



HAL
open science

GenMyModel: An Online UML Case Tool

Michel Dirix, Alexis Muller, Vincent Aranega

► **To cite this version:**

Michel Dirix, Alexis Muller, Vincent Aranega. GenMyModel: An Online UML Case Tool. ECOOP, 2013, Montpellier, France. hal-01251417

HAL Id: hal-01251417

<https://hal.science/hal-01251417>

Submitted on 6 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GenMyModel : An Online UML Case Tool

Michel Dirix, Alexis Muller, and Vincent Aranega

Axellience 59000 Lille - France

{michel.dirix,alexis.muller,vincent.aranega}@axellience.com

Abstract. Costs and markets lead engineering teams to collaborate from different locations all over the world. Modelling tools are present in development processes to produce complex software and these tools have to be highly collaborative to permit teams to be productive. Axellience tries to resolve issues about distributed collaboration and modelling with GenMyModel.

Keywords: UML, Collaboration

1 Introduction

Building complex software is a collaborative activity where modelling holds an important place. Since a decade, outsourcing and offshoring projects or part of them have become regular practices in software industry. These ones can have different reasons like the need to cut down costs or to explore new markets. Collaboration between project team members suffers from resulting geographical distribution.

Software practitioners usually draw diagrams when they design new applications, maintain existing ones or discuss with their clients using modelling tools [1]. Besides, models are important to manage knowledge and provide an efficient support to coordinate activities when the projects aim at producing complex software. It is important that modelling tools intent to support collaboration in the context of distributed teams, that is to say, to provide similar working conditions as co-located settings. Whitehead shows that the trend concerning collaboration support is the arrival of web-based tools in every phase of software development [2]. Their main advantage is that they do not require any installation or configuration: teams are quickly ready to work.

Axellience was launched from this observation and had the support of INRIA in April 2012. Now, Axellience offers GenMyModel which is an online UML and generator tool. Axellience has began to communicate about a beta-version of GenMyModel since January 2013 and there are already nearly a thousand of users spread over in more than 70 countries. Some models already reach more than 50 classes elements and includes several hundreds of model elements.

2 GenMyModel

The promises of SaaS (Software as a Service) is to allow users to access to services without installation or configuration. Users find the same environment

no matter where they connect from. In addition, SaaS applications can be used by users from anywhere, anytime, from any device and operating system, and users do not have to update their application. So, GenMyModel is online, in SaaS mode and its Graphical User Interface is shown in the figure 1. The intention of GenMyModel is to accelerate the modelling phase. Users are ready to work quickly and they access GenMyModel and the models from anywhere at anytime. For example, an architect can begin to model at his/her office, continue at home and present the result to the client the day after.

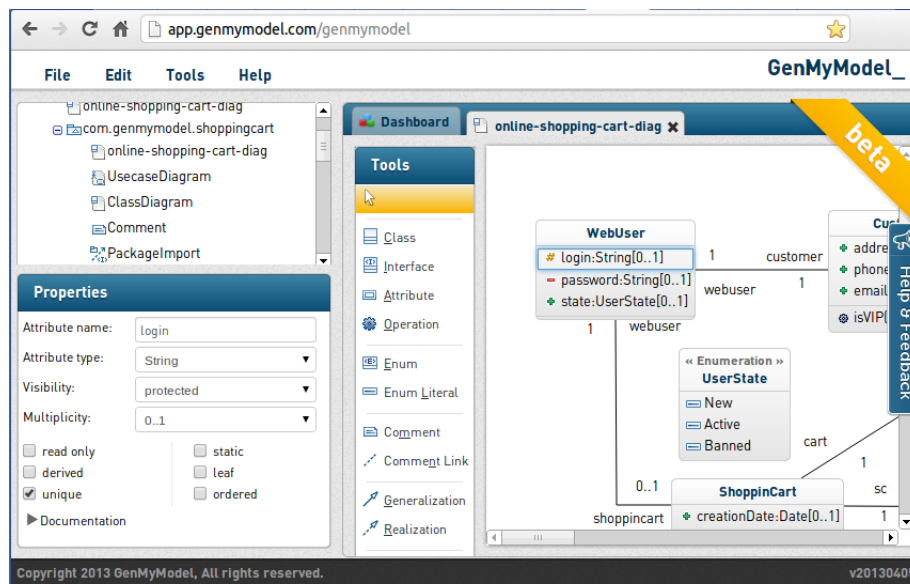


Fig. 1. GenMyModel Graphical User Interface

Today, GenMyModel supports Class Diagrams and Use Case. Others are in development. From Class Diagrams, GenMyModel offers its users the possibility to generate the code to multiple targets like Java, Java-JPA, SQL. Others will be added. In addition, the code is preserved throughout different generations. So, users can edit their code, model again, re-generate without code loss. Nevertheless, in most cases, a generator is not completely adapted to users needs. GenMyModel will add a generator upload service where users would add their own generators.

Another benefit of the SaaS applications is the simplification of real-time collaboration [3]. Real-time collaboration is an ongoing work into GenMyModel and a first version will be available soon. We are working on the model synchronization between users and the rights management. In the longer term, we have to study what kind of information is necessary to display to users in order to facilitate collaboration.

Future work will be on model repositories too. We plan to propose model repositories with versioning facilities. Thereby, users will be able to return to next version of their model.

One of the aims of GenMyModel is to offer a platform for specific needs or experimentations. For that, usage of GenMyModel will be free for public projects, allowing users to easily experiment and share their work. Additionally, the access to the model repositories will be granted to other tools through an API. This API will contain the generator upload service. By opening this API, Axellence wants to continue the collaboration with research laboratories where Axellence is already engaged.

3 Demonstration content

1. How to access to GenMyModel;
We will create a GenMyModel user in less than 30 seconds and will access to GenMyModel
2. Simple model creation;
We will present the GenMyModel Graphical User Interface and will create a class diagram.
3. Code generation;
From a simple model, we will generate associated code using the Java-JPA generator. This generation will be uploaded into a Github repository and we will show the generated code and annotations.
4. Model modification and code regeneration;
We will edit the generated code from a code editor tool in the Cloud. Then, we will edit our model to add new properties and we will generate the code again to show that new generations do not modify the code edited before.
5. Current state of collaboration.
To conclude, we will show our ongoing work on collaboration in editing the same model with two users.

References

1. Grossman, M., Aronson, J.E., McCarthy, R.V.: Does uml make the grade? insights from the software development community. *Inf. Softw. Technol.* **47**(6) (April 2005) 383–397
2. Whitehead, J.: Collaboration in software engineering: A roadmap. In: *Future of Software Engineering, 2007. FOSE'07, IEEE (2007)* 214–225
3. Brunelière, H., Cabot, J., Jouault, F.: Combining Model-Driven Engineering and Cloud Computing. In: *Modeling, Design, and Analysis for the Service Cloud - MDA4ServiceCloud'10: Workshop's 4th edition (co-located with the 6th European Conference on Modelling Foundations and Applications - ECMFA 2010), Paris, France (June 2010)*