as well. It provides guiding rules, ensuring uniformity and the possibility to justify decisions. It can be used by working groups laboring to define language profiles using existing domain models. This approach may be suitable to form the basis of a decision aid system for such groups.

5. Bibliography

Barbero M., Jouault F., Gray J., Bézivin J., « A practical approach to model extension », *Proc. of the 3rd European conf. on Model driven architecture-foundations and applications*, ECMDA-FA, Haifa, Israel, p. 32-42, 2007.

Bertin E., Architecture des services de communication dans un contexte de convergence, PhD thesis, Institut National des Télécommunications et Université Paris 6, 2009.

France R., Fleurey F., Reddy R., Baudry B., Ghosh S., « Providing Support for Model Composition in Metamodels », *Enterprise Distributed Object Computing Conference (EDOC)*, Annapolis, USA, p. 253-264, 2007.

Kühne T., « An Observer-Based Notion of Model Inheritance », *in* , D. Petriu, , N. Rouquette, , O. Haugen (eds), *Model Driven Engineering Languages and Systems (MODELS)*, vol. 6394 of *LNCS*, Oslo, Norway, p. 31-45, 2010.

Kolovos D. S., Di Ruscio D., Pierantonio A., Paige R. F., « Different models for model matching : An analysis of approaches to support model differencing », *Proc. of the ICSE Ws. on Comparison and Versioning of Software Models*, Vancouver, Canada, p. 1-6, 2009.

The Open Group, « ArchiMate 1.0 Specification », 2009.

Vallecillo A., « On the Combination of Domain Specific Modeling Languages », *Proc. of the 6th European Conf. on Modelling Foundations and Applications, ECMFA*, vol. 6138 of *LNCS*, Paris, France, 2010.

6. Annex

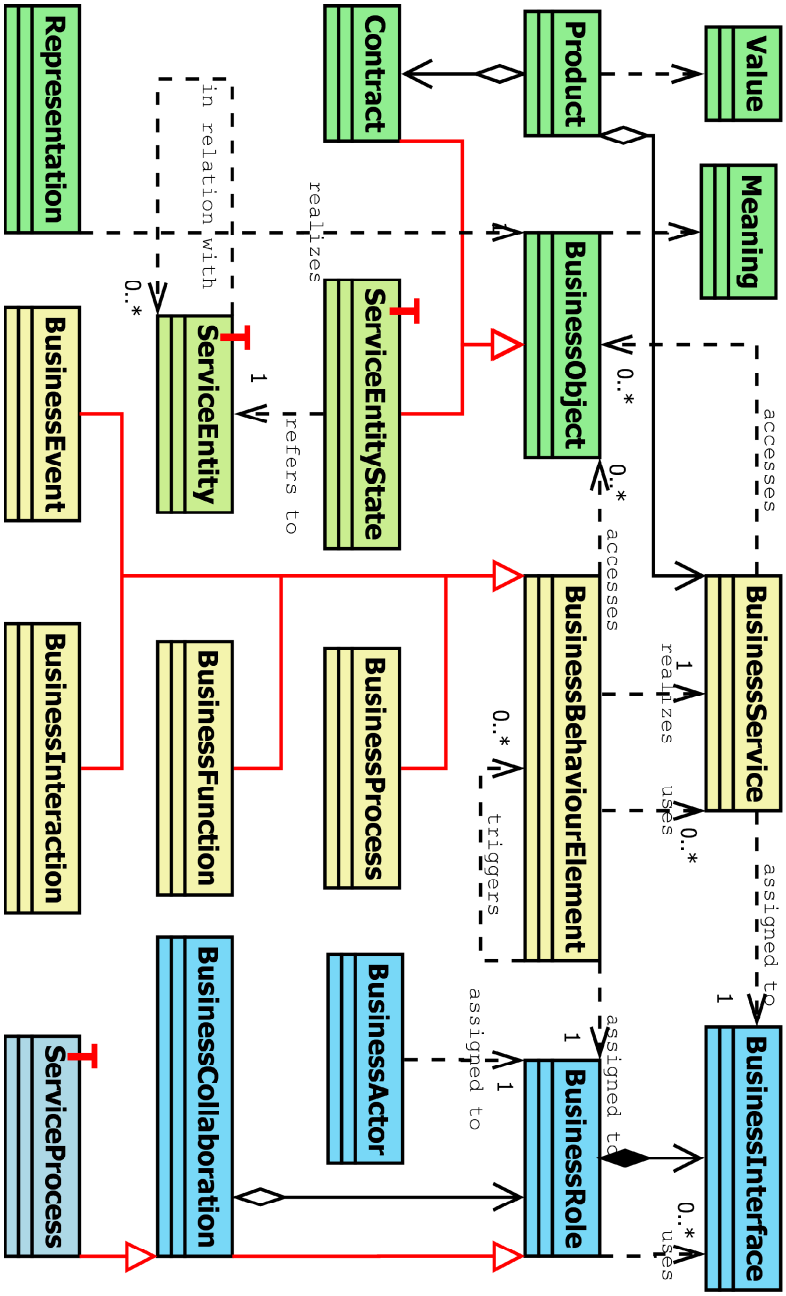

Figure 3. *ArchiMate business view MM extended by a Telecom profile (superscript T).*