# A Group Decision-Making Approach for Global Consistency of Heterogeneous Models

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

International Journal of Information Technology & Decision Making


# A GROUP DECISION-MAKING APPROACH FOR GLOBAL CONSISTENCY OF HETEROGENEOUS MODELS

SALOUA BENNANI
*SM@RT team, IRIT Laboratory, Toulouse, France*

SOPHIE EBERSOLD
*SM@RT team, IRIT Laboratory, Toulouse, France*

MAHMOUD EL HAMLAOUI
*IMS Team, Mohammed V University in Rabat, Morocco*

BERNARD COULETTE
*SM@RT team, IRIT Laboratory, Toulouse, France*

MAHMOUD NASSAR
*IMS Team, Mohammed V University in Rabat, Morocco*

Received Day Month Year
Revised Day Month Year

The design of complex systems goes through a multi-view paradigm in which separate teams, from different viewpoints, build partial source models describing the system. These source models are called heterogeneous models since they are expressed in different languages. The main objective of this paper is to provide an approach - called CAHM for Collaborative Alignment of Heterogeneous Models - that leverages collaborative engineering and especially group decision-making principles to ensure the overall consistency of heterogeneous source models. This approach defines two sub-processes : a first one to collaboratively match heterogeneous models to develop the inter-model correspondences and a second one ensuring the consistency of the produced model of correspondences in case of model evolution. In this paper, we restate the basis of the CAHM approach, then, we detail the second sub-process that aims at maintaining the coherence of the overall system. This sub-process handles the evolution of source models by managing the impact of these evolutions on the established model of correspondences. It incorporates mechanisms to calculate the impact of changes, as well as mechanisms to formalize the group decision-making, while addressing the inconsistencies that may occur due to changes. CAHM is illustrated and validated on a real example of a hospital emergency department case study.

*Keywords*: Multi-views, Consistency Management, Collaboration, Group Decision-Making, Evolution.

## 1. Introduction

Design of software systems involves actors with various skills and fields of knowledge, especially when these systems are complex. This principle of the separation of concerns [20] reveals its efficiency by giving each kind of actor a dedicated environment, tool or modeling language to represent their own partial views of the system [7]. In this context of design, each (kind of) actor separately produces an independent partial model[a] of the system, which corresponds to their partial view. While the produced design models - also called source (or partial) models - are independent, they give complementary descriptions of the whole system [26, 8].

In the context of Model Based System Engineering (MBSE), the system's viewpoints are depicted by a set of models and their respective metamodels. These models are considered heterogeneous as they conform to different metamodels. Considering a car as a complex system for example, different source models such as mechanical, electronic and ergonomic models are used to represent the system. Having various models ensures that different stakeholders viewpoints are reflected, yet working with models separately may be harmful for the system's integrity and global consistency [21]. This is particularly true when a model evolves due to a new requirement or a technical constraint, while the other source models of the system do not take this evolution into account. Taking a car system as example, models of this system may evolve for different reasons, for instance, a technological evolution of an embedded component like the anti-lock braking system can be supported (a part of a model may thus become invalid), an ergonomic or environmental aspect that was not considered in the first stages may be taken into account (a viewpoint may be added to the system), a decision to forsake the appearance aspect can be made (a viewpoint may be discarded), etc. So it is crucial to obtain a global perspective on the system. This global perspective can be used for several purposes: traceability [28], synchronization [22] or inter-model consistency management [53, 35]. We are mainly interested by the inter-model consistency issue, i.e., establishing the consistency of the overall system during its design, then maintaining this consistency in case of changes (changes in business rules or constraints for example).

Various techniques based on MBSE have been proposed to elaborate a global view of a system: fusion; weaving and federation [5, 18, 16, 31]. These techniques reduce the effort required since they are based on meta-modeling principles. We focus our study on federation and weaving techniques because they promote both having multiple modeling language and defining links (correspondences) between the models' elements. Subsequently, we denote these techniques by the term of alignment, borrowed from the field of schemes and ontologies [51]. Model alignment consists of (i) establishing links between models' elements (correspondences) and (ii) re-evaluating these correspondences in case of model evolution.

[a]A model is an abstraction of a physical system. It specifies the physical system from a certain vantage point (or viewpoint); that is, for a certain category of stakeholders, and at a certain level of abstraction, both given by the purpose of the model [46].

There are several approaches for heterogeneous model alignment. Here, we briefly summarize our main observations about these approaches: either the studied approaches propose a set of frozen relationships to relate models [6, 49] or suppose that a single actor (i.e. a systems expert) can perform alone the alignment [30, 23, 25, 10]. If the single actor assumption holds for small systems with a limited number of viewpoints, it is no longer valid in case of complex systems. In fact, no matter how expert in the application domain the actor performing the alignment is, he cannot grasp the technical and functional concerns of all involved viewpoints, especially in case of strongly heterogeneous models. Besides, studies have shown that integrating collaboration into MBSE improves the effective management of complex systems [27]. Involving all concerned actors enables the capture of wider knowledge and preoccupations, and facilitates the alignment processing by creating bridges among their viewpoint models [52, 14].

Our objective is thus to provide collaborative mechanisms that enable a participatory management of source model alignment. We have proposed the kernel of a collaborative matching process to establish inter-model correspondences [4]. This process is accomplished thanks to a metamodel called MMCollab (for MetaModel of Collaboration). MMCollab supports the description of collaborative sessions of collective decision elaboration [2], we have described how the group decision-making process is formalized through MMCollab in [3]. In this paper, we focus on the global system consistency when source models evolve: we provide a collaborative process to maintain the consistency by managing changes that may occur and their respective impacts.

The rest of this paper is structured as follows. First, we present in section 2 a hospital emergency department (ED) system that serves as a case study for the rest of the paper. We illustrate the targeted problems and motivations using this example. Section 3 describes our proposal for group decision-making processes formalization. In section 4, we start by giving an overview of our approach called CAHM (Collaborative Alignment of Heterogeneous Models), then we detail the sub-process of global consistency management. Section 5 presents the implemented support tool, Heterogeneous model Matching and Consistency management Suite in its Collaborative version (HMCS-Collab), and provides an experimental validation of CAHM on the ED system, using this tool. Section 6 discusses the experimental validation and the threats to validity. Section 7 is dedicated to the related work. In the last section, conclusions are drawn and we sketch out possible future work.

## 2. Problem's motivation illustrated by a case study

### 2.1. *Overview of the emergency department case study*

We use a case study of an emergency department of a hospital to enact the approach. An emergency department (ED) is an essential branch of the health system in all countries. This complex service requires specific skills provided by a multidisciplinary approach where viewpoints are complementary. The case study was

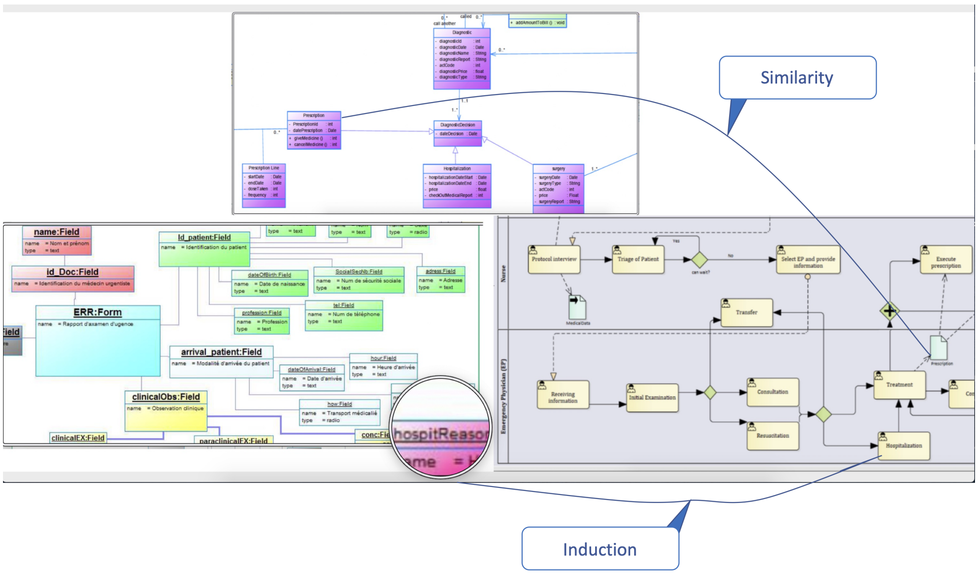



Fig. 1. Extract of ED's source models.

developed in cooperation with the medical staff of a French hospital. We identified a representative set of viewpoints which are necessary for the proper functioning of the ED. Thus, we consider the three following viewpoints:

- Software Design (SD): This is an object-oriented representation of the system. Its model describes the system as classes having attributes and operations;
- Business Process (BP): This viewpoint describes the system as a workflow of activities and flows among roles;
- Examination Report (ER): It represents the digital mockups of an emergency report as a set of fields.

Models associated with these viewpoints have been elaborated in collaboration with several teams across the globe. Each team has handled a given design viewpoint [25]. Figure 1 presents small extracts of the SD, BP and ER models. The SD model contains Classes concerning patients, their medical history and diagnostics. Roles and their respective Activities are described in the BP model. In the ER model, Fields that form the medical report are described. (e.g., socialSecurityNumber, clinicalObservations). Next, we consider as a local coordinator, the actor responsible for each model. Thus, we refer to Software Design (SD), Business Process (BP) and Emergency Report (ER) local coordinators respectively as $SD_{LC}$, $BP_{LC}$ and $ER_{LC}$.

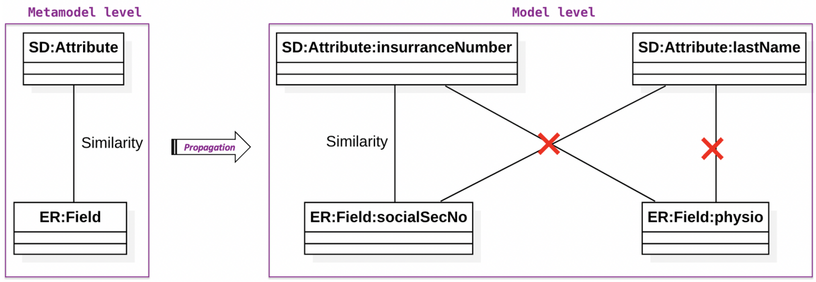



Fig. 2. Examples of correspondences at metamodel and model level using the similarity relationship.

## 2.2. *Motivation for model matching*

Model matching consists of identifying correspondences among the source model elements. Figure 1 illustrates that some concepts of a model have the same name or dependent name in another model. For instance, we have a concept Prescription in both BP and SD models. Likewise, we find a concept hospitalization in the BP model and a concept called hospitReason (abbreviation for hospitalization reason) in the ER model. These redundant or dependent names show that links need to be traced among these elements to have a consistent global view of the ED system.

To reduce effort and optimize the matching process, we choose to establish correspondences manually at the metamodel level (called meta-correspondences) between the concepts of the metamodels (called meta-elements) and propagate them at the model level. Obviously, not all the instances of the linked meta-elements have to be connected. To ensure this filtering we rely on relationship semantics. This ensures that the connected elements satisfy the ad-hoc semantics. This is achieved thanks to model matching concepts [25, 4] (briefly recalled in section 4.2). Figure 2 shows an example of a meta-correspondence and four of its associated correspondences. Considering a *Similarity* between the meta-element Field from the ER metamodel and the meta-element Attribute from the SD metamodel, a *Similarity* will relate the instances of Field and the instances of Attribute only if the semantics is satisfied; thus, a *Similarity* relates ER:Field:socialSecNo and SD:Attribute:insurranceNumber for example. On the contrary there is no *Similarity* between ER:Field:physio and SD:Attribute:lastName. We aim to define as automatic a matching process as possible incorporating the semantics of relationships like the similarity relationship in a dedicated tool.

## 2.3. *Motivation for handling model evolution*

As the source models are complementary to represent the system, constraints or changes in one model need to be reflected on the other models to ensure a consistent global perspective of the system. Let us consider the following change in the ED

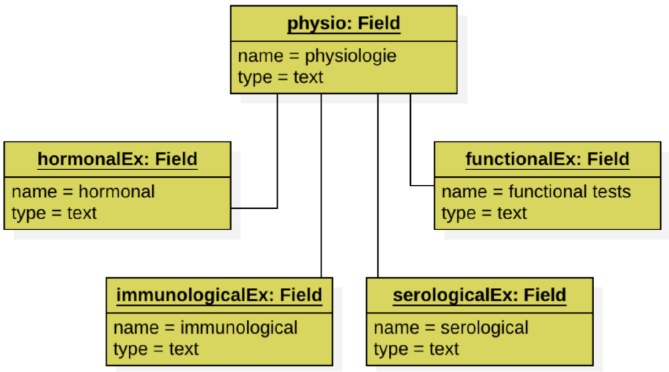



Fig. 3. Specialization of Field:physio during the evolution of the ER model.

system, Field physio (physiologicalExamination) of the ER model has undergone a specialization by adding the hormonalEx, immunologicalEx, serologicalEx and functionalEx subfields as summarized in Figure 3. These four Fields correspond to specializations of physiological examinations.

Assuming that the meta-element ER:Field is related to SD:Attribute by a *Similarity* as in Figure 2, the evolution of the Field physio may have repercussions on the SD model. In this case, correspondences should be established between all subfields of ER:Field:physio and the SD:Attribute to which physio is linked as will be detailed in section 4.

## 3. Formalization of group decision-making

The formalization that we propose to handle collaboration in the design phases of a system relies on a metamodel which we call MMCollab. MMCollab provides a set of meta-elements to describe group decision-making processes. *Collaboration* is the focal point of MMCollab, described in Figure 4. It is a specialization of SPEM's *Activity* [44] and enacted via a *GDMPattern* (*adoptedGDMPattern*).

### 3.1. *GDMPattern: Group decision-making pattern*

A *GDMPattern* is characterised by a *ParticipationMethod* and a *CodecisionMethod*.

- *ParticipationMethod* specifies how stakeholders participate in the decision-making. It is specified via the enumeration *ParticipationType*. It is *democratic* when all stakeholders are involved, and *restricted* when only the subset of stakeholders fulfilling the selection criteria are involved. For each *ParticipationMethod*, some parameters could be specified (i.e., *ParameterKind*: stakeholders anonymity and expertise level). In case of a *restricted* participation, the criterion for selecting stakeholders should be specified (either availability or expertise domain).

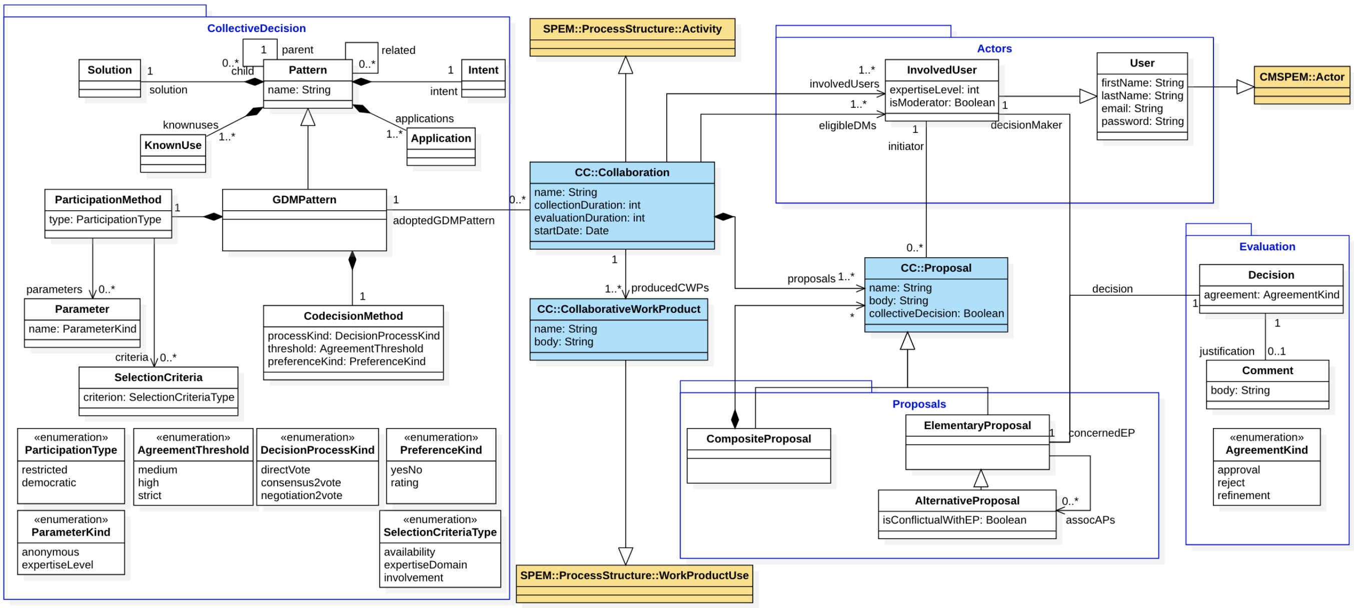



Fig. 4. Overview of MMCollab.

- *CodecisionMethod* is determined by three attributes:
  - *threshold* specifies the expected adherence of the group to the proposal. A strict threshold means that a 100% agreement is required whereas medium and high thresholds require less adhesion.
  - *preferenceKind* specifies how proposals are evaluated: by rating or by Yes/No.
  - *processKind* specifies the way proposals are evaluated. Since stakeholders may be in different locations, we propose three decision processes stored in the enumeration *DecisionProcessKind*. They allow opinions to captured through a final vote, and depend on the existence of a discussion preceding the vote: *directVote* (each local coordinator votes to express his/her choice), *consensus2vote* (the vote is preceded by a discussion leading to a consensus; it requires a strict threshold) and *negotiation2vote* (in case of medium or high threshold, a negotiation phase is held before the vote).

### 3.2. *Collaboration*

A *Collaboration* is a set of *Proposals* which have to be evaluated according to the *adoptedGDMPattern*. A *finalDecision* is set for each *Proposal* at the end of the collaborative session. A *Collaboration* implies a set of *involvedUsers*, including a moderator (*isModerator* attribute of *InvolvedUser* set to True). The moderator chooses the *GDMPattern* to be adopted. By default, the stakeholder who made the first proposal is considered as the moderator. A list of eligible decisionmakers (*eligibleDMs*) is initialized by the *involvedusers* who satisfy the *adoptedGDMPattern*.

A *Proposal* may be composite or elementary. *CompositeProposal* is a kind of atomic transaction, composed of a tree of *ElementaryProposal* (Elementary_P) that are either approved or rejected together. Each *Elementary_P* comes from a user (*initiator*) and has to be evaluated by the *eligibleDMs*.

A *decisionMaker* is an *InvolvedUser* who can evaluate a *Proposal*. The evaluation consists in producing an individual decision (*Decision*). The decision can be an *approval*, a *reject* or a *refinement* (enumeration: *AgreementKind*). When a *decisionMaker* rejects an Elementary_P he has to justify his choice by a *Comment*. If he thinks an Elementary_P needs to be refined, he provides an *AlternativeProposal* (Alternative_P). The attribute isConflictualWith Elementary_P of an Alternative_P specifies that this Alternative_P is conflicting with the Elementary_P to which it is attached. The value of *finalDecision* attribute of a *Proposal* is set by aggregating the individual decisions according to the *adoptedGDMPattern*.

### 3.3. *Instanciation of GDMPattern: DecisionPolicy*

A DecisionPolicy (DP) is an instance of a GDMPattern. More precisely, a DP is a combination of instances of elements that make up a *GDMPattern*: a *ParticipationMethod* and a *CodecisionMethod*, and therefore, a combination of instances of elements that characterize them (the type of participation (*type*), the type of decision-making process (*processKind*), the agreement threshold (*threshold*), and the type of preference (*preferenceKind*)). The combination of these elements allowed us to identify five decision policies that describe the policies commonly used in group decision-making. These policies (classes highlighted in Figure 5) take into account the various human styles of behavior: e.g., dominating; integrating; compromising; obliging and avoiding [12].

*MajorityDeciding* is a democratic decision-making policy. It also inherits the SingleElectionDP (SingleElectionDP) because it is carried out in a single round. This means that if the decision-makers have not reached the threshold defined at the end of the collaboration, they must adjust the threshold or re-evaluate the proposals.

*ConsentingTogether* and *NegotiatingTogether* are iterative DPs, which means they can be repeated until the set threshold is reached. *ConsentingTogether* requires a strict threshold (100% agreement) while *NegotiatingTogether* works with a low, medium or high threshold. *Delegating* and *TakingAdvice* are restricted DPs that require to specify the selection criteria for decision-makers.

These decision policies are not fixed and can be extended according to the contexts of application, by exploring the possible combinations of the elements that characterize them. For example, the *processKind* and *threshold* of the decision policy *Delegating* are not constant. They can therefore take a range of all possible values and provide a decision policy similar to *MajorityDeciding*, *ConsentingTogether*, or

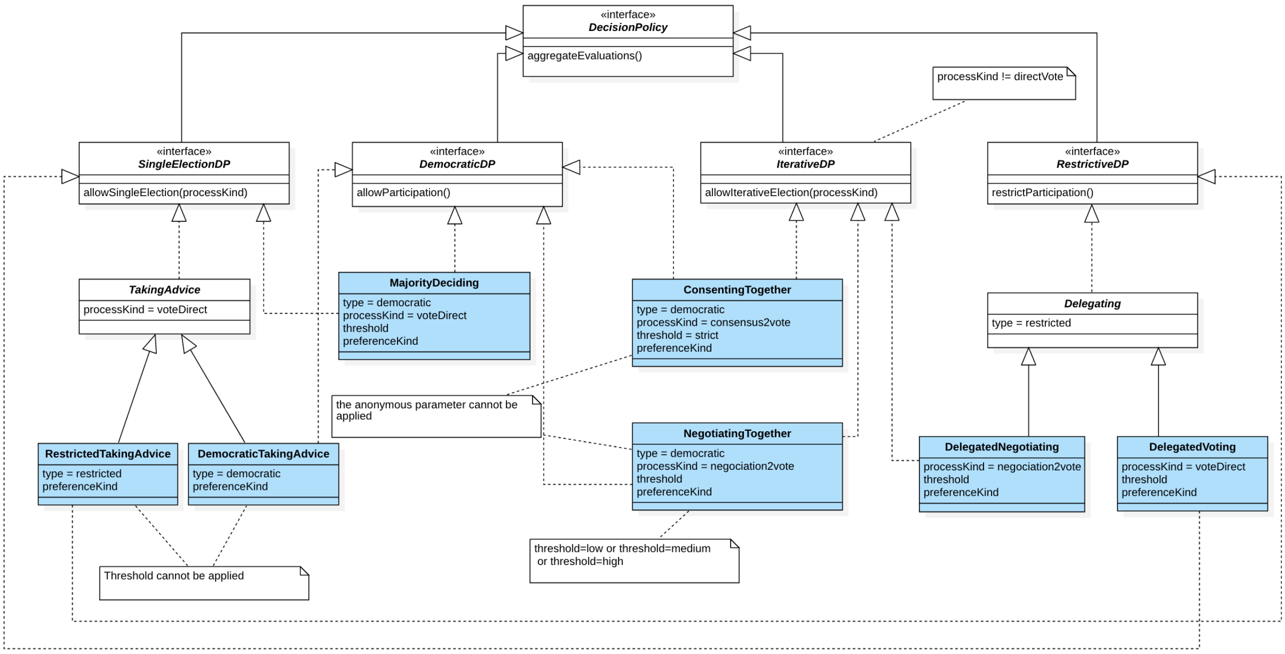



Fig. 5. Instances of GDMPattern and their Dependencies.

*NegotiatingTogether*, but in a restrictive mode.
To facilitate the choice among these decision policies, we provide a descriptive manual that summarizes the proposed patterns, according to the model of Gamma [29]. Appendix A shows an example that descibes the *ConsentingTogether* decision policy.

## 4. Consistency management in case of model evolution

### 4.1. *Overview of CAHM (Collaborative Alignment of Heterogeneous Models)*

CAHM is an approach which aims at developing a global view of a system, based on its source models. This approach offers an overall collaborative process on two consecutive phases:

- CAHM-Phase1: A collaborative matching sub-process [2, 4] whose objective is to establish correspondences among elements of the source models. This process is out of scope of this paper.
- CAHM-Phase2: A collaborative sub-process for maintaining consistency in case of model evolution (i.e., changes occuring in source models or meta-models). It aims to maintain the links established by the collaborative matching sub-process. This process is detailed in section 4.2.

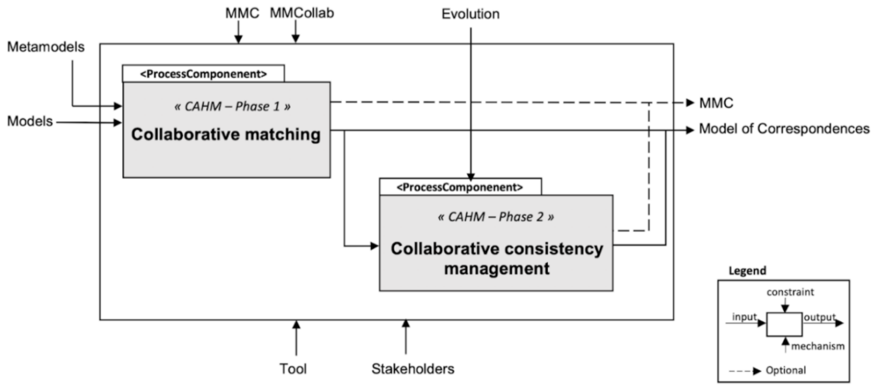



Fig. 6. The overall CAHM model alignment process.

Figure 6 illustrates the overall process of the approach as an SADT[b] diagram [40]. This diagram shows the input and output artifacts as well as the mechanisms used. The two sub-processes of CAHM take as input both source models and metamodels. The first sub-process produces a model of correspondences (M1C) that gathers the inter-model correspondences while the second sub-process updates the produced M1C.

CAHM sub-processes exploit both the knowledge of the stakeholders and a support tool HMCS-Collab (Heterogeneous Matching and Consistency-management Suite - Collaborative). The knowledge of the actors and their choices are used in the two phases to carry out manual definitions and decision-making activities. The tool performs the automatic tasks, namely generation of the model of correspondences, detection of changes and a part of the inconsistencies processing, as we will see in the following (in section 5.2).

The approach is based on two metamodels: MMCollab [3] and MMC [23].

MMCollab (MetaModel of Collaboration) - described in section 3 - is used to carry out decision-making whether at the level of the collaborative matching phase, or that of consistency management. In the first phase, MMCollab is instantiated to enact group decision-making about correspondences. In the second phase, MMCollab is instantiated to support the collective decisions regarding the consistency of the model of correspondences in case the source models or metamodels have undergone some changes.

The structure of MMC (MetaModel of Correspondences) enables the definition of the inter-model correspondences (related elements and the relationships that link them). It also sets the list of supported changes (cf. section 4.2 - Figure 8).

[b]Structured Analysis and Design Technics.

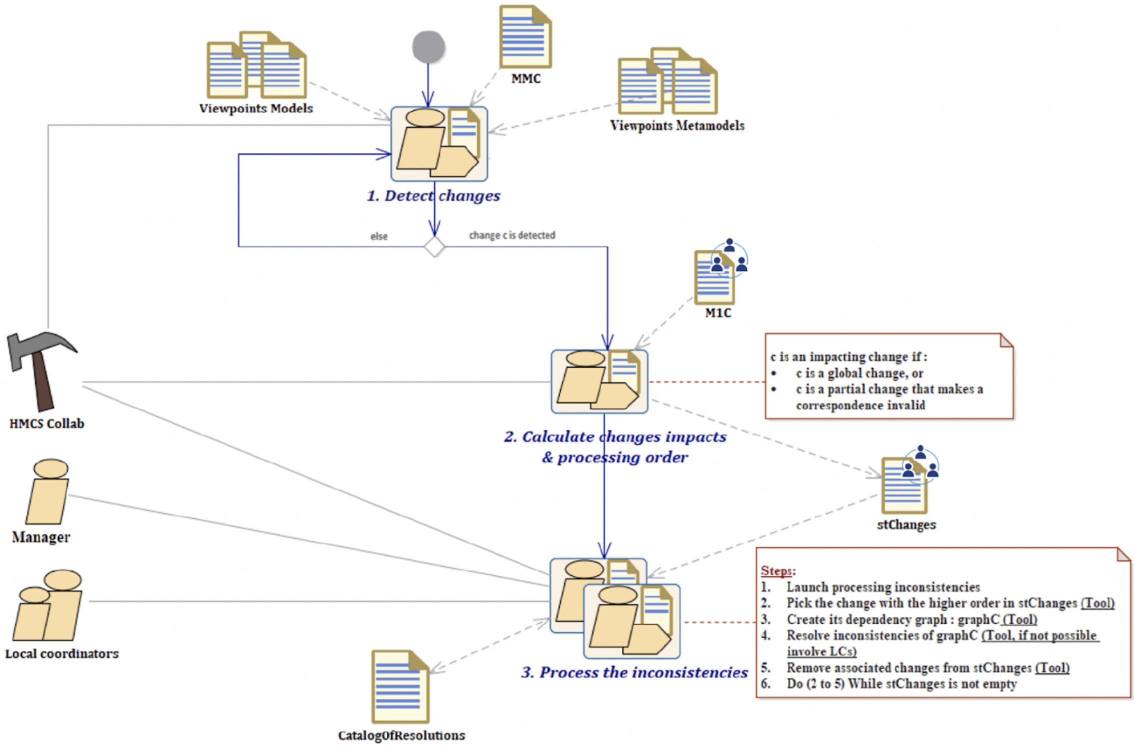



Fig. 7. Overview of CAHM consistency management collaborative sub-process.

### 4.2. *CAHM-Phase2: collaborative consistency management*

Figure 7 illustrates the consistency management sub-process. First, a tool - named HMCS-Collab (detailed in section 5.2) - observes models and their metamodels to capture changes (activity 1).

Detected changes are stored in a stack of changes (stChanges) according to their impact (activity 2). Afterwards, in activity 3, the inconsistencies caused by these changes are processed. This third activity is a collaborative activity composed of six sub-activities (cf. Figure 9). It may require the involvement of local coordinators if there is no unique/automatic resolution suitable to deal with an inconsistency.

#### 4.2.1. *Activity 1: Detect changes*

In Model Driven Engineering, many types of evolution can occur. The changes can concern metamodels or models. We propose distinguishing global changes from partial changes.

A *global change* occurs when a model or a metamodel is created or deleted, while a *partial change* occurs when an element of a model is added, deleted or modified.

We have extended the Metamodel of Correspondences (MMC) to incorporate these types of changes as summarized in the left part of Figure 8.

A *Repository* keeps the history of changes made to source models and metamodels. It consists of a set of versions. A *ChangesVersion* is composed of a set of *Changes* and has a state. A pending state means that the impact of modifications made to one model or metamodel has not yet been analyzed or considered on the other (meta-)models; a validated state means that the impacts of modifications made to one (meta-)model on the other (meta-)models have been analyzed and processed. A *GlobalChange* concerns a (meta-)model (*concernedModel*). Global changes (i.e., creation or deletion of a (meta-)model) are listed in the *ChangeGlobalKind* enumeration. A *PartialChange* concerns a model element (*concernedElt*). We list in the *ChangePartialKind* enumeration the following partial changes inspired from the work in the literature [32]:

- Create: addition of a property or a class to a model,
- Delete: deletion of a property or a class from a model,
- Move: move of a property from one class to another. It is a pull up/push down if the property is moved to a parent class/descendant class,
- Rename: renaming of a model element,
- Extract class: particular kind of move: creation of a new class and moving methods or attributes coming from another class to this class,
- Inline class: deletion of a class after having moved its properties to another class,
- Flatten class: removal of a set of properties p1, ..., pn from a parent class A, and their addition to subclasses B1, ... Bn. Then, deletion of A,
- Change multiplicity: change of the multiplicity of an association,
- Change type: change of the type of a property.

#### 4.2.2. *Activity 2: Calculate change impacts and processing order*

Once a change is detected, HMCS-Collab checks if it is impacting. A change is considered as *impacting* if it is: (i) a global change; (ii) a partial change of type addition or (iii) a partial change of an element present in the model of correspondences, making at least one correspondence invalid.
For non-impacting changes, HMCS-Collab validates the status of ChangesVersion (i.e. state = validated) and keeps listening for new changes. For impacting changes, HMCS-Collab calculates the change processing order. This order depends on the type of change as explained in Appendix B:

- A global change has a high priority; so, it is inserted at the top of the *stChanges* stack since it changes the whole structure of a viewpoint, and will be processed first.
- A partial change:

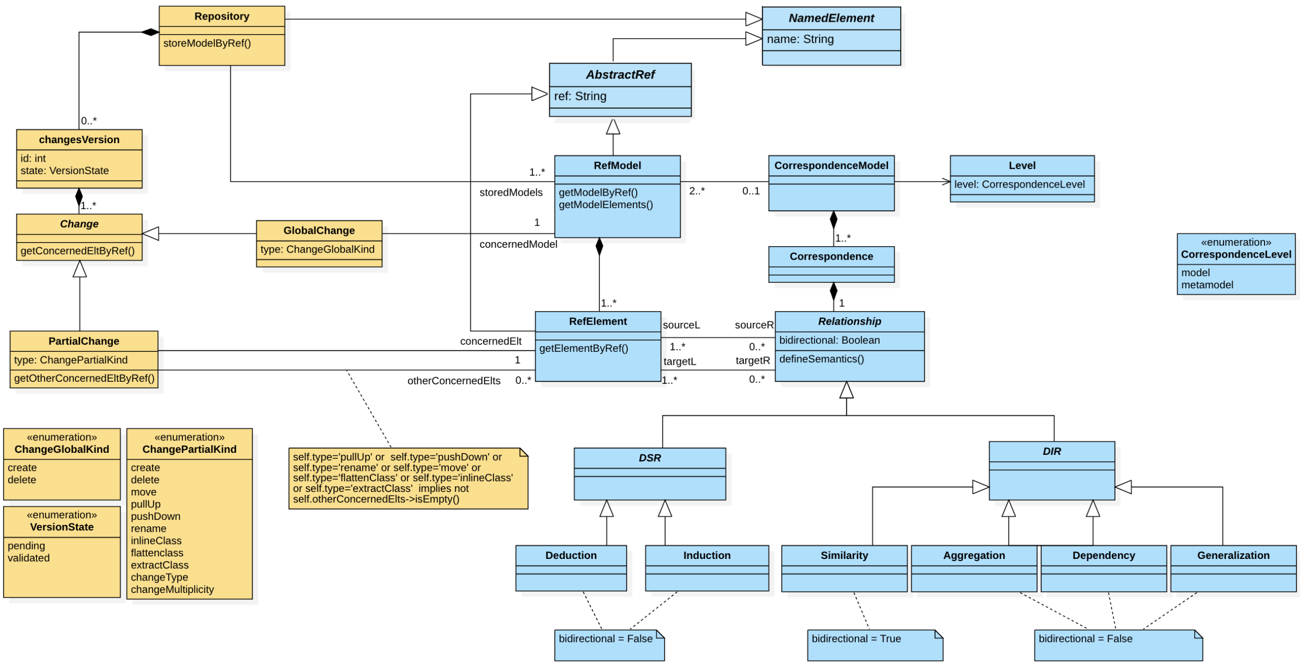



Fig. 8. MetaModel of Correspondences - MMC.

- of type modification (Move, Pull up, Push down, Inline class, Extract class, Flatten class) or deletion (Delete) is processed according to its order, which corresponds to the number of correspondences it makes invalid.
- of type addition (Create) is processed last as it has no effect on the already established model of correspondences.

### 4.2.3. *Activity 3: Process the inconsistencies*

Source models are locked during this activity to ensure that local coordinators can not make any further changes. Figure 9 illustrates the breakdown of this activity into six sub-activities. It takes as input the stack of changes stChanges and a catalog that contains applicable resolutions for each type of change.

Inconsistencies processing starts automatically in case of a global change. In case of a partial change, it is up to the manager to launch the inconsistency processing manually (activity 1) following a request from one of the local coordinators.

Once the processing is started, HMCS-Collab picks up (via the stChanges stack) the change having the greatest impact. Then, it builds a dependency graph of this change (activity 2). The purpose of this graph is to plot the elements directly or indirectly related to the changed element (it can be also a model or a metamodel).

Once all the graphs of all of the changes have been created, the processing of changes starts by dealing with the most impacting change until the stChanges stack is empty.

HMCS-Collab verifies the (meta-)correspondences that are no longer valid in the graph. Thus, the local coordinators are aware of the inconsistencies to be resolved. They can therefore rely on this graph to determine resolutions allowing them to obtain a correct graph. For this purpose, they exploit the *catalog of resolutions*. This catalog contains an expandable list of resolutions classified according to the types of changes. An extract of this catalog is given in Appendix C. HMCS-Collab searches appropriate resolutions in the catalog of resolutions (activity 3) and proposes them to support decision-making. In addition, some defined resolutions are fully automatic. When an automatic resolution is the only one considered for a change, it is applied without any human intervention by the HMCS-Collab tool (activity 4). Supervised resolutions are resolutions that involve local coordinators, either because:

(1) Several resolutions from the catalog of resolutions are appropriate for this change (activity 5). For example, in case of adding a metamodel, Adjust meta-correspondances or Define meta-correspondences are applicable (cf. Appendix C).
(2) The change does not have a suitable resolution (activity 6).

## 5. Proof of concept on an Emergency Department System

This section presents an experimental validation of the consistency management process in case of models evolution. We start by illustrating the model of correspondences established during the collaborative matching phase. Then we describe the consistency management of the ED system during model evolution (section 5.1). We also give an overview of HMCS-Collab's modules (section 5.2).

### 5.1. *Inconsistencies management of the ED system*

Figure 10 shows an excerpt from the model of correspondences of the emergency service in a tabular view (correspondences using an induction relationship). Column *Id* indicates the number of the correspondence, column *Relationship* indicates the name of the relationship used in the correspondence while columns *Source* and *Target Elements* respectively show the source and target elements of each correspondence. We assume the source models have undergone the following changes:

- The Field physio (physiologicalExamination) of ER model was specialized by adding four subfields: hormonalEx, immunologicalEx, serologicalEx and functionalEx.
- Renaming Task:Control from the BP model to MedicalMonitoring. Renaming Task:Control.

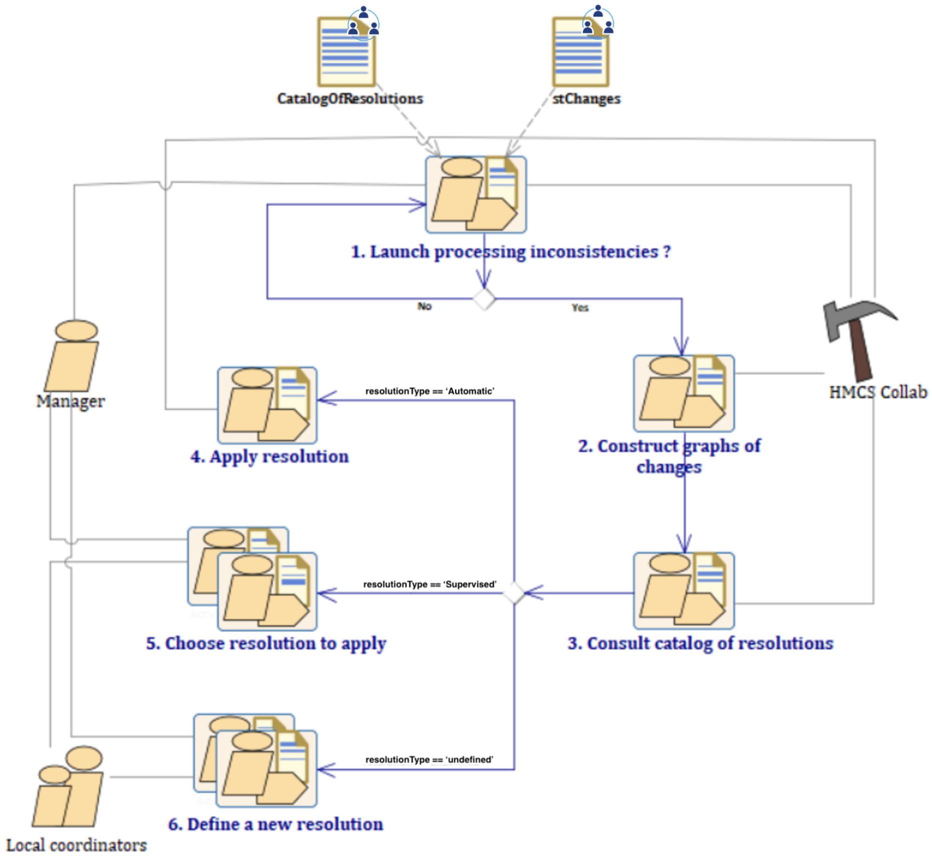



Fig. 9. Detailed tool-supported collaborative activity diagram of *Process the inconsistencies*.

Both these changes are partial. According to 1 and since we do not have any globale changes, we will deal with them.The change concerning the Field physio will be processed at the end since it is a partial change of type addition (case corresponding to line 16 on 1).

Renaming Task:Control requires a collaboration to deal with the inconsistencies it has generated (inconsistencies of C12 and C13 in Figure 10).

The catalog of resolutions (cf. Appendix C) considers two resolutions for this type of change. Let us consider $e$, the element of the model that has undergone the change, and *correspondences(e)*, the set of correspondences involving $e$ ($e$ is source or target of these correspondences). The R2 resolution imposes the removal of correspondences(e)[c] while R6 proposes maintaining all the correspondences being in *correspondences(e)*. It is therefore necessary to start a collaboration to choose

[c]e: element of model that has undergone the change.

**Model of correspondences**

**Table view of M1C**

Search all fields: Enter keyword

| Id | Relationship | Source Elements | Target Elements |
|---|---|---|---|
| | Induction | | |
| C11 | Induction | BP:Task:Consult, | SD:Method:Advice, |
| C12 | Induction | BP:Task:Execute medical analysis, | SD:Method:Cancel Medicine, |
| C13 | Induction | BP:Task:Medical control, | SD:Method:Cancel Medicine, |
| C14 | Induction | BP:Task:Select nurse and provide information, | SD:Method:Feed Patient Medical Data, |
| C15 | Induction | BP:Task:Treatment, | SD:Method:Feed Patient Medical Data, |
| C16 | Induction | BP:Task:Initial Examination, | SD:Method:Feed Patient Medical Data, |
| C17 | Induction | BP:Task:Triage of patient, | SD:Method:Feed Patient Medical Data, |
| C18 | Induction | BP:Task:Select physician and provide information, | SD:Method:Feed Patient Medical Data, |
| C19 | Induction | BP:Task:Receiving information, | SD:Method:Feed Patient Medical Data, |
| C20 | Induction | BP:Task:Hospitalization, | SD:Method:Feed Patient Medical Data, |
| C21 | Induction | BP:Task:Hospitalization, | SD:Method:Interview Patient, |
| C22 | Induction | BP:Task:Triage of patient, | SD:Method:Sort Patient, |
| C23 | Induction | BP:Task:Hospitalization, | SD:Method:Sort Patient, |
| C24 | Induction | BP:Task:Treatment, | SD:Method:Add Medical History, |
| C25 | Induction | BP:Task:Initial Examination, | SD:Method:Add Medical History, |
| | | BP:Task:Select physician and provide | |

Fig. 10. Extract of ED's model of correspondences (M1C).

the appropriate resolution.

The dependency graph generated by the renaming of Task:Control to MedicalMonitoring is illustrated in Figure 11. The Task:Control dependency graph contains elements from the SD model, the ER model and the BP model; hence, the collaboration involves the three local coordinators of these models (Bob, Alice and Claire). In this graph, two elements are related to its root (Task:Control), namely Method:treatment and Field:PrescriptionTreatment. For each node of the graph, Figure 11 specifies the name of the node, its distance from the root (1 for each element) and the correspondences in which it is involved (C12 and C13 for Method:treatment and C13 for Field:PrescriptionTreatment).

Figure 12 illustrates the choice of Alice to handle the inconsistencies of C12. Alice chooses to maintain it and explains her choice to the other designers (text box *Justification*). The same process is carried out for C13. Alice's decision is to maintain it as well. Thus, C12 and C13 are maintained using the new source element Task:MedicalMonitoring.

The Emergency Report model has undergone the changes described in Figure 3. The Field physio (physiologicalExamination) of ER model was specialized by adding four subfields: hormonalEx, immunologicalEx, serologicalEx and functionalEx.

The addition of the fields HormonalEx, ImmunologicalEx, SerologicalEx, and FunctionalEx triggers a matching phase that will modify the model of correspon-

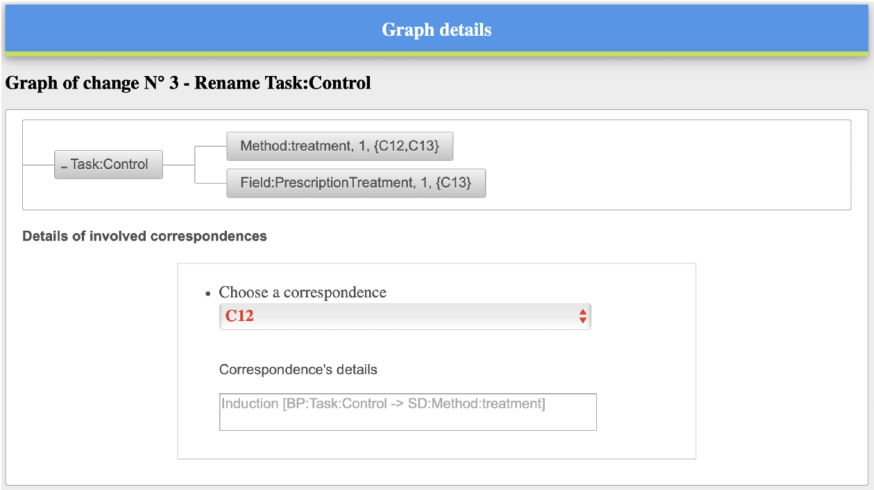



Fig. 11. Task:Control dependency graph.

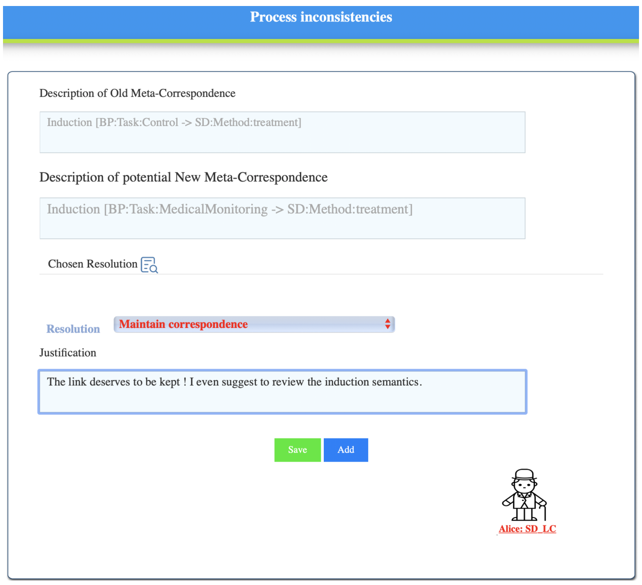



Fig. 12. Extract of the collaboration to manage the inconsistency of C12.

dences.

The correspondences which involve the added classes are handled in a supervised manner. In fact, the catalog of resolutions plans to remove the correspondence involving the parent class and study the possibility of forcing the link for its subclasses. The Field physio is involved in the following correspondence: Induction

[BP:Task:ExecuteAnalysis, SD:Method:checkUp $\longrightarrow$ ER:Field:Physio]. Managing the addition of its subclasses in the ER model requires the agreement of the three concerned local coordinators, Bob, Alice and Claire. They collaborate to take the necessary decisions, in particular regarding the ER model:

(a) Should an induction be defined between (BP:Task:ExecuteAnalysis, SD:Method:checkUp) and ER:Field:HormonalEx? should this be done for all the other subclasses of ER:Field:Physio?
(b) Should ad-hoc entities be added in the BP and SD models?
(c) If yes, should an induction be defined between ER:Field:HormonalEx and the entities added to the other two models?

This collaborative process results in a final consensual decision (according to the catalog of resolutions): the induction correspondence is forced with all the subclasses of Physio; which enriches the M1C with correspondences by the following correspondences :

Induction [BP:Task:ExecuteAnalysis, SD:Method:checkUp $\longrightarrow$ ER:Field:HormonalEx]
Induction [BP:Task:ExecuteAnalysis, SD:Method:checkUp $\longrightarrow$ ER:Field:ImmunologicalEx]
Induction [BP:Task:ExecuteAnalysis, SD:Method:checkUp $\longrightarrow$ ER:Field:SerologicalEx]
Induction [BP:Task:ExecuteAnalysis, SD:Method:checkUp $\longrightarrow$ ER:Field:FunctionalEx]

### 5.2. *HMCS-Collab tool overview*

HMCS-Collab provides a set of plug-ins addressing alignment and collaborative decision-making and consists of two parts: backside, and front-side. The backside of HMCS-Collab supports alignment. It is based on Eclipse Modeling Framework (EMF) of the Eclipse Modeling project (EMP) to enable the propagation of meta-correspondences and Xtext to define the semantics of relationships. The front-side of HMCS-Collab tool is based on web technologies to facilitate actors' collaboration while hiding the technical aspects related to the use of EMF. The front-side of HMCS-Collab is a web application, developed in Java, and in particular in JSF. HMCS-Collab is a proof of concept in the beta-testing phase.

The overall architecture of HMCS-Collab is shown in Figure 13. It contains five modules: two modules (DMT and CollabT) concern the decision-making and collaborative aspects. The remaining three modules concern the alignment of models (MT, CMT and TT). Below, we detail the Decision-Making Tool (DMT) and the Collaboration Tool (CollabT) to stress how local coordinators interact to make collective decisions, whether during matching or when maintaining the global consistency.

The Decision-Making Tool (DMT) implements a set of decision-making policies

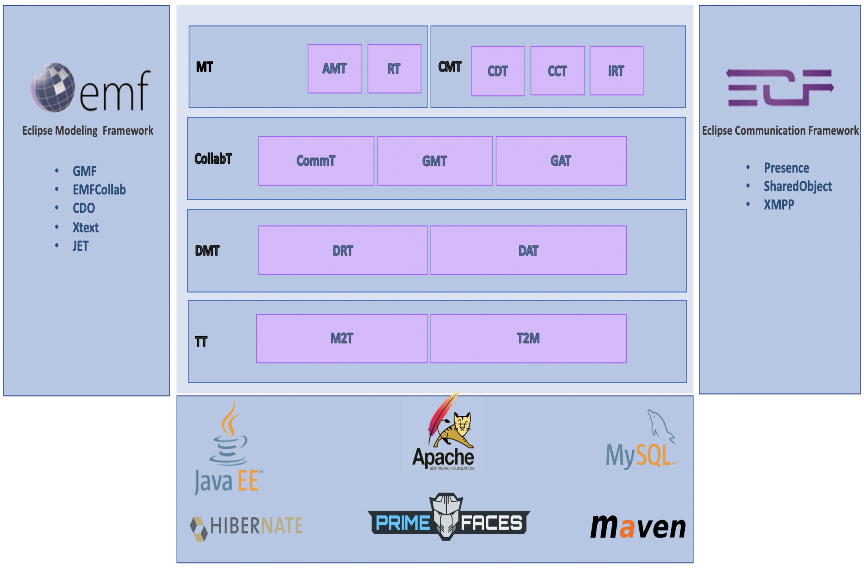



Fig. 13. Functional architecture of HMCS-Collab.

and their aggregation methods. It is divided into two sub-modules: Decision policies Repository Tool (DRT) and Decision Aggregator Tool (DAT). DMT module (right hand side of Figure 14) produces a collaborative decision for a given proposal by exploiting users' data (*UDB*), implemented decision-making policies (*DMP*), proposals (*PDB*) and their evaluations (*EDB*). The four data storages cited above are accessed by four distinct managers (respectively *UDB Manager*, *DMP Manager*, *PDB Manager* and *EDB Manager*). *UDB extractor* extracts for each proposal (1.a), the list of concerned users (1.b). Then, this list is transferred to the *Notification Center* (2.a) that notifies the concerned users (2.b). Afterward, users individually assess proposals and provide decisions (3.a) by *Decision Assessment* service. These decisions are stored in EDB via EDB Manager (3.b). Finally, *Decisions Aggregator* produces a group decision by combining the individual decisions (4.b) according to the *adopted policy* (4.a).

The Collaboration Tool (CollabT) ensures collaboration mechanisms (for example, rights management, group communication, group awareness) via a Group Management Tool (GMT), a Communication Tool (CommT) and a Group Awareness Tool (GAT).

- GMT (Group Management Tool) manages groups to find out which actors are involved in the collaborative session, point out their roles and assign their access rights.
- CommT (Communication Tool) ensures the communication of stakeholders, through a messaging center.

- GAT (Group Awareness Tool) ensures the stakeholders awareness of the presence of other actors and integrates different notification channels. It incorporates a notification system (Notification Center) concerning the key stages of the alignment and decision-making processes, i.e., assignment of a role, waiting for proposals, awaiting evaluations, completion of the collaboration, etc.

The left hand side of Figure 14 details the functional sequencing of the CollabT module. UDB extractor retrieves from UDB manager (1.2) and Users DB (1.1) the information about users, to constitute the list of members concerned by the collaboration (1.3). This list is sent to the senders of the notifications (dispatcher 1 to dispatcher n) (2.1). A single dispatcher is activated depending on the event associated with the notification (2.2). Once a dispatcher is activated, the Notification Center transmits the notification to the relevant actors who meet the selection constraints of the dispatcher (2.3). The members concerned respond to the notification. They transmit their messages to the Messaging Center (3.1) which then forwards them to the other actors (3.2) by broadcast, multicast or targeted message.

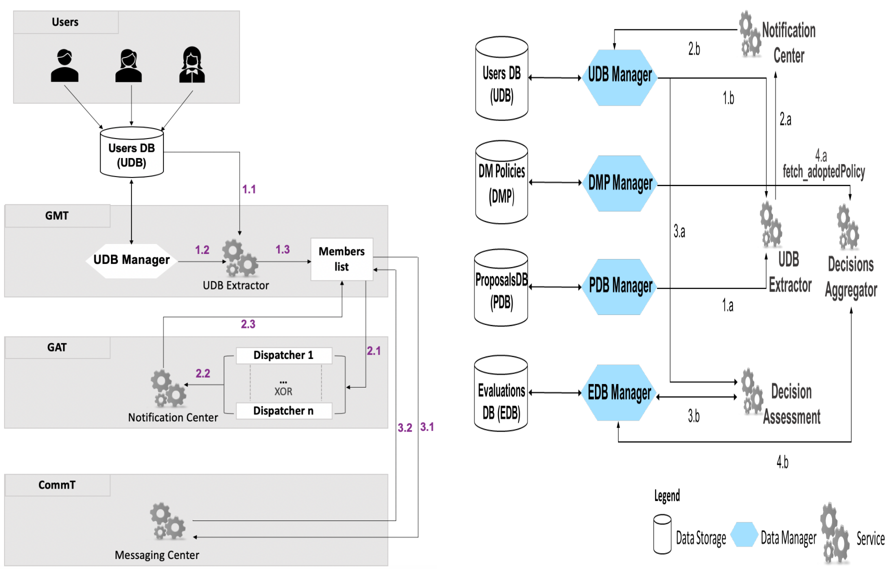



Fig. 14. Left: Sequencing of CollabT. Right: Decision-Making Tool (DMT).

## 6. Discussion

In this Section, we discuss the validity of this experimentation and the threats to validity.

### 6.1. *Experimental evaluation*

An experimental evaluation of CAHM was conducted on the Emergency System of a hospital. Two groups were formed, in two distinct time slots. The first group designed the models in a completely distributed and independent way. A co-author then heads the connection of the partial models to obtain a *golden model of correspondences*, in a centralized way. The second group of users were PhD students (some of them being part-time employed in industry) which connected the models in a collaborative way. The collaboration took place through a shared file, to define and evaluate meta-matches and to manage the consistency of the model of correspondences over time, since, during this experiment, the tool was not covering the collaborative aspects.

To evaluate this implementation, we first compared the *golden model of correspondences* to the model of correspondences obtained collaboratively by the PhD students. Then, we exploited the results of a survey conducted by the PhD students who had acted as local coordinators. The gain of CAHM compared to the centralized approach appears in terms of :

- Efficiency: In the centralized approach, some meta-correspondences did not give rise to correspondences which proves that these meta-correspondences were in fact unnecessary. In contrast, all collaboratively established and validated meta-correspondences are propagated in the M1C. Furthermore, when the alignment is performed by a single actor, fewer correspondences are produced and the alignment requires several rounds to obtain the same model of correspondences as the one obtained by the collaboration of all local coordinators.
- Flexibility: In CAHM, the alignment of models is done collectively and the actions of the actors are combined, so that a solid knowledge of all points of view of the overall system is no longer necessary. The actors' views complement each other. Each actor can thus design his solution more freely, regardless of the other points of view.
- Quality of the decision: in the collaborative approach, each actor has to deal with his point of view, the possibility of discussing the connections with the other designers allows him to understand not only the system as a whole, but also his viewpoint on it, better. Furthermore, the definition of correspondences between metamodels in a two-stage collaboration (proposal, evaluation) distributes the responsibilities between the local coordinators involved and ensures that all the correspondences are relevant to the system (since they are validated by several stakeholders).
- Satisfaction: feedback from stakeholders emphasized that CAHM processes makes their task easier as they do not have to do an in-depth analysis of other (meta)-models.

  Of course, there are some drawbacks to working together in this context. For example, participants noted the problem of communication delays.

Nevertheless, two points are to be noted. First, a certain loss of time can be tolerated when the advantages of the approach are considered, such as (i) reduction of the total effort, (ii) less dependence on one expert, (iii) more flexibility in the way the work is done, (iv) faster identification of inconsistencies. Second, this highlights the need of a support tool. Aware of that, we are currently improving HMCS-Collab to obtain an efficient suite.

For the time being, we have not taken into account the time needed to perform the alignment, as we have focused on the gain, in terms of quality of the alignment in particular, of a collaborative approach. Of course, the study of alignments proposed in the field of ontologies and the automation that it could support is to be put in perspective with a collaborative approach. But we consider that the latter will always be necessary to complete the former.

### 6.2. *Threats to validity*

Our approach presents some other threats to validity that constitute, according to their nature, a possible bias in (i) internal validity, or (ii) external validity [54]. Internal validity measures the fact that if a relationship is observed between the treatment and the result, it is a causal relationship and not the result of a factor on which we have no control or that we have not measured. External validity concerns the generalization of results. It assesses whether, if there is a relationship of cause and effect, the result of the study can be generalized outside its scope.

- Internal validity: In our experimental validation, we involved participants from academia and industry. The description of the ED case study was elaborated in collaboration with a medical staff and the source models were designed by research teams working closely in the fields of these models. This allowed us to ensure these models respect the building rules for the models of these viewpoints.

  Regarding the actors who implemented the experiment, we chose doctoral students in computer science but with varied knowledge in model-driven engineering to see the feasibility of the approach even with novice actors.
- External validity. Our approach was implemented on models from different viewpoints (software design, business process, multi-agent system, relational database, presentation). We are convinced that the approach is applicable to any other DSL, provided that it conforms to the Meta Object Facility (MOF) metamodel [43] and that semantic links can be defined between the elements of these models. However, we have not yet performed the study with other categories of models (e.g., behavioral models, test models, etc.).

## 7. Related work

In this section, we first discuss related work in modeling of group decision-making (GDM), then we present major works dealing with collaborative alignment of heterogeneous models.

### 7.1. *GDM modeling*

A GDM process is a collaborative work where stakeholders aim to produce a co-decision. A GDM process follows five steps as defined in [1, 36] : (i) define the problem; (ii) identify problem parameters (*alternatives* and *selection criteria*[d]); (iii) establish evaluations (estimate alternatives according to all criteria); (iv) select decision-making method, and (v) aggregate evaluations (provide a final aggregated evaluation allowing stakeholders to make a decision). This process may vary depending on the methods used, for the AHP method, for example presented in [17].

Several approaches deal with GDM modeling. Some of them focus on consensus reaching [33, 38, 37, 15] while others propose different method to aggregate alternatives. Collaboro [33], OntoGDSS [13], DMO [34], Metamodel of Malavolta [39], DSO [47] and CRP approach (Consensus Reaching Process) [38] provide features including concepts and relationships for GDM description. To compare these approaches, we analyzed how they manage the following aspects:

- *Organization of Alternatives* (OA): does the approach support dependencies between alternatives, if any?
- *Selection Criteria of alternatives* (SC): does the approach specify criteria to evaluate alternatives?
- *Method of alternatives Aggregation* (MA): does the approach support any aggregation method to come up with a collective decision?
- *existence of a Support Tool* (ST): does the approach provide a tool?

OntoGDSS, DSO and DMO provide ontologies supporting the definition of at least a selection criterion. However, they do not offer any tool for enacting the GDM process. DSO was developed independently of the decision-making aggregation method. Collaboro's main goal is to define new Domain-Specific-Languages collaboratively. Its metamodel is generic and can thus be applied to various group decision-making problems. It has a dedicated tool which only adopts a consensus-based policy; actors need to agree on all of their proposals. The metamodel of Malavolta is designed independently of any decision-making method. It defines a generic concept for the aggregation method but does not propose concrete methods. CRP approach deals with large-scale group decision-making based on bounded confidence and social network to manage experts' opinions. It uses a fast unfolding algorithm to reduce the dimension of the large-scale. The experts' weights are obtained by social network analysis. This approach incorporates the Manhattan

[d] A selection criterion can be any type of information that enables the alternatives comparison

distance, and a feedback mechanism to adjust experts' opinions based on bounded confidence and social network when the experts do not reach a consensus.

Table 1. Comparison of main approaches of GDM modeling

| **Approach\Criterion** | **OA** | **SC** | **MA** | **ST** |
|---|---|---|---|---|
| Collaboro [33] | ✓ | ∅ | ✓ | ✓ |
| OntoGDSS [13] | ? | ✓ | ✓ | ∅ |
| Metamodel of Malavolta [39] | ✓ | ✓ | ∅ | ✓ |
| DMO [34] | ∅ | ✓ | ✓ | ∅ |
| DSO [47] | ∅ | ✓ | ∅ | ∅ |
| CRP approach [38] | ✓ | ✓ | ✓ | ? |
| MMCollab | ✓ | ✓ | ✓ | ✓ |

∅: Not supported, ✓: Well Supported, ~: Basic support, ?: No information found

Table 1 sums-up the features proposed per approach. None of them covers all of the aspects defined above. the metamodel of Malavolta, Collaboro and the CRP approach stand out, but the former does not provide concrete methods for preferences aggregation nor a way to evolve the proposals during the GDM process, whereas the two other approaches offer a unique method of alternatives aggregation (i.e. consensus). Besides, in Collaboro, there are no criteria set for alternatives selection. In the last line, we can see how MMCollab (part of our proposed approach) responds to the given criteria.For the *Organization of Alternatives* feature, MMCollab provides the conflict and specialization relationships. For the feature *Selection Criteria*, MMCollab allows to distinguish the parties involved in decision-making by including mechanisms (*SelectionCriteriaType*: eg expertise, involvement) for the selection of subsets of actors and for the weighting of the preferences of the actors. For the criterion *Methods of Alternatives Aggregation*, the instantiation of the concept *GDMPattern* makes it possible to define a configurable and scalable set of group decision policies, adaptable according to the intrinsic characteristics of the groups and the specificities collaborative situations.

### 7.2. *Collaborative Alignment*

Alignment of heterogeneous models may be achieved for various objectives: synchronization [22], traceability [28], global consistency management [35], model mapping, etc. We used it specifically for a consistency management purpose during multi-model design. Several approaches deal with heterogeneous models alignment, for instance: VirtualEMF [16], AHM [24], EMFViews [9], Shosha's work [50], OpenFlexo [31], and CIMA [55]. A comparison of the approaches supporting matching ( i.e., correspondences definition) has been proposed in [4] while in [24] a comparison of approaches dealing with heterogeneous models consistency management has been drawn.

With the emergence of new approaches and technologies (such as Agile Methods or DevOps), collaboration reveals to be a key character of software engineering [41, 27, 19]. As no approach is completely automatic, human effort is always required during its implementation. In a context of model-driven design by heterogeneous viewpoints, this effort cannot come from a single actor (given the difficulty of mastering all viewpoints), but rather from a set of actors, with diverse and complementary knowledge. Thus, approaches differ depending on how they support, if any, the collaborative aspect when managing inter-model consistency.

The collaborative aspect in heterogeneous model alignment approaches is poorly supported. Several approaches do not target it at all, assuming that an expert can perform all the manual tasks alone. Other approaches do not detail their alignment mechanisms. As a result, they describe alignment techniques, but the proposed tools do not manage collaboration nor multi-use.

Besides CAHM, only Shosha's work and CIMA handle collaboration, but these two approaches support a single type of relationship between model elements (similarity) and do not deal with consistency in case of evolution. Table 2 summarizes the supported fields by approach. Only CAHM considers both the consistency management during evolutions and collaboration support. Indeed, on the one hand, CAHM maintains consistency during evolution by offering a set of resolution recommendations based on an expandable catalog containing automatic and supervised resolutions. On the other hand, CAHM supports the collaboration and participation of business actors in the development and maintenance of the correspondence model, this guarantees that the identified correspondences reflect the real business interests.

Table 2. Comparison of main approaches of Collaborative Alignment

| **Approach\Criterion** | **Change Detect** | **Resolve Inconsistency** | **Collaboration** |
|---|---|---|---|
| VirtualEMF [16] | ~ | ∅ | ∅ |
| AHM [24] | ✓ | ✓ | ∅ |
| EMFViews [9] | ~ | ∅ | ∅ |
| Shosha's work [50] | ∅ | ∅ | ✓ |
| OpenFlexo [31] | ~ | ~ | ∅ |
| CIMA [55] | ∅ | ∅ | ✓ |
| CAHM | ✓ | ✓ | ✓ |

∅: Not supported, ✓: Well Supported, ~: Basic support, ?: No information found

## 8. Conclusion and Future Work

In this paper, we have described CAHM (Collaborative Approach for Heterogeneous Matching), a collaborative approach allowing stakeholders to get a global vision of a system's models, based on the viewpoints that describe this system. This global model is crucial to ensure the consistency of the system, especially in case of evolution of some partial models. The main advantage of the proposed approach is its support to actors' interactions and collaboration, allowing them to make a group decision. It also leverages their participation in the development of inter-model correspondences, and in their maintenance in case of model evolution. This collaborative participation guarantees the consistency of the system's model, since it ensures that established correspondences really meet the needs of stakeholders. To this end, CAHM provides (i) a semi-automatic process where stakeholders collaboratively establish typed links (meta-correspondences) at the metamodel level by discussing them and judging their relevance, (ii) a tooled process for propagating these meta-correspondences at the model level (producing a model of correspondences which reflects the consistency of the overall system), (iii) a way to capture changes at both metamodel and model levels, and to manage their impacts on the model of correspondences and on the other source models. So CAHM allows the stakeholders to handle the matching collaboratively and the evolutions of partial design models of a system to ensure its global consistency.

We do not claim the completeness of the approach. Conceptual, technical and application limitations have to be mentioned. We have set several limiting hypotheses to frame our approach: for example, only the changes fixed via the metamodel of correspondences are supported. In addition, we do consider a model per viewpoint and assume that the source models are all heterogeneous. Another limitation concerns the provided tool. Its effectiveness and its ability to lighten human efforts depend mainly on the relevance of the semantics of the handled relationships.

Regarding the validation of the approach, we implemented it through concrete examples and a representative but simplified case study. As specified, the (meta-)models of the case study have average sizes which do not correspond to the actual sizes of models of large-scale industrial projects. This validation allowed us to verify the applicability of the approach and to measure its strengths. But it does not guarantee the scalability of the approach nor its applicability by large sets of actors.

As future work, we want first to integrate the types of changes that were left out when defining the general model alignment process (e.g., partial changes in metamodels, composite changes, etc.).

We consider then that, as the human actor is essential during the running of CAHM, since its process is based on human knowledge (for the formal definition of semantic relations, the definition of correspondences and the treatment of inconsistencies), collaboration and decision-making can still be improved. We are convinced that the application of Artificial Intelligence together with model-driven engineering could lighten human labor in a collaborative perspective [42]. Our goal is therefore to extend our approach by integrating machine learning techniques for managing inter-model consistency. This can be done, examples given, by the definition of a learning method to help in the definition of semantic relations, by the definition of a system of recommendations for the choice of decision policies and even for the co-decision support, or else thanks to intelligent elements like bots for improving communication [11, 45, 48].

Regarding the scalability of the approach, we intend to conduct CAHM processes with industrial partners using bigger models (i.e., megamodels) and bigger teams.

## Appendix A. ConsentingTogether Decision policy

In this appendix we detail the *ConsentingTogether* decision policy.

## Appendix B. Change processing order algorithm

In this appendix, we present the algorithm for change processing order calculation.

## Appendix C. catalog of inconsistencies resolutions

This appendix presents an extract of the proposed catalog of resolutions. Table 3 shows an expandable list of resolutions classified according to the type of change. Some proposed resolutions can be performed automatically. For the rest, HMCS-Collab searches in the catalog, and proposes the resolutions to the stakeholders. The Resolution column summarizes the possible resolutions, while the Resolution Type column specifies the nature of each resolution (i.e., automatic versus supervised).

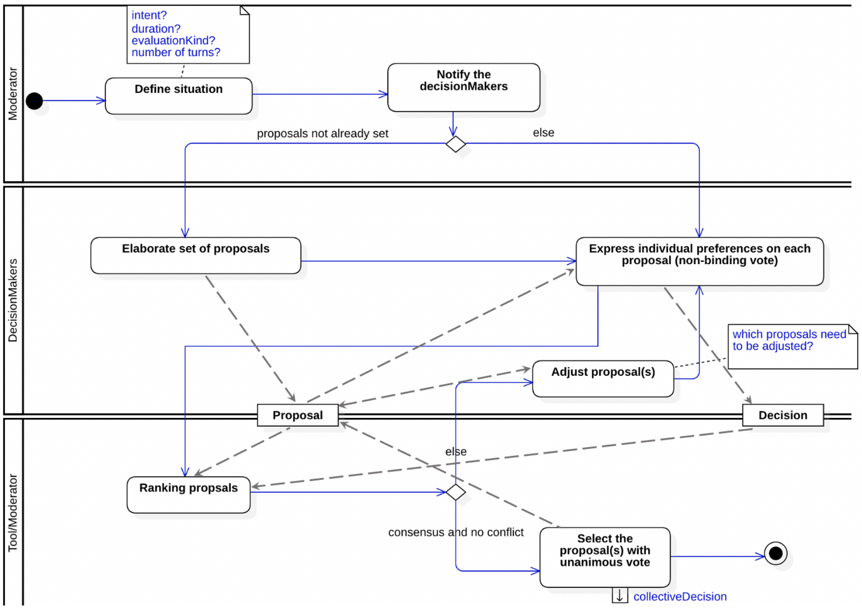
**Name** *Consenting Together*

**Intent** Obtain a solution satisfying all the group's members. This requires refining proposals/decisions until a compromise is reached.

**Applications** This model should be used in case of :
- A small group with converging interests.
- Homogeneous distribution of decision powers within the group.
- No time constraints since the process may take a long time.
- Group members wishing to be involved.

**Known uses** Face-to-face meetings involving an homogeneous group.

**Solution** It enables a consensual process in a face-to-face or distributed environment. The implementation of this decision policy requires the presence of three roles: the moderator, the set of decision-makers and the aggregator (tool or moderator) as summarized below.




First, the moderator defines the collaboration situation (*intent*, *duration*), the *processKind* being set to *consensus2vote* and the *threshold* being set to *strict*. Then, he/she notifies the *decisionMakers*.
The *decisionMakers* must first produce the list of proposals if they are not already set, then they express their individual choices on these proposals using an indicative vote.
A tool (*tool*) (or possibly the moderator) sorts these votes. If the proposals are approved unanimously and are non-conflictual, they are accepted, otherwise group members refine them until they satisfy all members.
The process succeeds if unanimity is reached for at least one proposal after one or more rounds without exceeding the number of rounds allowed and the maximum time allowed for collaboration. Otherwise the collaboration fails and the moderator chooses either to change the policy or to reset the collaboration.

**Related patterns** Negotiating together.

**Algorithm 1:** Calculate change processing order

```
Input  : c // a change
         stChanges // stack of changes
Output : stChanges
if c.type == 'Global' then
    stChanges.insertAtTop(c) // c has the highest priority
else // c.type == 'Partial'
    if c.subtype != 'A' // c.subtype == 'D' or c.subtype == 'M'    then
        ce = c.getChangedElement()
        ce_corresps = getCorresps(ce) // get related elements to ce in M1C
          (directly and indirectly)
        for each corresp in ce_corresps do
            if verifySemantics(corresp) == False then
                ce.getInvalidCorresps().add(corresp)
            end
        end
        if ce.getInvalidCorresps().size() > 0 then
            stChanges.position(c) // Put c in its right position and Shift the
              rest of higher changes by a position to the top; a change c1 has
              higher priority than c if
              c1.getInvalidCorresps()>c.getInvalidCorresps()
        end
    else // c.subtype == 'A'
        stChanges.insertAtBottom(c)// c has the lowest priority
    end
end
```

| Level | Type | Change Subtype | Resolution | Resolution Type |
|---|---|---|---|---|
| Metamodel | Global | Delete MM1 | (**RM1** or **RM2**) and **R1**, with :<br>**RM1** : Remove meta-correspondences (MM1)<br>**R1** : Remove correspondences(M1)<br>**RM2** : Adjust meta-correspondances (MM1) | <br>Automatic<br>Automatic<br>Supervised |
| | | Add MM1 | **RM2** or **RM3**<br>**RM3** : Define meta-correspondences (MM1) | Supervised |
| | | Delete + add MM1 | **RM1** or **RM2** or **RM3** | Supervised |
| Model | Global | Delete M1 | **R1** : Remove correspondences(M1) | Automatic |
| | | Add M1 | **RM4.0** : Propagate meta-correspondences (MM1) | Automatic |
| | | Delete + add M1 | **R1** and **RM4.0** | Automatic |
| Model | Partial | Create element e | **RM4.1** : Propagate meta-correspondences Of me | Automatic |
| | | Delete element e | **R2** or **R3** or **R4** or **R5**, with :<br>**R2** : Remove correspondences(e) if cardinality = 2<br>**R3** : Remove correspondence's extremity if it still holds<br>**R4** : Adjust the other models<br>**R5** : Restore e | Supervised |
| | | Rename element e / Move/ Pull up / Push down | **R2** or **R6**, with :<br>**R2** : Remove correspondences(e) if cardinality = 2<br>**R6** : Maintain correspondences(e) | Supervised |
| | | Inline class c to d / Flatten class c to d | **R2** or (**R2** and **R7**), with :<br>**R2** : Remove correspondences(c)<br>**R7** : Force correspondences(d) | Supervised |

Table 3. Extract of the catalog of inconsistencies resolutions

## References

[1] D Baker et al. *Guidebook to decision-making methods. Developed for the Department of Energy.* Tech. rep. WSRC-IM-2002-00002, 2001.

[2] Saloua Bennani et al. "A collaborative decision approach for alignment of heterogeneous models". In: *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE).* IEEE. 2019, pp. 112–117.

[3] Saloua Bennani et al. "A Formalization of Group Decision Making in Multi-Viewpoints Design". In: *Computer and Information Science* 13.1 (2020), pp. 1–58.

[4] Saloua Bennani et al. "Collaborative model-based matching of heterogeneous models". anglais. In: *International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanjing, China.* IEEExplore, 2018.

[5] Jean Bézivin and Ivan Kurtev. "Model-based technology integration with the technical space concept". In: *Proceedings of the Metainformatics Symposium.* Vol. 20. Springer-Verlag, 2005, pp. 44–49. URL: https://hal.archives-ouvertes.fr/hal-00483587.

[6] Matthias Bräuer and Henrik Lochmann. "An ontology for software models and its practical implications for semantic web reasoning". In: *European Semantic Web Conference.* Springer. 2008, pp. 34–48.

[7] Johan Brichau and Theo D'Hondt. "Introduction to Aspect-Oriented Software Development". In: *European Network of Excellence on Aspect-Oriented Software Development, August 2005* (2005).

[8] Manfred Broy et al. "Seamless model-based development: From isolated tools to integrated model engineering environments". In: *Proceedings of the IEEE* 98.4 (2010), pp. 526–545.

[9] Hugo Bruneliere et al. "EMF views: A view mechanism for integrating heterogeneous models". In: *International Conference on Conceptual Modeling.* Ed. by Paul Johannesson et al. Springer. 2015, pp. 317–325. ISBN: 978-3-319-25264-3. DOI: 10.1007/978-3-319-25264-3_23.

[10] Hugo Brunelière et al. "Emf views: A view mechanism for integrating heterogeneous models". In: *International Conference on Conceptual Modeling.* Springer. 2015, pp. 317–325.

[11] Jordi Cabot et al. "Cognifying Model-Driven Software Engineering". In: Software Technologies: Applications and Foundations. Jan. 2018, pp. 154–160. ISBN: 978-3-319-74729-3. DOI: 10.1007/978-3-319-74730-9_13.

[12] João Carneiro et al. "Arguing with behavior influence: a model for web-based group decision support systems". In: *International Journal of Information Technology & Decision Making* 18.02 (2019), pp. 517–553.

[13] Junyi Chai and James NK Liu. "An ontology-driven framework for supporting complex decision process". In: *World Automation Congress (WAC), 2010.* IEEE. 2010, pp. 1–6.

[14] Mohammad Chami et al. "Towards Solving MBSE Adoption Challenges: The D3 MBSE Adoption Toolbox". In: *INCOSE International Symposium* 28 (July 2018), pp. 1463–1477. DOI: 10.1002/j.2334-5837.2018.00561.x.

[15] Xiangrui Chao et al. "Large-scale group decision-making with non-cooperative behaviors and heterogeneous preferences: An application in financial inclusion". In: *European Journal of Operational Research* 288.1 (2021), pp. 271–293. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2020.05.047. URL: https://www.sciencedirect.com/science/article/pii/S0377221720304938.

[16] Cauê Clasen, Frédéric Jouault, and Jordi Cabot. "VirtualEMF: a model virtualization tool". In: *Advances in Conceptual Modeling. Recent Developments and New Directions*. Ed. by Olga De Troyer et al. Springer. 2011, pp. 332–335. ISBN: 978-3-642-24574-9. DOI: 10.1007/978-3-642-24574-9_43.

[17] Ricardo M Czekster et al. "Decisor: A software tool to drive complex decisions with analytic hierarchy process". In: *International Journal of Information Technology & Decision Making* 18.01 (2019), pp. 65–86.

[18] Marcos Didonet Del Fabro and Patrick Valduriez. "Towards the efficient development of model transformations using model weaving and matching transformations". In: *Software & Systems Modeling* 8.3 (2009), pp. 305–324. ISSN: 1619-1374. DOI: 10.1007/s10270-008-0094-z. URL: https://doi.org/10.1007/s10270-008-0094-z.

[19] Davide Di Ruscio et al. "Envisioning the future of collaborative model-driven software engineering". In: *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE. 2017, pp. 219–221.

[20] Edsger Wybe Dijkstra. *A discipline of programming*. Vol. 1. prentice-hall Englewood Cliffs, 1976.

[21] Zinovy Diskin and Harald König. "Incremental consistency checking of heterogeneous multimodels". In: *Federation of International Conferences on Software Technologies: Applications and Foundations*. Springer. 2016, pp. 274–288.

[22] Zinovy Diskin, Harald König, and Mark Lawford. "Multiple model synchronization with multiary delta lenses with amendment and K-Putput". In: *Formal Aspects of Computing* 31.5 (2019), pp. 611–640. ISSN: 1433-299X. DOI: 10.1007/s00165-019-00493-0. URL: https://doi.org/10.1007/s00165-019-00493-0.

[23] Mahmoud El Hamlaoui et al. "A MDE Approach for Heterogeneous Models Consistency". anglais. In: *International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), Madeira*. SciTePress, 2018.

[24] Mahmoud El Hamlaoui et al. "AHM: Handling Heterogeneous Models Matching and Consistency via MDE". In: *International Conference on Evaluation of Novel Approaches to Software Engineering*. Springer. 2018, pp. 288–313.

[25] Mahmoud El Hamlaoui et al. “Alignment of viewpoint heterogeneous design models: Emergency Department Case Study”. In: *4th International Workshop On the Globalization of Modeling Languages (GEMOC 2016) co-located with ACM/IEEE MODELS 2016*. Saint-Malo, France, 2016, pp. 18–27.

[26] Robert France and Bernhard Rumpe. “Model-driven development of complex software: A research roadmap”. In: *Future of Software Engineering (FOSE’07)*. IEEE. 2007, pp. 37–54.

[27] Mirco Franzago et al. “Collaborative model-driven software engineering: a classification framework and a research map”. In: *IEEE Transactions on Software Engineering* 44.12 (2017), pp. 1146–1175.

[28] Ismenia Galvao and Arda Goknil. “Survey of Traceability Approaches in Model-Driven Engineering”. In: *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*. IEEE. 2007, pp. 313–324. DOI: 10.1109/EDOC.2007.42.

[29] Erich Gamma. *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.

[30] Fahad R Golra et al. “Addressing Modularity for Heterogeneous Multi-model Systems Using Model Federation”. In: *Companion Proceedings of the 15th International Conference on Modularity*. MODULARITY Companion 2016. Malaga, Spain: ACM, 2016, pp. 206–211. ISBN: 978-1-4503-4033-5. DOI: 10.1145/2892664.2892701.

[31] Fahad R Golra et al. “Addressing Modularity for Heterogeneous Multi-model Systems using Model Federation”. In: *Companion Proceedings of the 15th International Conference on Modularity*. ACM. ACM, 2016, pp. 206–211. ISBN: 9781450340335. DOI: 10.1145/2892664.2892701. URL: https://doi.org/10.1145/2892664.2892701.

[32] Regina Hebig, Djamel Eddine Khelladi, and Reda Bendraou. “Surveying the Corpus of Model Resolution Strategies for Metamodel Evolution”. In: *2015 Asia-Pacific Software Engineering Conference (APSEC)*. IEEE. Dec. 2015, pp. 135–142. DOI: 10.1109/APSEC.2015.40.

[33] Javier Luis Cánovas Izquierdo and Jordi Cabot. “Collaboro: A collaborative (meta) modeling tool”. In: *PeerJ Computer Science* 2 (2016), e84.

[34] Elena Kornyshova and Rébecca Deneckère. “Decision-making ontology for information system engineering”. In: *International Conference on Conceptual Modeling*. Springer. 2010, pp. 104–117.

[35] Max E Kramer. “A Generative approach to change-driven consistency in multi-view modeling”. In: *Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures*. 2015, pp. 129–134.

[36] Irwin P Levin, Mary E Huneke, and John D Jasper. “Information processing at successive stages of decision making: Need for cognition and inclusion–exclusion effects”. In: *Organizational behavior and human decision processes* 82.2 (2000), pp. 171–193.

[37] Guangxu Li, Gang Kou, and Yi Peng. "A Group Decision Making Model for Integrating Heterogeneous Information". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.6 (2018), pp. 982–992. DOI: 10.1109/TSMC.2016.2627050.

[38] Yanhong Li et al. "Consensus reaching process in large-scale group decision making based on bounded confidence and social network". In: *European Journal of Operational Research* (2022).

[39] Ivano Malavolta, Henry Muccini, et al. "Enhancing architecture design decisions evolution with group decision making principles". In: *International Workshop on Software Engineering for Resilient Systems*. Springer. 2014, pp. 9–23.

[40] David A Marca and Clement L McGowan. *SADT: structured analysis and design technique*. McGraw-Hill, Inc., 1987.

[41] Pieter J Mosterman and Justyna Zander. "Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems". In: *Software & Systems Modeling* 15.1 (2016), pp. 5–16.

[42] Gunter Mussbacher et al. "Towards an Assessment Grid for Intelligent Modeling Assistance". In: *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. MODELS '20. Virtual Event, Canada: Association for Computing Machinery, 2020. ISBN: 9781450381352. DOI: 10.1145/3417990.3421396. URL: https://doi.org/10.1145/3417990.3421396.

[43] Object Management Group (OMG). *Meta Object Facility (MOF) Specification*. Tech. rep. 2000.

[44] Object Management Group (OMG). *Software & Systems Process Engineering Meta-Model Specification V2.0*. Tech. rep. 2008. URL: http://www.omg.org/spec/SPEM/2.0.

[45] Sara Perez et al. "The rise of the (modelling) bots: Towards assisted modelling via social networks". In: *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. Oct. 2017, pp. 723–728. DOI: 10.1109/ASE.2017.8115683.

[46] Manny Rayner et al. "OMG unified modeling language specification". In: *Version 1.3,© 1999 Object Management Group, Inc.* Citeseer. 2005.

[47] Justin A Rockwell et al. "A Decision Support Ontology for collaborative decision making in engineering design". In: (2009).

[48] Rijul Saini et al. "DoMoBOT: A Bot for Automated and Interactive Domain Modelling". In: *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. MODELS '20. Virtual Event, Canada: Association for Computing Machinery, 2020. ISBN: 9781450381352. DOI: 10.1145/3417990.3421385. URL: https://doi.org/10.1145/3417990.3421385.

[49] Ramy Shosha, Christophe Debruyne, and Declan O'Sullivan. "Towards an adaptive tool and method for collaborative ontology mapping". In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer. 2015, pp. 319–328.

[50] Ramy Shosha, Christophe Debruyne, and Declan O'Sullivan. "Towards an adaptive tool and method for collaborative ontology mapping". In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Ed. by Ioana Ciuciu et al. Springer, 2015, pp. 319–328. ISBN: 978-3-319-26138-6. DOI: 10.1007/978-3-319-26138-6_35.

[51] Pavel Shvaiko and Jérôme Euzenat. "Ontology matching: state of the art and future challenges". In: *IEEE Transactions on knowledge and data engineering* 25.1 (2011), pp. 158–176.

[52] Nirnaya Tripathi et al. "An anatomy of requirements engineering in software startups using multi-vocal literature and case survey". In: *Journal of Systems and Software* 146 (2018), pp. 130–151. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2018.08.059. URL: http://www.sciencedirect.com/science/article/pii/S0164121218301729.

[53] Frank Trollmann et al. "Formalizing model consistency based on the abstract syntax". In: *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*. ACM. 2011, pp. 79–84. DOI: 10.1145/1996461.1996498.

[54] Claes Wohlin et al. *Experimentation in Software Engineering*. Springer Science & Business Media, 2012.

[55] Xingsi Xue and Jianhua Liu. "Collaborative ontology matching based on compact interactive evolutionary algorithm". In: *Knowledge-Based Systems* 137 (2017), pp. 94–103. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2017.09.017. URL: https://doi.org/10.1016/j.knosys.2017.09.017.