Work in Progress - Modeling Telecommunications Services

Vanea Chiprianov and Yvon Kermarrec

***Abstract*—Present day Telecommunications market imposes a short concept-to-market time for service providers. To reduce this time, we propose a modeling language specific for service designers, with support for collaborative work and for checking properties on models. We started by defining a prototype of the Meta-Model (MM) of the service domain. Using this prototype, we defined a simple graphical modeling language. We are currently enlarging the MM of the domain using model transformations from Network Abstractions Layers (NALs). In the future, we will investigate approaches to ensure the support for collaborative work and for checking properties on models.**

I. Introduction

Present day Telecommunications market, with its high demand rate and fierce competition, imposes a fast pace to service providers. To shorten the concept-to-market time, the service providers need to make their service definition more efficient. The current service definition process is largely based on trays of documents being exchanged between service designers and programmers. To shorten this process, an envisaged solution is to capture the service definition knowledge into a computer-aided design tool, also enabling capitalization on previous experience.

Capturing domain specific knowledge may be done by iteratively constructing a Domain Definition Meta-Model (DDMM), as presented in section II and represented by the central entity in Fig. 1. This DDMM can be further used to replace the current paper-based service definition process by a computer-based modeling language adapted to the needs of service designers. For the definition of this modeling language, we chose to use techniques and tools from the field of Domain Specific Languages (DSL) (see section III). Defining a service is a collaborative work which involves several designers. They need specific support for team work and especially a solution to put together their individual work into a composed definition of the service. We discuss this aspect in section IV, together with the need to verify and validate the definition of a service.

II. Elaborating the Domain Definition Meta-model

We approached the definition of the DDMM with an iterative method in mind. We started with the definition of a simple Meta-Model (MM), for prototype purposes. This prototype is aimed at defining a simple virtual private network (see bottom right green entity in Fig. 1; in this figure we represent with green what we have already done; in parentheses we indicate the toolkit we used). The prototype consists of a *Network*, which may contain several inner networks and several *Node*s. The nodes are either *Computer*s, *Internet* or *Router*s; they are connected by links which constitute outlinks for the source nodes, and inlinks for the target nodes. The routers can be either customer edge routers (*CE*) or provider edge routers (*PE*). Each PE and CE has an *Interface*, which contains a virtual routing and forwarding (*VRF*) table containing the VrfRouteTargets and information about the neighboring PEs (*BgpIpv4AddressFamilyNeighbor*s). PEs use the Border Gateway Protocol (*BgpRoutingProtocol*) to communicate with each other. We also enriched the DDMM with validation rules [1], thus enabling domain level validation. As tool we chose TOPCASED [2], a strongly model oriented system engineering toolkit for critical and embedded applications.

We are currently working on enlarging the DDMM. For this, we start from existing Network Abstraction Layers (NALs) which are specified in UML and simplify them to suit the service designers' needs. We specify the reduction rules using the ATL [3] model transformation language. We chose this approach because a NAL already captures a big part of the service domain, but in a much more detailed manner than necessary for a designer. By eliminating all entities that are unknown to service designers and shrinking the inheritance hierarchies, we believe we can elaborate a MM that is close to the service domain. In addition, such a MM would have the advantage of being easy to map to existing NALs. In the future, we consider enriching the DDMM obtained by reduction from NALs iteratively, by using specific domain analysis methods, such as Family-oriented Abstractions Specification and Translation [4] or Organization Domain Modeling version 2 [5].

III. A Simple Graphical Telecommunications Specific Modeling Language (SGTSML)

If we consider the DDMM corresponding to the abstract syntax of a language (for more details of language definition using model-based tools see [6]), we can define a modeling language specific to service definition. For the concrete syntax (see green entities on the right of Fig. 1) we consider that a graphical syntax will be much easier to use by service designers, as it provides a synthetic, high-level view of the system being considered. Therefore, we defined one using TOPCASED, which has a feature that allows automatic generation of graphical editors for DSLs based on their MM. To describe the semantics of

Vanea Chiprianov is with the LUSSI Department, Institut Telecom, Telecom Bretagne, UMR CNRS 3192 Lab-STICC, Technopole Brest Iroise CS 83818 29238 Brest Cedex 3, France. e-mail: vanea.chiprianov@telecom-bretagne.eu

Yvon Kermarrec is with the LUSSI Department, Institut Telecom, Telecom Bretagne, UMR CNRS 3192 Lab-STICC, Technopole Brest Iroise CS 83818 29238 Brest Cedex 3, France. e-mail: yvon.kermarrec@telecom-bretagne.eu

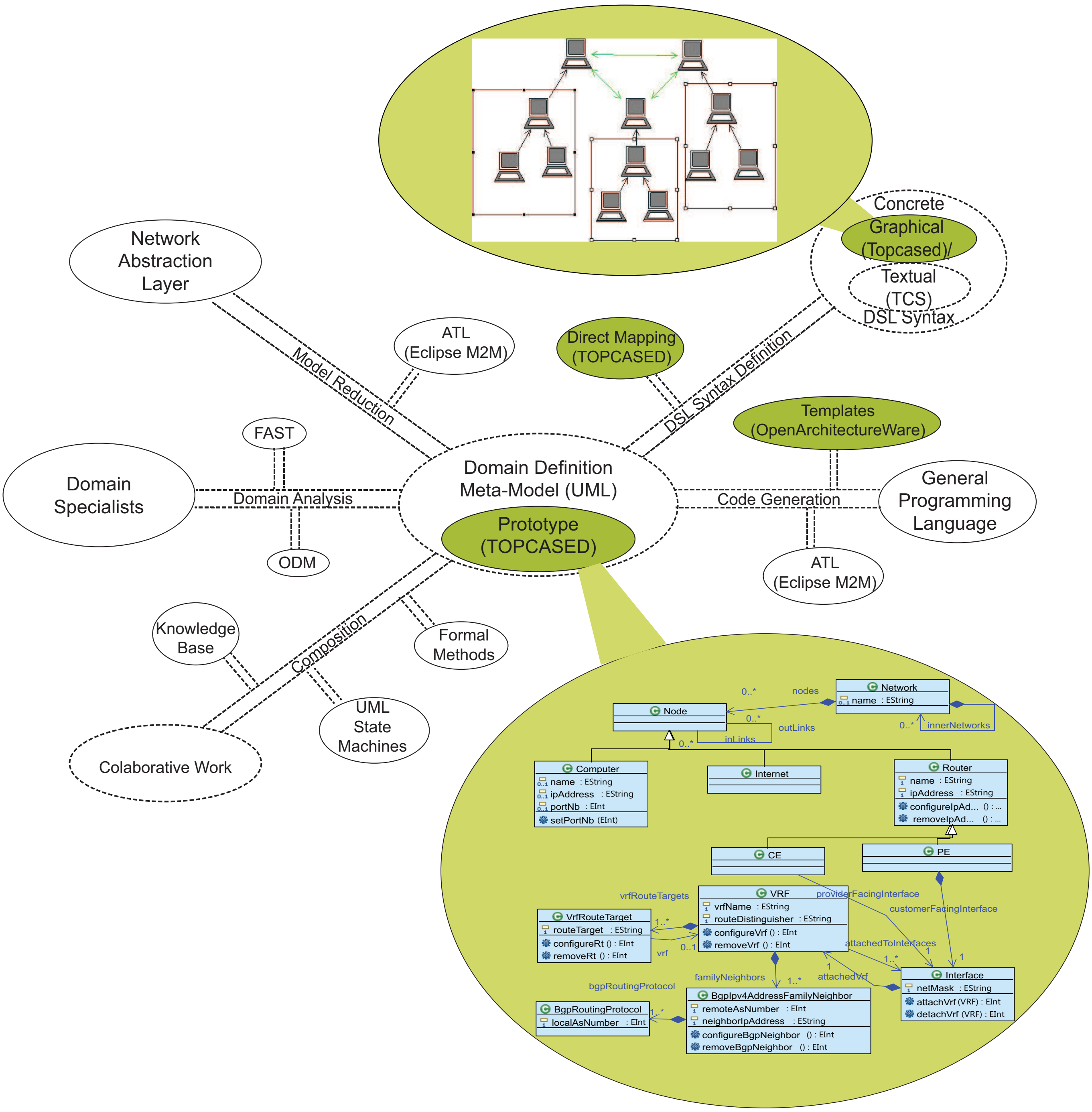

Figure 1
Modeling Telecommunications Services

our SGTSML we decided to use the semantics of an existing general programming language, Smalltalk. Consequently, we defined templates for code generation towards Smalltalk, using OpenArchitectureWare [7]. More details about the definition of SGTSML can be found in [1]. In the future, we intend to extend the concrete graphical syntax to represent the entire enlarged DDMM and to define a concrete textual syntax too, using tools such as TCS [8], or more classical approaches, such as compilers or translators.

IV. Towards Collaborative Work and Checking Properties on Models

Using the modeling language, the designers will define a service. However, because a service is a complex entity, several designers are required to collaborate for its definition. Therefore, we have to provide them with adequate communication and interaction tools. We plan to construct a knowledge base which will contain the decisions that are taken during service definition and their justifications. We are considering also a form of behavior modeling and model composition, but have yet to investigate and decide between UML State Machines, formal methods, ontologies or other approaches.

We are studying as well methods to ensure the models produced by service designers are valid. We envisage defining model transformations from the MM (abstract syntax) of the modeling language towards the MM of formalisms that are capable of verifying a number of properties of interest on the models defined using the modeling language. However, we have yet to identify the properties of interest and the best formalisms to check them.

V. Conclusion

Our purpose is to replace the current paper-based service design process with a more computer-aided version. For this, we prototyped a DDMM and used it to define a SGTSML. We are currently working at enlarging this DDMM through simplifications from existing NALs. In the future, we plan to integrate in the language support for collaborative work and for checking properties on models, but have yet to investigate the best approaches.

References

[1] Vanea Chiprianov and Yvon Kermarrec, “Model-based DSL Frameworks: A Simple Graphical Telecommunications Specific Modeling Language,” in *Actes des 5 émes journées sur l'Ingénierie Dirigée par les Modéles*, 2009.

[2] P. Farail, P. Gaufillet, A. Canals, C. Le Camus, D. Sciamma, P. Michel, X. Cregut, and M. Pantel, “The TOPCASED project: a Toolkit in Open source for Critical Aeronautic Systems Design,” in *ERTS*, 2006.

[3] J. Bezivin, G. Dupe, F. Jouault, G. Pitette, and J.E. Rougui, “First experiments with the ATL model transformation language: Transforming XSLT into XQuery,” in *2nd OOPSLA Workshop on Generative Techniques in the context of Model Driven Architecture*, 2003.

[4] James Coplien, Daniel Hoffman, and David Weiss, “Commonality and Variability in Software Engineering,” *IEEE Softw.*, vol. 15, pp. 37–45, 1998.

[5] M. Simos and J. Anthony, “Weaving the Model Web: A Multi-Modeling Approach to Concepts and Features in Domain Engineering,” in *ICSR '98*, 1998.

[6] Ivan Kurtev, Jean Bezivin, Frederic Jouault, and Patrick Valduriez, “Model-based DSL frameworks,” in *OOPSLA '06:*, 2006, pp. 602–616.

[7] C. Features, “openArchitectureWare 4.2,” Tech. Rep., Eclipse, 2007.

[8] F. Jouault, J. Bezivin, and I. Kurtev, “TCS: a DSL for the specification of textual concrete syntaxes in model engineering,” in *Proceedings of the 5th international conference on Generative programming and component engineering*. ACM Press New York, NY, USA, 2006, pp. 249–254.

Vanea Chiprianov was born in Deva, Romania, in 1983. He received the B.Eng. degree in computer systems engineering from Politehnica University, Timisoara, Romania, in 2007, and the M.S. degree in human machine interaction from Telecom Bretagne 'Grande Ecole', France, in 2008. He is currently a Ph.D. candidate at Telecom Bretagne, where he is researching aspects of model driven engineering under the supervision of Professor Yvon Kermarrec.

Yvon Kermarrec was born in Plouzevede, Finistere, France, in 1962. He received the Ph.D. degree from the University of Rennes I, in 1988, and the 'Habilitation a Diriger des Recherches', from the University of Bretagne Occidentale, in 2005.

He has been involved in research projects related to software engineering, Ada, architecture description languages and has contributed to national, European and international research and development projects. He is Professor of computer science and head of LUSSI department at Telecom Bretagne.