

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

DiCoMEF: A Distributed Collaborative Model Editing Framework

Koshima, Amanuel; Englebert, Vincent

Published in:

Tool Demo at SEKE2014 (International Conference on Software Engineering and Knowledge Engineering)

Publication date:

2014

Document Version

Early version, also known as pre-print

[Link to publication](#)

Citation for pulished version (HARVARD):

Koshima, A & Englebert, V 2014, DiCoMEF: A Distributed Collaborative Model Editing Framework. in *Tool Demo at SEKE2014 (International Conference on Software Engineering and Knowledge Engineering)*. Hyatt Regency, Vancouver, Canada.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

DiCoMEF: A Distributed Collaborative Model Editing Framework

Amanuel Koshima
University of Namur
PReCISE Research Center, Belgium
amanuel.koshima@unamur.be

Vincent Englebert
University of Namur
PReCISE Research Center, Belgium
vincent.engelebert@unamur.be

Abstract

Domain specific modeling languages have matured overtime and are widely used, but facilities for collaborative modeling are still limited. Hence, there is a demand for tool supports that facilitates sharing of modeling artifacts (collaboration), conflict detection, reconciliation, and merging of concurrently edited models. This paper presents a collaborative model editing framework called DiCoMEF.

1 Introduction

Domain Specific Modeling Languages (DSML) have matured overtime and widely used as an efficient alternative to General Purpose Modeling languages (e.g., UML, Petri Nets) for modeling complex systems [2]. DSML describes concepts at different levels of abstraction using models, meta-models and meta-meta-models. A model is an abstraction of a software system and a meta-model is a language that describes concepts and constraints of models. A meta-meta-model is a minimum set of concepts which defines languages (e.g. EMF/Ecore¹).

Several metaCASE tools have been developed to support the use of models defined with DSML, but most of these tools do not support collaboration [3]. Nevertheless, whenever the complexity of a problem increases, the diversity of users in groups increases as well [4]. Hence, methods and tools to facilitate the cooperative work of these users are required. A model version control system, which facilitates sharing of modeling artifacts, conflict detection, merging, and versioning of models, is then required. More particularly, conflicts (textual, syntactic, or semantic [1]) that cause inconsistency should be identified and resolved.

Different approaches have been adopted to ensure collaboration, for instance, a central repository (e.g. SVN) with locking techniques and merging mechanism. However, the locking technique is not scalable [1] and this approach forces all users to be dependent on a central repository. This might introduce unnecessary access right bureaucracy that

would lead to dissatisfaction among members. There is another mode of collaborative work where each member has his/her local copy of a shared modeling artifact and works in isolation with other members. This approach provides users a better control of modeling artifacts and mitigates the problem of being dependent on a central server. However, it is challenging to keep all copies of concurrently edited models consistent. Most of available versioning tools detect conflicts in text or tree based documents, but they are not suitable for models that have a graph based nature [1].

2 DiCoMEF Framework

DiCoMEF is a distributed collaborative model editing framework based on EMF/Ecore meta-model and it lets each member of the group having his/her local copy of (meta)model (see Figure 1) [3]. In DiCoMEF, modifications are controlled by human agents (controllers). They manage the evolution of either models or meta-models depending of their role in the cooperative work. Model (resp. meta-model) controller roles are flexible meaning that they can be assigned (delegated) to other members of a group as long as there is one unique coordinator per group. Whenever members of a group modify (meta-)models locally, elementary change operations (*create*, *delete*, and *update*) are stored locally. DiCoMEF extends a history meta-model of EDAPT² to capture edit operations. These operations are used later as a means of communication among members of a group. A controller is used as a central hub of communication among a group, but members could still communicate with other colleagues directly. Peer-to-peer communication could hinder convergences of all copies of (meta)models.

DiCoMEF relies on two concepts (*main-line* and *branches*) to ensure the communication framework. The main-line stores different versions of a copy (meta-)model locally at each editors site. Editors cannot modify (meta-)models stored on the main-line; they can only adapt those stored on the branch and send then their local modifications

¹<http://www.eclipse.org/modeling/emf/>

²<http://www.eclipse.org/edapt/>

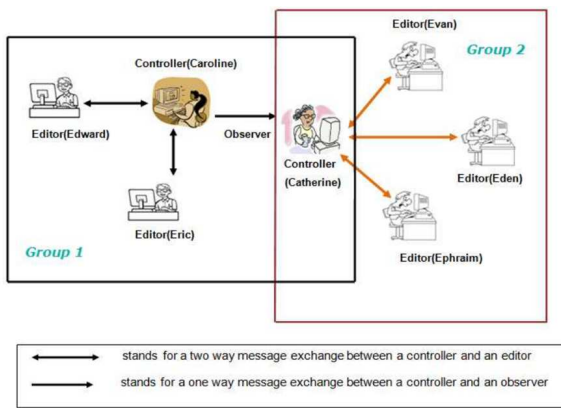


Figure 1. DiCoMEF architecture

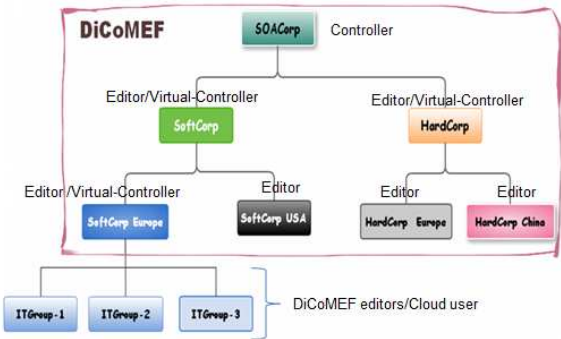


Figure 2. Extended architecture of DiCoMEF

to a controller as “change requests” so as to commit changes on the main-line. DiCoMEF detects structural conflicts (using edit operations) (see Figure 3) and semantic conflicts (with OCL constraints). Edit operations can be annotated with multimedia files to describe rationale of modifications. DiCoMEF framework could be extended to support a large community of users as shown in Fig. 2, where an editor acts as a *virtual controller* for other editors (*side editors*) working under her/his supervision. These new roles (Virtual controller and side editor) are transparent for the DiCoMEF controller. Side editors could also modify (meta-)models concurrently (e.g. by using the *Cloud*) but these modifications would be out of the scope of DiCoMEF.

3 Status and Evaluation

DiCoMEF framework is an eclipse plugin (54K LOC) based on EMF/Ecore meta-model and it provides a fully operational support for collaborative meta-modeling. Screen-casts of DiCoMEF tool, plugin, presentations, and publications are available online on DiCoMEF Web page³. Indeed, we have conducted a preliminary evaluation of DiCoMEF

³<https://sites.google.com/site/dicomef/home>

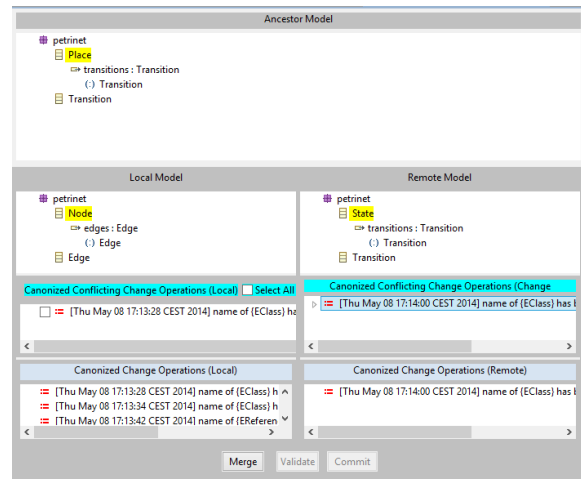


Figure 3. Merge tool

framework with masters students to assess: (1) the feasibility of collaborative methods and processes of DiCoMEF framework, (2) correctness of conflict detection (recall and precision) (3) usability of merge tool and DiCoMEF framework (4) measuring user efforts (time) needed to merge concurrently edit meta-models manually and using DiCoMEF merge tool. Overall result of the evaluation was positive and it showed DiCoMEF can be used for collaborative meta-modeling.

4 Future Work

The reconciliation process of DiCoMEF will be improved by letting users work with concrete syntax editors and meta-model semantics (i.e. OCL rules for instance). Besides, more advanced collaborative workflows will also be investigated and defined on top of DiCoMEF. Collaborative modeling part of DiCoMEF framework will be fully implemented. In addition, we plan to conduct more experiments and evaluations in the future.

References

- [1] K. Altmanninger, M. Seidl, and M. Wimmer. A Survey on Model Versioning Approaches. Technical report, Johannes Kepler University Linz, 2009.
- [2] S. Kelly. Case tool support for co-operative work in information system design. In *Info. Sys. in the WWW Environment*. Chapman & Hall, 1998.
- [3] A. Koshima and V. Englebort. Collaborative editing of emf/ecore metamodels and models: Conflict detection, reconciliation, and merging in dicomef. In *Proc. of the MODEL-SWARD 2014*, Lisbon, Portugal, 2014.
- [4] T. Zimmermann and C. Bird. Collaborative software development in ten years: Diversity, tools, and remix culture. In *Proc. of the CSCW Workshop on the Future of Collaborative Software Development*, 2012.