TELECOMMUNICATIONS SERVICE CREATION: TOWARDS EXTENSIONS FOR ENTERPRISE ARCHITECTURE MODELING LANGUAGES

Vanea Chiprianov[*,**], Iyas Alloush[***,*], Yvon Kermarrec[*,**], Siegfried Rouvrais[*]
[*]*Institut Telecom, Telecom Bretagne, Université Européenne de Bretagne*
Technopole Brest Iroise, CS 83818 29238, Brest Cedex 3, France
[**]*UMR CNRS 3192 Lab-STICC*
[***]*Faculty of Information Technology Engineering - Damascus University, Syria*
firstname.lastname@telecom-bretagne.eu

Keywords: Enterprise Architecture, Domain Specific Modeling Language, Model Driven Engineering.

Abstract: From the 90's, the telecommunications service creation industry has undergone radical change. Services have shifted from being based on a switching environment to being mainly based on software. To remain competitive in these new dynamic conditions of an open market, telecommunications organizations need to produce high quality services at low prices within short periods of time. Concerning Service Providers, they need an overall representation of service creation taking in all business, management, and technical activities. To reduce their concept-to-market time for new services, they also need tools specialized for their tasks and domain. In this position paper, we argue that a telecommunications profile for an Enterprise Architecture modeling language answers these needs. We also design a telecommunications profile for ArchiMate that offers conformity to standards through the reuse of a recognized Enterprise Architecture modeling language. Moreover, this profile provides easier adoption by Service Providers due to inclusion of domain specific concepts. The profiling mechanism we propose may be used for defining language extensions specific to other industries as well.

1 SERVICE CREATION CHALLENGES IN A SOFTWARE WORLD

From the 90's, the telecommunications (telecom) service creation industry has undergone radical change (Hållstrand and Martin, 1994). Traditionally, services were offered on equipment from switch manufacturers. The complexity of the switching environment allowed only the switch manufacturer to develop services. Nowadays, services have become more software based. This changes the focus from switching environments to computer environments. To remain competitive in these new dynamic conditions, telecom organizations have to produce high quality services at low prices within short periods of time.

Service creation is a complex activity not only because of the technical activities involved, but also of the number of actors and the difference in each actor's perspective and objectives. To simplify this situation, roles have been identified to allow separation of the different perspectives from the organizations concerned (Hållstrand and Martin, 1994):

- End User: the actual user of a service; primarily interested in service functionality, quality and cost.
- Service Subscriber: the person or organization who contracts and pays for the service. It may need to customize and change services.
- Service Provider: the organization that provides value-added services. The traditional Operator is not the only actor in the role of Service Provider, separate (network independent) organizations also act in this role. The Service Provider is primarily interested in considerations related to the End User (e.g. anticipating and determining requirements for new services, promotion of services in the marketplace) and deployment (e.g. validation of services against requirements, testing of services in the target network, the interaction between new and existing services in the network).
- Service Developer: the organization who performs all the actions necessary to create a new service. They are interested in rapid prototyping and customization of services.
- Network Provider: the organization who operates,

administers and maintains telecom networks; provides bearer and management services.

- Manufacturer: the traditional telecom manufacturer which provides a platform upon which services can execute. They are interested in providing platforms that are open and flexible.
- Tools Vendor: the organization who produces tools which are used by the various actors in the service industry. They are interested in producing tools that cover specific tasks but are adaptable to each organization's needs.

In such a complex stakeholder ecosystem, to assist service creation, a complex software Service Creation Environment is required. We focus in this paper on the role of Service Providers. Their requirements (as identified by (Hållstrand and Martin, 1994)) for Service Creation Environments include:

1. *An overall model* : They are interested in selling and marketing their services. A graphical model describing the service creation process from a market standpoint is useful. This model should include all business planning and economic analysis activities. In addition, network planning should also be represented by a separate model. These models should be integrated with the model of the technical activities involved in development so that an overall model of service creation taking in all business, management, and technical activities is obtained.
2. *Domain specificity* : They want to reduce the concept-to-market time, as the market windows are decreasing from years to months. This means that tools specialized for the task and domain are essential, like those for specifying service functionality.
3. *Rapid prototyping* : They need a rapid prototype of a service, to asses its potential market success.

To answer the Overall model requirement 1, we propose to rely on Enterprise Architecture (EA) frameworks and modeling languages. However, these are general purpose, without any specificity to the tasks and domain of telecom. To answer the Domain specificity requirement 2, we propose defining a Domain Specific Language (DSL), dedicated to telecom specific tasks. To integrate these two approaches, we propose defining the telecom DSL as a profile to an existing EA modeling language. Among the many methods for defining a language, the one which seems to fulfill best the Rapid prototyping requirement 3 of Service Providers is the Meta-modeling approach.

EA frameworks and modeling languages are discussed in Section 2. DSLs are treated in Section 3. The Meta-modeling approach for the definition of DSLs is discussed in Section 4. The proposed telecom profile is presented in Section 5. Perspectives and expected impacts are indicated in Section 7.

2 ENTERPRISE ARCHITECTURE FRAMEWORKS AND MODELING LANGUAGES

EA answers the Overall model requirement 1 of Service Providers. An Enterprise Architecture (EA) approach consists of a set of models describing the structure and functions of an enterprise. These models are organized into levels, from more generic (e.g. objectives) to more specific (e.g. technology used). Integrating all the models from these levels results into an overall model of service creation taking in all business, management, and technical activities. An EA approach is beneficial for: system complexity, agile business alignment with technology platforms, interoperability and integration of constituting systems of an enterprise, common understanding.

However, there are several EA methods and frameworks (e.g., TOGAF, RM-ODP, Zachman Framework, eTOM). The enhanced Telecom Operations Map (eTOM) (TMF Forum, 2008) defines processes and functional areas related to the management of telecom specific data. However, as argued by (Bertin et al., 2010) for example, eTOM focuses on the internal activities of an enterprise for providing services, and it does not cover the core value of a service as seen by the End User.

The Open Group Architecture Framework (TOGAF) (The Open Group, 2009b) provides methods for assisting in the acceptance, production, use and maintenance of an EA. At the core of TOGAF is the Architecture Development Method (ADM), consisting of eight phases and providing one of the most complete (Sessions, 2007) guiding step-by-step process for creating an EA. A comparison between TOGAF and several architecture initiatives (e.g., RM-ODP, Zachman Framework) is provided by (The Open Group, 2007). TOGAF is one of the strongest (Urbaczewski and Mrdalj, 2006) frameworks on the business and technical layers, providing much detail for them. As these layers are most important for our industrial Service Provider partners, we retain TOGAF for our approach.

EA frameworks define processes, guides, methods, etc. They are theoretical in nature, and to apply them modeling languages are needed. There are many modeling languages, both EA specific (e.g., ArchiMate, IDEF, UEML) and for general purpose (e.g. UML). UML is a family of graphical modeling languages allowing structural and behavioral specification. While modeling a telecom service using it is possible, UML is not purposely conceived to offer the EA abstraction levels and relations between them.

One EA specific modeling language is ArchiMate.

It shares (Berrisford and Lankhorst, 2009) important concepts with TOGAF, and thus ”the two are usable together”. It models three phases of TOGAF Architecture Development Method into three layers: Business, Application and Technology. It regulates interoperability between them by defining possible dependencies. Due to its proximity with TOGAF, we retain ArchiMate for our approach.

3 TOWARDS PROFILES FOR ENTERPRISE ARCHITECTURE MODELING LANGUAGES

In this section, we argue the need to extend EA modeling languages with domain specificity at the technical level. As extensions mechanisms we investigate DSMLs and profiles.

An EA modeling language offers the advantage of a unified language, capable of describing a wide range of domains. It makes possible to create integrated models of the enterprise, which can be understood by all stakeholders. While this is very useful at the business level, at the technical level, where more detail is needed to describe a system, this unified language lacks the semantic strength required. By lacking semantic strength we understand that the concepts present in the language are too abstract and they need refinement and specification. To cover this lack, an EA modeling language development method (Khoury, 2007) proposes to use a unified modeling language at the business level, while at more detailed levels use DSMLs and methods.

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” (Deursen et al., 2000). A Modeling Language is, ”a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system” (Booch et al., 2005). 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.

DSMLs allow experts in general, and Service Providers in our case, to express, validate, modify solutions and achieve tasks specific to their domain, as the Domain specificity requirement 2 demands. A DSML requires less cognitive (Green and Petre, 1996) effort from experts than a more general purpose language, as it offers a higher closeness of mapping between the problem world and the program world. Thus quality, productivity, reliability, maintainability, reusability can be enhanced.

In accordance with the EA modeling language development method proposed by (Khoury, 2007), we propose defining a telecom DSML corresponding to the technology level of ArchiMate. This raises the question of integration between the telecom DSML and ArchiMate. To answer it, we propose defining the telecom DSML as a profile to ArchiMate.

A profile (Alhir, 2002) is a generic extension mechanism for customizing reference languages with constructs that are specific to particular domains, platforms. Extension mechanisms allow refining the reference language in a strictly additive manner, so that they can’t contradict standard semantics. Although profiles are usually associated with UML, they can be generalized to other modeling languages as well. We use the term profile in its general meaning (as extension), and not in its UML-bound sense.

Profiles put a strictly additive constraint on extensions to reference languages. This means that profile tools, defined as extensions to existing tools for the reference language, only have to process the additions. There is no need to change in other way the tools for the reference language. The initial cost of DSML tool implementation as profile is lower than that for a standalone DSML. Also, interoperability between several DSMLs, defined as profiles, is facilitated by the reference language. The constructs common between the reference language and each DSML respectively, and the connections between the constructs from the reference language, facilitate defining connections (interoperability) between profiles.

To conclude this section, we highlight the need for EA modeling languages to have stronger semantics to describe in more detail the system under study at lower (e.g. technical) levels of abstraction. As these detailed descriptions are specific to each domain, extensions (profiles) to EA modeling languages, specific to each domain, are necessary.

4 LANGUAGE DEFINITION USING THE META-MODELING APPROACH

Language extensions have to be defined using a formalism and tools have to be provided for them. These concerns are specific rather to Tools Vendors, but choices taken here also influence Service Providers.

There are several methods for defining a language.

Compiler theory uses: abstract and concrete syntaxes, and semantics. It provides (meta-)tools (e.g., lex, yacc) that generate language-specific tools (e.g., lexical, syntactical parsers). However, it deals with textual languages, whereas Service Providers, as indicated by the Overall model requirement 1, need graphical models, so a graphical language.

In the Meta-modeling approach for graphical modeling languages definition (Clark et al., 2001), the abstract and concrete syntaxes from compiler theory are described using meta-models. One way of describing operational semantics is by generating code from the new language to existing (programming) languages, and so producing an executable form of the model. The Meta-modeling approach also provides (meta-)tools (e.g., Eclipse Modeling Framework (EMF), Xpand) that generate language-specific tools (e.g., graphical editor, code generation). This reduces significantly the time, effort and cost of constructing language-specific tools.

In our telecom context, for Tools Vendors, an even more important characteristic of the Meta-modeling approach is its advantage of rapid prototyping. Through the use of meta-tools, language-specific tools can be rapidly generated. Telecommunications services are rapidly evolving and new requirements of Service Providers have to be met by Tools Vendors. When language definition evolves, it impacts the meta-models describing the syntax and the semantics. Due to the generative, meta-tool based approach, language-specific tools can be re-generated to include the new language features, with limited modifications, fast and inexpensive.

The model driven paradigm presents advantages to Service Providers also. Using a Meta-modeling defined language, Service Providers can describe a service graphically, independently of technical details and constraints (or what in Model Driven Engineering - MDE is called a Platform Independent Model (PIM)). Then they can partially generate code to simulate and test the new service for several different platforms (Platform Specific Models (PSMs) in MDE). In this way, Service Providers can rapidly generate prototypes of new services, as the Rapid prototyping requirement 3 demands.

5 A TELECOMMUNICATIONS ARCHIMATE PROFILE

As a consequence to discussions from previous sections, we propose defining a telecom ArchiMate profile relying on a Meta-modeling approach. ArchiMate is already defined using the Meta-modeling approach, and it has a meta-model describing its abstract syntax. It also provides two mechanisms for profiling: adding attributes to ArchiMate concepts and relations, and specialization of concepts. In our approach, we use the specialization mechanism.

For example, Figure 1 presents concepts from the ArchiMate technology layer meta-model (The Open Group, 2009a): 1) Structural: *InfrastructureInterface*, *Node*, *CommunicationPath*, *Network*; 2) Behavioral: *InfrastructureService*; 3) Informational: *Artifact*.

These can be derived using inheritance by telecom IP Multimedia Subsystem (IMS) (3GPP, 2010) specific concepts: 1) Structural: *Proxy-Call/Session Control Function (P-CSCF)*, *PSTN/CS Gateway*, *Media Gateway Control Function (MGCF)*, *Breakout Gateway Control Function (BGCF)*, *ApplicationServer*, *Session Initiation Protocol Application Server (SIPAS)*, *Open Service AccessService Capability Server (OSASCS)*, *IP Multimedia Service Switching Function (IMSSF)*, *Interrogating-CSCF (I-CSCF)*, *Serving-CSCF (S-CSCF)*, *Signaling Gateway (SGW)*, *Subscription Locator Function (SLF)*, *Media Resource Function (MRF)*, *Media Resource Function Controller (MRFC)*, *Home Subscriber Server (HSS)*, *Media Gateway (MGW)*, *Media Resource Function Processor (MRFP)*, which are derived from the ArchiMate *Node* concept, *NodeInterface*, *Protocol*, *Session Initiation Protocol (SIP)*, *Diameter*, *Configuration Access Protocol*; 2) Behavioral: *StartSession*, *Policy Decision Function (PDF)*, *CompressMessage*, *AuthenticateUser*, *SIP Request Check (SipReqCheck)*, *Billing*, *SelectCsNetwork*, *SelectCsGate*, *SendLocationInfoToS-CSCF*, *QueryUserLocation*, *Forward*, *InformHssOfUser*, *ModifySIPMessage*, *AnalyzeFilterCriteria*, *CheckPolicy*, *AnalyzeSIPRequest*, *ExecuteMediaRequest*, which are derived from the *TechnologyFunction* concept.

From the IMS we focused on the Core Network Subsystem, as it contains the essential concepts. For more details on the IMS we refer the reader to (Camarillo and Garcia-Martin, 2008). We will just explain the rationale for choosing the IMS for inclusion into the telecom profile for Service Providers.

In the context of telecom service becoming more software based, the problem of interfacing traditional switch networks with computer networks (e.g. Internet) has received an answer in the form of the IMS architecture. All services are therefore defined for the IMS, as it provides this interface. From there on, services are implemented in one or several types of networks. That is why the concerns of Service Providers stop at the level of detail of the IMS.

Concerning profile tools, we are currently extend-

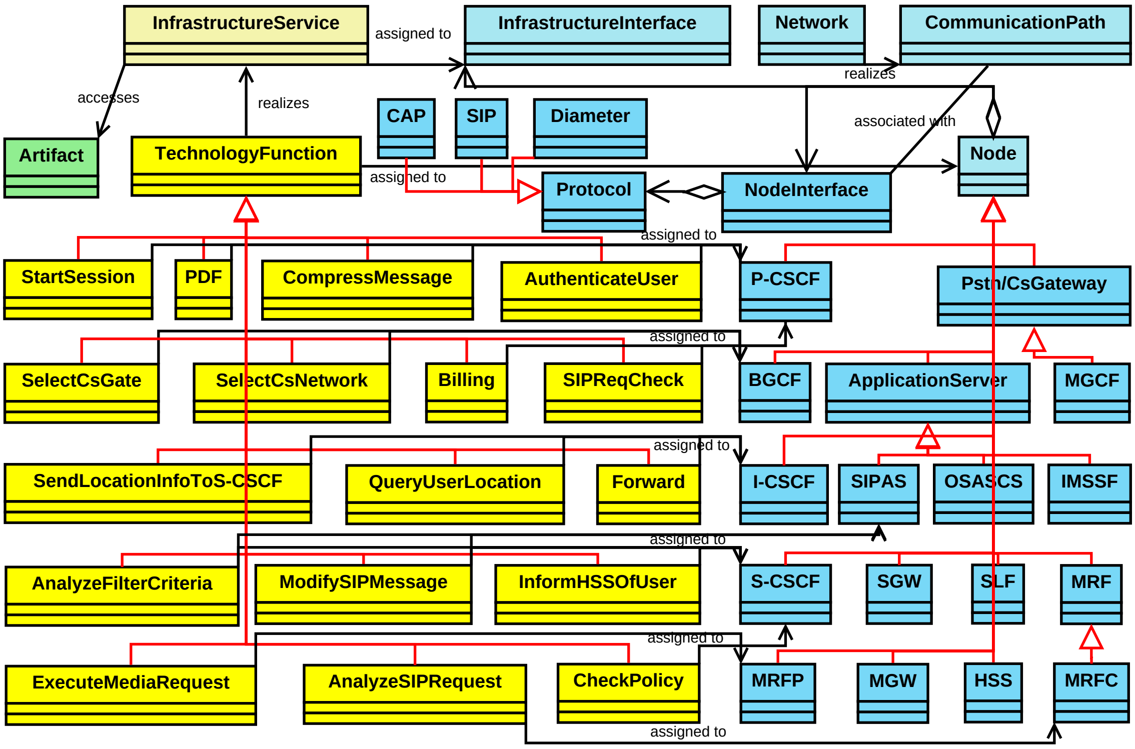

Figure 1: Proposed telecom extension for ArchiMate technology layer meta-model.

ing existing open source tools (e.g. Archi[1] graphical editor) for ArchiMate. Archi is a free, open source, cross-platform editor to create ArchiMate models. Archi is developed as a plug-in for Eclipse 3.6.1 and as such enjoys the possibilities of easy integration with other tools developed as plug-ins.

Eclipse provides an open source platform for creating an extensible Integrated Development Environment (IDE). With the exception of a small runtime kernel, everything in Eclipse is considered as a plug-in. This platform allows building tools that integrate seamlessly with the environment and other tools. We leverage Eclipse as the framework in which we integrate all tools for the telecom profile.

Together with the profile editor, we use in Eclipse plug-ins to implement the profile semantics. For this, we are currently generating code to Java using Xpand (Efftinge and Kadura, 2006). The plug-in architecture enables us to easily add new tools (e.g. for simulating and testing new services).

6 RELATED WORK

MEGAF (Hilliard et al., 2010) is an infrastructure for realizing architecture frameworks. It is an extensible repository of viewpoints, views, model kinds, architecture models, system concerns, and stakeholders. Based on a megamodeling approach, it enables the architect to express and enforce relations between elements inside an architecture description and across architecture descriptions. However, it is a generic infrastructure, in which specific architecture frameworks and languages have to be defined before they can be used, while our approach starts from existing frameworks, languages and tools, and extends them.

There are several proposals for telecom service creation. For example, (Blum et al., 2009) describe a Service Creation Environment based on MDE, intended for orchestrated real-time communications services, through a service broker, on top of Next Generation Networks. However, their approach is focused on composition of services, while our proposal, being based on EA frameworks and languages, offers an overall representation of service creation.

There are several ArchiMate proposed extensions. For example, one (Quartel et al., 2009) introduces an extension for modeling and managing motivation, principles and requirements in TOGAF. Another (Jonkers et al., 2010) argues for an extension for modeling TOGAFs implementation and migration phases. However, these extensions propose adding concepts and relations in a non strictly additive manner, while our approach is intended for strictly additive extensions (i.e. profiles).

[1] `http://archi.cetis.ac.uk/index.html`, accessed on 13.05.2011

7 CONCLUSION AND FUTURE CHALLENGES

In this position paper we proposed to answer the main requirements of Service Providers for telecommunications service creation with a domain specific profile for ArchiMate. We introduced a meta-model extension for the definition of this profile. Applying a meta-modeling approach, we are currently developing profile-specific tools (graphical editor and code generation), extending existing tools for ArchiMate.

There still remain requirements of Service Providers that we did not yet address in this paper. Simulating the behavior of a new service is essential to demonstrate its capabilities to decision makers. Testing the behavior of a new service before deployment and especially its interaction with existing services is important as well. This means that new tools have to be integrated with those being developed now for the profile. We enable this future integration by using a plug-in architecture, which allows new tools to be developed and more easily integrated.

By presenting in detail the case of a telecommunications profile, this position paper also advocates the need of Enterprise Architecture modeling languages for more specificity and higher degree of detail at lower levels of abstraction. In the future, defining Enterprise Architecture modeling language extensions (profiles) specific for other domains may be envisaged. To enable it, we hope to raise awareness among Enterprise Architecture modeling languages tool providers about this need, so that they support extension mechanisms.

REFERENCES

3GPP (2010). 3GPP TS 23.228 V10.3.1 IP Multimedia Subsystem (IMS) Stage 2 (Release 10).

Alhir, S. S. (2002). *A Guide to Successfully Applying the UML*. Springer-Verlag New York, Inc.

Berrisford, G. and Lankhorst, M. (2009). Using ArchiMate with TOGAF–Part 1: Answers to nine general questions about methods. *Via Nova Architectura, https://doc.novay.nl/dsweb/Get/Document-101474*.

Bertin, E., Bécot, S., and Nedelec, R. (2010). Applying enterprise architecture principles to telco services. In *14th Intl Conf. on Intelligence in Next Generation Networks (ICIN)*, pages 1–6, Berlin, Germany.

Blum, N., Magedanz, T., and Margaria, T. (2009). Rapid service creation using eXtreme Model Driven Design for real-time communications services on top of Next Generation Networks. In *13th Intl Conf. on Intelligence in Next Generation Networks (ICIN)*, pages 1 –6, Bordeaux, France.

Booch, G., Rumbaugh, J., and Jacobson, I. (2005). *Unified Modeling Language User Guide*. Addison-Wesley Professional, Reading, MA, USA.

Camarillo, G. and Garcia-Martin, M. A. (2008). *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*. John Wiley & Sons Ltd.

Clark, T., Evans, A., Kent, S., and Sammut, P. (2001). The MMF approach to engineering object-oriented design languages. In *Ws. on Language Descriptions, Tools and Applications (LDTA)*, Genova, Italy.

Deursen, A. V., Klint, P., and Visser, J. (2000). Domain-specific languages: an annotated bibliography. *SIGPLAN Not.*, 35(6):26–36.

Efftinge, S. and Kadura, C. (2006). OpenArchitectureWare 4.1 Xpand Language Reference. Technical report.

Green, T. and Petre, M. (1996). Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages and Computing*, 7(2):131–174.

Hållstrand, J. and Martin, D. (1994). Industrial requirements on a service creation environment. In *Proc. of the 2nd Intl Conf. on Intelligence in Broadband Services and Networks: Towards a Pan-European Telecommunication Service Infrastructure*, pages 17–25, London, UK.

Hilliard, R., Malavolta, I., Muccini, H., and Pelliccione, P. (2010). Realizing architecture frameworks through megamodelling techniques. In *Proc. of the IEEE/ACM intl conf. on Automated software engineering (ASE)*, pages 305–308, Antwerp, Belgium.

Jonkers, H., van den Berg, H., Iacob, M. E., and Quartel, D. (2010). ArchiMate Extension for Modeling the TOGAF Implementation and Migration Phases. Technical report, The Open Group, Catalog number W111.

Khoury, G. R. (2007). *A unified approach to enterprise architecture modelling*. PhD thesis, University of Technology, Sydney.

Quartel, D., Engelsman, W., Jonkers, H., and van Sinderen, M. (2009). A Goal-Oriented Requirements Modelling Language for Enterprise Architecture. In *IEEE Intl Enterprise Distributed Object Computing Conf. (EDOC)*, pages 3 –13, Auckland, New Zealand.

Sessions, R. (2007). Comparison of the top four enterprise architecture methodologies. Technical report, ObjectWatch, Inc.

The Open Group (2007). TOGAF Version 8.1.1.

The Open Group (2009a). ArchiMate 1.0 Specification.

The Open Group (2009b). TOGAF Version 9.

TMF Forum (2008). Enhanced Telecom Operations Map (eTOM), GB921, Release 8.0.

Urbaczewski, L. and Mrdalj, S. (2006). A comparison of enterprise architecture frameworks. *Issues in Information Systems*, 7(2):18–23.