



HAL
open science

Heterogeneous design models alignment: from matching to consistency management

Mahmoud El Hamlaoui, Saloua Bennani, Mahmoud Nassar, Sophie Ebersold,
Bernard Coulette

► **To cite this version:**

Mahmoud El Hamlaoui, Saloua Bennani, Mahmoud Nassar, Sophie Ebersold, Bernard Coulette. Heterogeneous design models alignment: from matching to consistency management. 33rd ACM/SIGAPP Symposium On Applied Computing (SAC 2018), Apr 2018, Pau, France. pp.1695-1697. hal-02319706

HAL Id: hal-02319706

<https://hal.science/hal-02319706>

Submitted on 18 Oct 2019

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.



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:

<http://oatao.univ-toulouse.fr/22385>

Official URL

DOI : <https://doi.org/10.1145/3167132.3167425>

To cite this version: El Hamlaoui, Mahmoud and Bennani, Saloua and Nassar, Mahmoud and Ebersold, Sophie and Coulette, Bernard *Heterogeneous design models alignment: from matching to consistency management*. (2018) In: 33rd ACM/SIGAPP Symposium On Applied Computing (SAC 2018), 9 April 2018 - 13 April 2018 (Pau, France).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Heterogeneous design models alignment: from matching to consistency management

Mahmoud EL HAMPLAOUTI*, Saloua BENNANI,
Mahmoud NASSAR
IMS Team, ADMIR Laboratory, ENSIAS, Rabat IT Center,
Mohammed V University in Rabat
firstName.lastName@um5.net.ma

Sophie EBERSOLD, Bernard COULETTE
IRIT Laboratory, MACAO Team, University of
Toulouse-Jean Jaurès
firstName.lastName@irit.fr

ABSTRACT

Complex systems involve several business expertises that are designed as models in different modeling languages. These partial models are manipulated by different designers, and are thus generally heterogeneous (i.e conform to different metamodels). To create a complete view of the system, we proposed a process to organize partial models as a network of models through a virtual global model. As models evolve, changing elements involved in a correspondence, may cause the inconsistency of the global model. So, we have defined a process that automatically identify changes, classify them and treat their impacts on elements of other partial models in order to maintain the global model consistency.

KEYWORDS

process, metamodel, heterogeneous models, matching, consistency, correspondences, impacts

1 INTRODUCTION

Complex systems design involve a varied set of modeling experts from different business areas. This allows them to focus in isolation on different parts (partial models) of the system. However, at some point, it is mandatory to construct a global model to understand and effectively exploit the whole system. Partial models may evolve during the system life cycle. As their design was made by different designers, their evolution within a system may occur in an uncoordinated manner. Changing one or several elements, involved in a correspondence, may cause the inconsistency of the global model. Our current objective is to ensure the consistency of the global model by re-evaluating correspondences after the evolution of each partial model.

Our proposition takes part in the GEMOC initiative [4]. In this paper, we present an overview of the matching approach (detailed

in [2]) and present thereafter the consistency management which is the added value of this work. The consistency management process is automatically activated when the matching process produces the Model of Correspondences (M1C). It takes as input all the partial models and the M1C as presented in Figure 1. Our approach does not concern intra-model changes; it is up to partial models designers' to manage the internal repercussions of changes made on their models. The rest of this paper is organized as follows: Section 2 presents a general view of the matching approach whereas section 3 describes the consistency management approach. We conclude this paper by some perspectives and a conclusion in section 4.

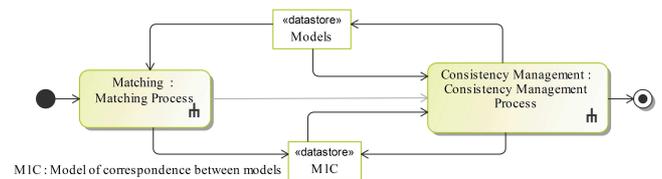


Figure 1: Overall process of our approach

2 MATCHING APPROACH

Our matching approach consists in analyzing input models (and their respective metamodels) in order to identify correspondences that exist among them. Correspondences are stored into a model of correspondences (M1C) conforming to a metamodel of correspondences (MMC). The M1C cannot be constructed in a monolithic manner. It follows a process that we call matching process. This process involves two stakeholders, namely, an integrator expert who is the supervisor of the application domain, and a tool that assists the supervisor in the automatic parts.

Firstly, the process takes as input the various metamodels and the kernel of the MMC. Subsequently, the supervisor verifies if the MMC contains all needed relationships to set up correspondences between partial (meta)models. If not, the DSR meta-class of MMC is specialized. The third activity of this process enriches the MMC with a Semantic Expression (SE) for each relationship. For this purpose, we proposed a Semantic Expression DSL [3] that is woven with the MMC as annotations. The advantage of using this DSL is primarily to have a structured common definition of each relationship. Secondly, it helps build M2C in an assisted way. Thirdly, it helps filter out the correspondences in the selection step by keeping only those that verify the semantics of their relationship.

Once the MMC is specialized, the matching operation begins, the supervisor identifies correspondences between meta-elements

in order to produce the model of correspondences called M2C. M2C (Model of correspondences between metamodells) stores High Level Correspondences that contain meta-elements linked by High Level Relationships. HLCs are then refined in order to produce LLCs. Our developed tool produces them semi-automatically by performing a reproduction operation on the M2C followed by an operation of selection.

3 CONSISTENCY MANAGEMENT APPROACH

Since models evolution is generally not coordinated between partial models designers, each model may evolve independently. So, it is very tedious to rerun the matching process after each change due to the required human effort and the lack of changes tracking.

Our approach provides a consistency management process. This process is automatically activated using the Observer pattern [6][5] at the end of the matching process. It takes as input the system's partial models and the M1C and follows six steps as shown in Figure 2. It is carried out by a developed tool and imply the supervisor's intervention in phases that require a human expertise or configuration. Throughout this section, we are going to detail these six steps.

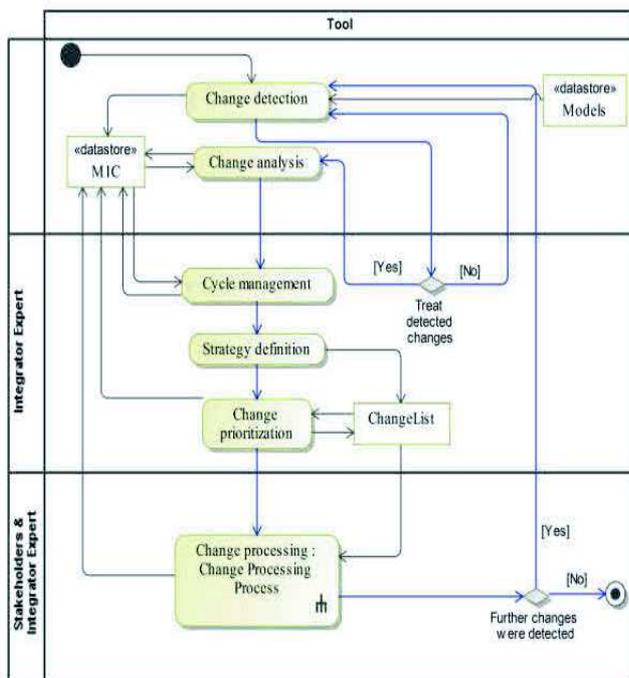


Figure 2: Consistency management sub-process

3.1 Changes detection

Changes are detected when they occur through the Observer pattern. They are added to M1C using the MMC meta-classes *History*, *DiffElt*, *AddedElt*, *DeletedElt*, *ModifiedElt* (part 1 of Figure 3). *History*

is used to keep track of applied changes. *DiffElt* allows to record the trace of evolved elements. It has two attributes. The first attribute contains the change classification type. The second one contains the reference of the element. *DeletedElt* refers to an element that no longer exists in the original model but that is maintained for tracing purpose. *AddedElt* and *ModifiedElt* respectively represent newly added element and changed element. The *Observer* meta-class (part 2 of Figure 3) specifies the model's element to be observed. It is a generalization of the subject meta-class which has three methods. Two of them (*attach* and *detach*) allow to fix or detach an observer object from a model element. The third method (*notify*) makes it possible to notify the M1C of occurring changes. The *update* method of the meta-class *Observer* is used during the phase of changes processing to maintain the consistency of domain's models. The third part of Figure 3 (i.e. the *impact* meta-class) defines the *impact kind* and the *solution* for each change.

3.2 Changes analysis

This analysis includes defining the type of change and the M1C elements that may be affected. The extension of the MMC allows to find, for each modified element, the correspondence(s) to which it belongs and thus to find the element(s) that may be affected directly or indirectly via a cascading effect.

We also classify changes in two categories: the automatic mode for added and deleted elements and the monitored one for modified elements. In this latter, when an element has changed, the correspondence must be assessed in terms of the semantics of its type of relationship. According to [1], when the relationship semantic comes into play, version management problems become more complex and can not be processed automatically. Thus, human intervention is necessary to decide about the change's impact.

3.3 Cycle management

Once changes and their direct or indirect impacts are detected, the tool catches automatically the cycles of cascading effects. The expert decides which correspondences should be removed in order to break the cycle. Let's consider three correspondences. The first one relates an element A to an element B, the second one connects B to C and the third one relates C to A. If A is modified, B can be directly influenced by this change, which indirectly influences C. In the case where C is the one that causes the modification of A, we will have a cascading cyclical effect.

3.4 Change scheduling strategy

This step aims at producing an ordered list of changes. We propose two strategies for changes ordering: classification-based strategy and impact-based strategy.

The classification-based strategy consists of creating a list of changes that contains changes that are classified in automatic mode followed by those in monitored mode. The second strategy creates an ordered list depending on the type of impact of each change. For example, the expert may start by processing the changes that have both direct and indirect impacts on other elements and leave changes that have only direct impacts to the end. These two scheduling strategies work for changes that have different modes of change or different types of impact. Next, we will see how to deal with

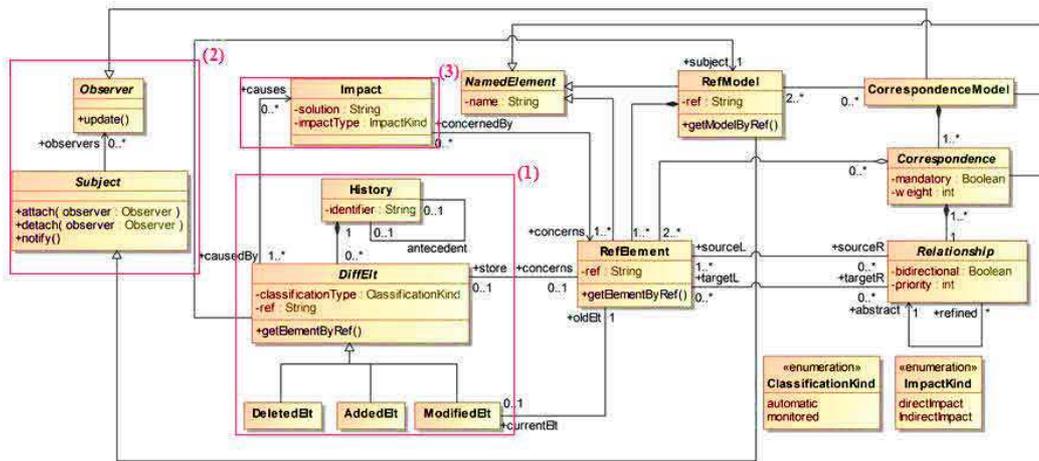


Figure 3: Extension of the Correspondence Metamodel MMC to handle consistency management

changes that have the same type of impact or the same mode of classification.

3.5 Change prioritization

Changes processing order has an impact on the system and its consistency. That's why we attribute a weighting coefficient to each correspondence. This coefficient is calculated according to the following formula:

$$\begin{aligned}
 weight &= \sum_{k=0}^n (DirectlyAffectedElement_k * priority) \\
 &+ \sum_{k=0}^n (IndirectlyAffectedElement_k * priority)
 \end{aligned}$$

3.6 Change processing

M1C and the partial models may be modified to take account of detected changes. Changes categorized in automatic mode are processed automatically. The matching process is restarted at the end of the change process to handle all added elements at once. When an element is deleted, all correspondences involving it become orphaned. Hence, the expert checks if it is mandatory for the system (mandatory=true). If so, the deleted item is restored, otherwise the correspondence is removed from the M1C. Concerning changes occurring in a monitored mode, they are managed semi-automatically. The correspondence is maintained if it remains correct regarding the semantic associated to its type of relationship after the change of one of its ends. Otherwise, it is necessary to modify each of the elements tied to a modified element if this modification is possible. If not, the correspondence is deleted if it is not mandatory (mandatory=false). Otherwise, a group decision making takes place to decide which end of the correspondence has to be modified.

4 CONCLUSION AND PERSPECTIVES

Our general research work addresses view-based complex systems design. During the modeling cycle, the description of models evolves frequently due to the emergence of new requirements and constraints. In a multi-modeling environment, several changes can

occur on different models of the system. To manage the consistency between these models, we propose to exploit the correspondences model to treat the changes that are identified automatically on partial models in order to maintain the consistency of the interconnected models. Once the changes are identified, the consistency management process proceeds to their classification and the potential impacts are identified automatically as well as the possible presence of cycles. These latter are managed by the expert. Change prioritization is important because without coordination the evolutions treatment could become unmanageable. For this, according to the chosen strategy, a list of changes is generated according to the calculation of weighting coefficients. Finally changes proceed automatically based on a change processing sub-process.

As a POC of our approach we are developing a support tool called HMCS (Heterogeneous Matching and Consistency management Suite). It provides assistance to the expert in the creation of the model of correspondences and the management of the consistency between heterogeneous partial models when they evolve. HMCS is operational but only supports the matching sub-process. This tool, once completed, will allow us to validate our approach and conduct experiments to verify its scalability.

REFERENCES

- [1] A. Cicchetti and F. Ciccozzi. 2013. Towards a Novel Model Versioning Approach based on the Separation between Linguistic and Ontological Aspects. In *ME 2013-Models and Evolution Workshop Proceedings*. CEUR-WS, 60.
- [2] M. El Hamlaoui, B. Coulette, S. Ebersold, S. Bennani, M. Nassar, A.I Anwar, A. Beugnard, J.C. Bach, Y. Jamoussi, and H.N. Tran. [n. d.]. Alignment of viewpoint heterogeneous design models: Emergency Department Case Study. In *International Workshop On the Globalization of Modeling Languages (GEMOC) co-located with MODELS 2016*. CEUR Workshop Proceedings.
- [3] M. El Hamlaoui, S. Ebersold, B. Coulette, A. Anwar, and M. Nassar. 2015. Maintien de la cohérence de modèles de conception hétérogènes. *Technique et Science Informatiques* 34, 6 (2015), 667–702.
- [4] GEMOC. 2017. Initiative On the Globalization of Modeling Languages. (April 2017). <http://gemoc.org/ins/>
- [5] C. Larman. 2002. *Applying UML and patterns: an introduction to object-oriented analysis and design and the unified process*. Prentice Hall.
- [6] J. Vlissides, R. Helm, R. Johnson, and E. Gamma. 1995. *Design patterns: Elements of reusable object-oriented software*. Reading: Addison-Wesley (1995).