Towards Semantic Interoperability of Graphical Domain Specific Modeling Languages for Telecommunications Service Design

Vanea Chiprianov, Yvon Kermarrec
Institut Telecom, Telecom Bretagne
Université européenne de Bretagne
UMR CNRS 3192 Lab-STICC
Technopole Brest Iroise, CS 83818 29238
Brest Cedex 3, France
Vanea.Chiprianov@telecom-bretagne.eu

Siegfried Rouvrais
Institut Telecom, Telecom Bretagne
Université européenne de Bretagne
Technopole Brest Iroise, CS 83818 29238
Brest Cedex 3, France
Yvon.Kermarrec@telecom-bretagne.eu
Siegfried.Rouvrais@telecom-bretagne.eu

***Abstract*—High competition pressures Telecommunications service providers to reduce their concept-to-market time. To manage more easily service complexity among several actors in the design process and to ensure a more flexible maintainability, service decomposition into stakeholder dedicated views is now largely investigated by companies. However, there is still a lack of tools to fully support and implement this approach in various domains, especially Telecommunications. Consequently, in this position paper, we defend using a Domain Specific Modeling Language for each viewpoint. We also regroup them into a family of modeling languages, relying on a meta-modeling approach. To ensure better interaction and coherence between the various viewpoints, we focus on some interoperability issues early at design time. To adequately and systematically manage interoperability between distinct graphical models, interoperability between their meta-models should be established as well. For this we rely on model transformations between meta-models. However, most often model transformations address only the syntactic level. To increase the formality of languages and of their interoperability, semantics must be taken into consideration as well. Therefore, we propose lifting the meta-models into ontologies, enriching and matching them into shared ontologies. This allows for semi-automatic generation of model transformations from shared ontologies.**

***Keywords*-Interoperability, DSML, ontology, semantics.**

I. TELECOMMUNICATIONS SERVICE DESIGN

Every time we call, send text or videos with smartphones, talk using a Skype©-like program or share documents using a secure connection, we are end-users of Telecommunications services (e.g., call, voice over IP, Virtual Private Network (VPN)). These services are delivered by service providers, more and more by operators. They use telecommunications, next generation or computer networks. Traditionally, before a service offers acceptable quality of service and can be launched to a market, it has to pass through several phases (e.g., from design, to implementation, test and deployment). These phases tend to be long and not sufficiently adapted to the current competitive market. More and more companies like Google© and Skype© appear on the service provider market, offering shorter time delivery for innovative services. Consequently, traditional providers are pressured to reduce their concept-to-market time for new services while still maintaining a high level of quality to guarantee a smooth integration with their infrastructure.

A. Viewpoints

To support the increasing complexity of new services and reduce their concept-to-market time, the International Telecommunications Union has introduced the Intelligent Network Conceptual Model (INCM) [1], as ”a framework for the design and description of the Intelligent Network architecture”. It consists of four ”planes”, or views, each refining the service definition from the upper-level plane. More recent proposals, like Enhanced Telecom Operations Map [2] for Telecom, or more general ones, like TOGAF [3] for enterprise architecture, also advocate reducing complexity through division into several layers or views. For greater designer usability, a Domain Specific Modeling Language (DSML) may be defined for each view.

A *Domain Specific Language* (DSL) is ”a language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain” [4]. A *Modeling Language* is, ”a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system” [5]. A *Domain Specific Modeling Language* (DSML) is therefore taken in this position paper to be a graphical language that offers, through appropriate notations and abstractions, expressive power focused on a particular problem domain, to visualize, specify, construct and document the artifacts of a software-intensive system.

A frequent approach to developing DSMLs is the Meta-Modeling approach [6], which defines a DSML as a set of:

- *Concrete syntax*: a human-centric representation of the syntax domain, which defines the symbols used to represent the concepts in the language;
- *Abstract syntax*: a computer-centric representation of the syntax domain;

- *Semantic domain*: the meaning of the language constructs;
- *Display mapping*: links the abstract to the concrete syntax;
- *Semantic mapping*: links abstract syntax to semantic domain.

The concrete and abstract syntaxes are usually defined as Meta-models (MMs). MMs play the same role for DSMLs as grammars for programming languages.

The display and semantic mappings can be defined as Model Transformations (MTs) [7]. A MT is the automatic generation of a target model from a source model, according to a set of transformation rules. A transformation rule is a description of how constructs in the source language can be transformed into constructs in the target language.

The semantic domain is the hardest to define. It may be defined through a semantic mapping towards the precise semantics of an existing programming language [7] so that tools can work on it. Dynamic semantics may be described through operational, denotational or axiomatic frameworks [8] and static semantics through ontologies.

B. Interoperability Issues

To ensure better interaction and coherence between various modeling viewpoints, we focus on interoperability (interop.) issues at design time. One DSML per design view favors in depth control of designers on a particular domain. However, having DSMLs for several views introduces interop. issues between the models designed, in an ideal top-down approach, with adjacent view DSMLs. Therefore, in what follows, we address the issue of ensuring semantic interop. between the models defined with two different DSMLs, which we instantiate to Telecommunications.

II. ON INTEROPERABILITY OF MODELING LANGUAGES

There are numerous definitions for interop. in literature, depending on the domain. For our purposes, and following [9], we consider interoperability to be the ability of two or more *tools* to *exchange models* so as to use them in order to operate effectively together. Considering this definition, to operate together, tools for adjacent view DSMLs need to exchange models. Considering that models are conformant with MMs, and that, in a meta-modeling approach, MMs define (the syntax of) DSMLs (Sect. I-A), the issue of tools exchanging models written in different DSMLs becomes an interop. issue between DSMLs. So, to ensure interop. between models, one must address interop. between DSMLs.

Because interop. is a complex problem, there are numerous proposals of decomposing it into levels. One particularly suitable for our approach is the C4IF (Connection, Communication, Consolidation, Collaboration Interoperability Framework) [9]. This is due to its mapping between Information Systems (IS) Communication and Linguistics. Linguistics and the Meta-Modeling approach (Sect. I-A) share concepts (e.g., syntax, semantics), thus establishing the connection with C4IF. The C4IF defines four levels:

1) *Connection*: the ability of ISs to *exchange signals*.
2) *Communication* refers to the ability of ISs to *exchange data*. *Syntactic* communication includes data in commonly accepted data syntax/schemas.
3) *Consolidation* refers to the ability of ISs to *understand data*. The focus is on data meaning (i.e., *semantics*).
4) *Collaboration* refers to the ability of ISs to *act together*.

These levels of interop. are usually defined in such a manner so as to ensure a (strict) linearity [9] between them - to reach an upper level of interop., all the previous levels must have been successfully addressed.

In order to ensure interop. between two DSMLs we ideally have to ensure all four levels of the C4IF. The ISs of C4IF, in our case, are the tools associated to DSMLs, and the data they exchange, are thus the models. We consider the C4IF connection level as being implemented by existing communication and signaling media in computers.

The mapping proposed in [9] assigns the Communication interop. level of ISs Communication to Syntax of Linguistics. So, communication, *syntactic interop.*, between DSML tools, is the level of interop. between the syntaxes of DSMLs. Approaches to ensure syntactic interop. between different DSMLs have been proposed, like combining MMs [10]: extension, merge, embedding, weaving or hybrid approaches. However, we strongly recognize that the most flexible way to describe relations between two MMs, is through MTs. Using MTs, one can describe the similarity relations between two MMs and capture the intersection between the concepts of their respective DSMLs. Nevertheless, MMs describe only the syntaxes of DSMLs. So MTs, or other combination approaches between MMs, can describe interop. only at a syntactic level.

The mapping proposed in [9] assigns the Consolidation interop. level of ISs Communication to Semantics of Linguistics. So, consolidation, *semantic interop.*, between DSML tools, is the level of interop. between the semantics of DSMLs. We focus in what follows on semantic interop.. We do not yet address collaboration, as we consider, in conformance with their (strict) linearity property, that this level must be ensured first.

III. TOWARDS SEMANTIC INTEROPERABILITY THROUGH ONTOLOGIES

Formal semantic description is significant for the design, reasoning and standardization of programming languages, ensuring their final unambiguous execution or interpretation. It is usually classified into static and dynamic. The frameworks for formal dynamic semantics are usually classified [8] as operational, denotational, or axiomatic. In surveying them, [8] concludes that ”compared to the amount of effort that has been made to the research of various semantic frameworks over more than forty years, their actual applications are definitely frustrating”. Therefore, even if there are approaches using formal semantics to address interop. in a family of DSLs [11], we do not tackle dynamic semantics here. We restrict at static semantics and further investigate ontologies to describe it. Even if ontologies in a broader sense can also define ”dynamic” concepts such as Process, State, Event, they are typically used to describe static concepts, and that is how we use them. We restrict here to using ontologies for static semantics and don’t investigate using ontologies for dynamic semantics.

A. On the use of Ontologies with Meta-models

The common thread in defining ontology [12] is that it is a *formal description* of a domain, intended for *sharing* among different applications, and expressed in a language that can be used for reasoning.

To date, to the best of our knowledge, there is no common agreement on the relationship between MMs and ontologies in the scientific community. While many agree that MMs and ontologies share many and ”deep” characteristics, there are also numerous highlighted differences, and some consider that MMs and ontologies are complementary [13]. Mostly, ontologies have been used with MMs for:

- *Model checking*: using automated reasoning techniques for validation of models in formalized languages.
- *Model enrichment*: expressing the semantics of modeling concepts whose syntax is defined by a MM.
- *Semi-automatic identification of mappings between MMs*: discovering mappings between MMs.

B. Ensuring Semantic Interoperability between Static Semantics of Modeling Languages

We propose to use ontologies for: describing the static semantics of DSMLs (i.e., model enrichment) and discovering a common reference ontology (i.e., semi-automatic identification of mappings between MMs). A common ontology will ensure semantic interop. and coherence between two adjacent view DSMLs. It can be discovered by determining the mapping between two ontologies, each describing the semantics of one DSML. For this, we promote this approach:

1) *Lift*. It transforms each MM into an ontology. We implement it through a MT between the meta-MM describing the modeling technical space (e.g., Ecore[1]) and the meta-MM describing the ontology space (e.g., OWL DL[2]). OWL DL is particularly suited for our approach, as its definition is already given in the form of a MM.

2) *Enrich*. The lifted MMs are enriched by applying patterns. Finding correspondences between relationships of different MMs can be addressed this way. Patterns similar to that of ”Association Class Introduction” [14] can be used. A new class is introduced in the ontology similarly to an association class in UML, thus transforming relationships from MMs into concepts in ontologies. We implement it through an endogenous MT, with input and output the meta-MM describing the ontology space.

3) *Align*. In the ontology technical space we apply ontology-specific techniques [15] (e.g., alignment) on the lifted and enriched MMs of two adjacent views, thus discovering their intersection. Because the lifted and enriched MMs describe semantics of DSMLs, the discovered *shared ontologies* represent in fact the semantics of the MTs between the original MMs. Rediscovering these shared ontologies each time the (lifted and enriched) MMs describing static semantics of DSMLs evolve, is what we mean by ensuring (static) semantic interop. between two DSMLs.

4) *Generate*. MTs which have as input and/or as output other MTs are called Higher Order model Transformations (HOTs). We use shared ontologies as input for HOTs between the meta-MM describing the ontology technical space and the meta-MM describing the MT space (e.g., QVT[3]), which generate MTs between the original MMs.

Consequently, we can automatically generate and evolve MTs for a family of DSMLs, through their connections with shared ontologies, thus ensuring their syntactic and static semantic interop.. The whole process can be automatized and thus enables a high rate of reuse and faster iterations on evolving MMs.

C. Related Work

Kappel et al. [14] propose a process which semi-automatically lifts MMs into ontologies, refactors, enriches, and then applies ontology matching on them. However, unlike our approach, they do not use the discovered matchings to generate MTs. On a more technical point, they implement the lifting step by specifying a weaving model from which they generate ATL code, while we use MTs in QVT.

Hoss and Carver [16] propose connecting MMs with ontologies to assist in software evolution. While they connect MMs with generic ontologies, using what could be called an alignment strategy, we lift MMs into ontologies, using a generative strategy. Also, they have to create model weavings every time new (versions of) MMs are introduced. In our approach, MTs defined between meta-MMs (cf. e.g. Sect. IV) are sufficient for handling any MMs.

IV. TELECOMMUNICATIONS CASE STUDY

Figure 1 exemplifies the proposed approach on two MMs for the adjacent planes/views Global Functional Plane (GFP) and Distributed Functional Plane (DFP) of INCM (i.e., MM_{GFP} and MM_{DFP}). Each MM describes a DSML for VPN at GFP [17] and respectively DFP.

Each MM is *lifted* into an ontology (e.g., O_{GFP} and O_{DFP}) by means of a MT (i.e., $MT_{Ecore2OWLDL}$). This MT is sufficient for lifting any MM into an ontology, as it transforms concepts from Ecore, the language (meta-MM) in which MMs are written, into concepts from OWL DL, the language in which ontologies are written. To write this

---

[1] http://www.eclipse.org/modeling/emf, accessed 24th November 2010

[2] http://www.omg.org/spec/ODM/1.0/, accessed 24th November 2010

[3] http://www.omg.org/spec/QVT/1.0/, accessed 24th November 2010

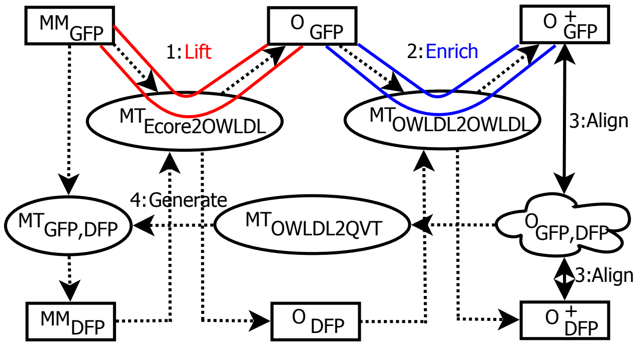

Figure 1. Syntactic and semantic interop. through MTs and ontologies.

MT, we build on the mapping provided by [14], updating it to the new versions of Ecore and OWL DL.

On each lifted MM (e.g., O_{GFP} and O_{DFP}), patterns for refactoring, checking and *enriching* are applied through a MT (i.e., $MT_{OWLDL2OWLDL}$). Similarly to lifting, this MT is sufficient for the enrichment of all lifted MMs.

The enriched ontologies (e.g., O^{+}_{GFP} and O^{+}_{DFP}) are *aligned*, resulting a shared ontology (e.g., $O_{GFP,DFP}$).

With the shared ontology as input, $MT_{OWLDL2QVT}$ *generates* the MT between the initial MMs (e.g., $MT_{GFP,DFP}$). Similarly to lifting and enrichment, this MT is sufficient for the generation of all MTs between the initial MMs.

Currently, we are writing MTs in QVT Relations. For ontology matching, evaluations [18] suggest ASMOV [19] as a good mature candidate tool.

V. DISCUSSION

For Telecommunications, we defend that to manage interoperability between distinct graphical models in a viewpoint approach, interoperability between their meta-models should be established as well. For this we propose using model transformations between meta-models and lifting the meta-models into ontologies. As formulated in this paper, using a meta-modeling approach combined with ontologies has the advantage of co-evolving syntactic and semantic bridges that ensure interoperability between DSMLs. However, this co-evolution depends greatly on the shared ontology between views. If this would be poor or even empty, the interoperability bridge would be narrow. Consequently, in order for the proposed approach to be effective one should first make sure that the vocabularies for different viewpoints have a fair amount of concepts in common. This supports the idea that such an approach would be beneficial especially in the case of families of modeling languages.

REFERENCES

[1] Study Group XVIII, *Principles of Intelligent Network Architecture. ITU-T Recommendation Q.1201*, International Telecommunication Union Std., 1992.

[2] TMF Forum, *Enhanced Telecom Operations Map (eTOM), GB921, Release 8.0*, TMF Forum Std., November 2008.

[3] The Open Group's Architecture Forum, *TOGAF Version 9 Enterprise Edition*, Std.

[4] A. Deursen, P. Klint, and J. Visser, "Domain-specific languages: an annotated bibliography," *SIGPLAN Not.*, vol. 35, no. 6, pp. 26–36, 2000.

[5] G. Booch, J. Rumbaugh, and I. Jacobson, *Unified Modeling Language User Guide*. Reading, MA, USA: Addison-Wesley Professional, 2005.

[6] T. Clark, A. Evans, S. Kent, and P. Sammut, "The MMF approach to engineering object-oriented design languages," in *Wksh. on Language Descriptions, Tools and Applications (LDTA)*, 2001.

[7] I. Kurtev, J. Bezivin, F. Jouault, and P. Valduriez, "Model-based DSL frameworks," in *OOPSLA '06*, 2006, pp. 602–616.

[8] Y. Zhang and B. Xu, "A survey of semantic description frameworks for programming languages," *ACM SIGPLAN Notices*, vol. 39, no. 3, pp. 14–30, March 2004.

[9] V. Peristeras and K. Tarabanis, "The Connection, Communication, Consolidation, Collaboration Interoperability Framework (C4IF) For Information Systems Interoperability," *Intl. Jour. of Interoperability in Business Information Systems*, vol. 1, no. 1, pp. 61–72, 2006.

[10] A. Vallecillo, "On the combination of domain specific modeling languages," in *Proc. of the 6th European Conf. on Modelling Foundations and Applications (ECMFA)*, ser. LNCS, vol. 6138, Paris, France, 2010.

[11] I. Ober, A. Abou Dib, L. Féraud, and C. Percebois, "Towards Interoperability in Component Based Development with a Family of DSLs," in *Proc. of the 2nd Europ. conf. on Software Architecture (ECSA)*, Paphos, Cyprus, 2008, pp. 148–163.

[12] C. Welty, "Ontology research," *AI Magazine*, vol. 24, no. 3, pp. 11–12, 2003.

[13] M.-N. Terrasse, M. Savonnet, E. Leclercq, T. Grison, and G. Becker, "Do we need metamodels AND ontologies for engineering platforms?" in *Proc. of the Intl Wksh on Global integrated Model Management (GaMMa)*. New York, NY, USA: ACM, 2006, pp. 21–28.

[14] G. Kappel, E. Kapsammer, H. Kargl, G. Kramler, T. Reiter, W. Retschitzegger, and M. Wimmer, "Lifting metamodels to ontologies - a step to the semantic integration of modeling languages," in *Proc. of the ACM/IEEE 9th Intl Conf. on Model Driven Engineering Languages and Systems (MoDELS/UML)*, 2006, pp. 528–542.

[15] N. Choi, I. Song, and H. Han, "A survey on ontology mapping," *ACM Sigmod*, vol. 35, no. 3, p. 41, 2006.

[16] A. Hoss and D. Carver, "Towards Combining Ontologies and Model Weaving for the Evolution of Requirements Models," in *Innovations for Req. Analysis. From Stakeholders' Needs to Formal Designs*, ser. LNCS, 2008, vol. 5320, pp. 85–102.

[17] V. Chiprianov, Y. Kermarrec, and P. Alff, "A Model-Driven Approach for Telecommunications Network Services Definition," in *Proc. of the 15th Open European Summer School and IFIP TC6. 6 Wksh on The Internet of the Future*, ser. LNCS, 2009, pp. 199–207.

[18] J. Euzenat, A. Ferrara, C. Meilicke, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Svab-Zamazal, V. Svatek, and C. Trojahn dos Santos, "First results of the Ontology Alignment Evaluation Initiative 2010," in *Proc. of the 5th Intl. Wksh. on Ontology Matching (OM), with the 9th Intl. Semantic Web Conf. (ISWC)*, Shanghai, China, 2010.

[19] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 235 – 251, 2009.