Chapter 7
Cloud-Based Multi-View Modeling Environments

Jonathan Corley
University of Alabama, USA

Eugene Syriani
University of Montreal, Canada

Huseyin Ergin
University of Alabama, USA

Simon Van Mierlo
University of Antwerp, Belgium

ABSTRACT
Complex systems typically involve many stakeholder groups working in a coordinated manner on different aspects of a system. In Model-Driven Engineering (MDE), stakeholders work on models in order to design, transform, simulate, and analyze the system. Therefore, there is a growing need for collaborative platforms for modelers to work together. A cloud-based system allows them to concurrently work together. This chapter presents the challenges for building such environments. It also presents the architecture of a cloud-based multi-view modeling environment based on AToMPM.

INTRODUCTION

Complex systems engineering typically involves many stakeholder groups working in a coordinated manner on different aspects of a system. Each aspect addresses a specific set of system concerns and is associated with a domain space consisting of problem or solution concepts described using specialized terminology. Therefore engineers express their models in different domain-specific languages (DSL) to work with abstractions expressed in domain-specific terms (Combemale, Deantoni, Baudry, France, Jézéquel, & Gray, 2014).

Model-Driven Engineering (MDE) (Stahl, Voelter, & Czarnecki, 2006) is considered a well-established software development approach that uses abstraction to bridge the gap between the problem domain and the software implementation. MDE uses models to specify, simulate, test, verify, and generate code for applications. A model represents an abstraction of a real system, capturing some of its essential properties, to reduce accidental complexity present in the technical space. A model conforms to a metamodel (Kühne, 2006), which defines the abstract syntax of a DSL. The metamodel specifies the permissible concepts, relations and properties that models conforming to the metamodel can have. Models are rep-
represented with a concrete syntax (graphical or textual) which defines the notations used to represent each model element. MDE activities typically include the development of modeling languages, the design of models, the implementation of model transformations, the run-time execution of models, and the analysis of models. Several modeling tools exist today, such as AToMPM (Syriani, Vangheluwe, Mannadiar, Hansen, Van Mierlo, & Ergin, 2013), EMF (Steinberg, Budinsky, Paternostro, & Merks, 2008), GME (Ledeczi, et al., 2001), and MetaEdit+ (Kelly, Lytinen, & Rossi, 1996).

Recently, there has been a growing trend toward collaborative environments especially those utilizing browser-based interfaces. Common tools include Google Docs (Google Inc., 2015), Trello (Trello Inc., 2015), Asana (Asana, 2015), and more. Additionally, this trend can be seen in software development tools including WebGME, a web-based collaborative modeling version of GME (Maróti, et al., 2014) and Eclipse webIDE (The Eclipse Foundation, 2015). These tools bring together developers, including geographically distributed teams, in a collaborative development environment to work on a shared set of software artifacts. However, the introduction of these collaborative environments bring new concerns.

In this chapter, we address the need to provide a collaborative environment for domain-specific modeling. Furthermore, in order to ensure consistency and synchronization among the artifacts produced by each stakeholder, we favor a cloud-based environment. Although there is a growing need for such environments, few modeling tools allow multiple stakeholders to work on their modeled system concurrently. In this chapter, we first define the requirement and challenges for a collaborative modeling environment where we enumerate the possible collaboration scenarios. We then present our tool AToMPM, which was designed for collaborative modeling in the cloud, and describe the detailed architecture of how AToMPM solves the challenges of the collaboration scenarios. Additionally, we present competing approaches to ensure consistency and synchronization among shared artifacts with discussion of how the competing approaches differ from the approach applied in AToMPM.

**BACKGROUND**

Online collaboration tools are very popular with the rise of new HTML5 technologies. Offerings such as Google Docs (Google Inc., 2015), Trello (Trello Inc., 2015), Asana (Asana, 2015) and many others take advantage of sophisticated features of new web technologies. They enable users to accomplish tasks without the need for a native client. Modern modeling tools are primarily native desktop applications, e.g., EMF (Steinberg, Budinsky, Paternostro, & Merks, 2008) and MetaEdit+ (Kelly, Lytinen, & Rossi, 1996). AToMPM is the first web-based collaboration tool for model-driven engineering. It takes advantage of the ever increasing capabilities of web technologies to provide a purely in-browser interface for multi-paradigm modeling activities. Nevertheless, Clooca (Hiya, Hisazumi, Fukuda, & Nakanishi, 2013) is a web-based modeling environment that lets the user creates domain-specific modeling languages and code generators. However, it does not provide any collaboration support.

Recently, Maroti et al. proposed WebGME, a web-based collaborative modeling version of GME (Maróti, et al., 2014). WebGME offers a collaboration where each user can share the same model and work on it. It has a Mongo database server as a backend for model repository. In contrast to the modelverse, which is a specialized modeling-optimized repository, WebGME uses a simple branching scheme with a generic document-based NoSQL database to manage the actions of different users on the same