# Chapter 28
# Integrating DSLs into a Software Engineering Process: Application to Collaborative Construction of Telecom Services

**Vanea Chiprianov**
*Telecom Bretagne, France*

**Yvon Kermarrec**
*Telecom Bretagne, France*

**Siegfried Rouvrais**
*Telecom Bretagne, France*

## ABSTRACT

*The development of large and complex systems involves many people, stakeholders. Engineeringly speaking, one way to control this complexity is by designing and analyzing the system from different perspectives. For each perspective, stakeholders benefit from means, tools, languages, specific to their activity domain. A Domain Specific Language (DSL) per perspective is such a dedicated means. While DSLs are used for modeling, other means, tools, and languages are needed for other connected activities, like testing or collaborating. However, using such different types of tools together, integrating DSLs into stakeholders' software process is not straightforward. In this chapter, the authors advance an integration process of DSLs with other tools. The chapter proposes each stakeholder have their own DSL with associated graphical editor, operational semantics, and generation of scripts for off the shelf simulators, e.g., testing. Additionally to the integrated stakeholders' software process, the authors introduce a model driven process dedicated to the tool vendor which creates the DSLs and its associated tools. Due to the integration of DSLs into this process, they contend that stakeholders will significantly reduce system construction time. The chapter illustrates the two processes on Telecommunications service construction.*

DOI: 10.4018/978-1-4666-4301-7.ch028

## INTRODUCTION

The development of large and complex systems involves many people, stakeholders, each with their own perspective, viewpoint. To document, understand and master this complexity, models are an important means. Another manner is by separation of concerns, i.e. distributed definition of specifications, of models, from multiple perspectives, viewpoints. In ISO/IEC 42010 (2007), a viewpoint is defined as a ''work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns''. According to the same standard, a view is defined as a ''work product expressing the architecture of a system from the perspective of specific system concerns''. Within the same standard, a concern is an ''interest in a system relevant to one or more stakeholders'', where a stakeholder is an ''individual, team, organization, or classes thereof, having an interest in a system''.

One approach that handles system complexity is *Model Driven Engineering (MDE)*. The main artifact of MDE is the model. *Models* are representations of the reality for a given purpose. One way to manage complexity of a system is to describe it from different perspectives, using a Domain Specific Language (DSL) for each viewpoint as a modeling means. MDE also proposes a *meta-modeling approach for language definition* (Clark et al., 2001), targeted to languages that allow specifying models. It is frequently used to define graphical DSLs. As such, 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, Klint & Visser, 2000). DSLs can enhance quality, productivity, reliability, maintainability, re-usability, flexibility (Baker, Loh & Weil, 2005) (Kieburtz et al., 1996). Hence, it is advantageous they were used.

Although DSLs are numerous and their numbers are increasing, they address only part of modeler's activities. We consider here modelers to be system and software architects and designers. In practice, there are legacy or mature, domain specific, off the shelf (i.e. software that other software projects can reuse and integrate into their own products) tools that are used regularly by modelers (e.g., for testing purposes, for communication and collaboration). Therefore, DSLs should be conceived as part of the larger activities of a modeler and integrated into his/her process of work. A first question that needs addressing is: *How do DSLs integrate with other tools used by modelers in their regular activities?* Moreover, the tool vendors need to integrate the DSL development process into their tool building process. As such, *How do they integrate the DSL development process into their tool building process?* As they are focused on specific, narrow domains, it is expected more than one DSL be used at one time. How about composing several DSLs? There can be made a distinction between system/product DSLs, used to describe the product under construction, and transversal DSLs, that are not specific to the domain, but rather to connected activities (e.g., describing decision rationale, model versioning).

One example of complex systems are telecom services. They have a long life-cycle, with many stakeholders intervening at its different phases. Recent challenges (e.g. convergence with the Internet, the telecom market deregulation) have determined former national telecom providers to investigate ways to reduce the creation time of new services, while affecting non-negatively other parameters like cost, quality of service. Modeling telecom services using MDE and DSLs constitute a promising investigation path for reducing this time. However, telecom stakeholders already have a wealth of tools specific to their domain. To be successful, to increase their chances of being accepted by professionals, DSLs must integrate as seemingly as possible with these tools. This integration is at two levels: the one of usage, but also the level of development.