This is an author's version published in: http://oatao.univ-toulouse.fr/24869

Official URL

DOI : https://doi.org/10.1109/WETICE.2019.00032

To cite this version: Bennani, Saloua and Ebersold, Sophie and El Hamlaoui, Mahmoud and Coulette, Bernard and Nassar, Mahmoud *A collaborative decision approach for alignment of heterogeneous models.* (2019) In: 28th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2019), 12 June 2019 - 14 June 2019 (Capri, Italy).

A collaborative decision approach for alignment of heterogeneous models

Saloua Bennani
ADMIR/ IRIT
Morocco/ France
saloua.bennani@irit.fr

Sophie Ebersold
IRIT Laboratory
Toulouse, France
sophie.ebersold@irit.fr

Mahmoud El Hamlaoui
ADMIR, ENSIAS, Rabat IT Center
Rabat, Morocco
mahmoud.elhamlaoui@um5.net.ma

Bernard Coulette
IRIT Laboratory
Toulouse, France
bernard.coulette@irit.fr

Mahmoud Nassar
ADMIR, ENSIAS, Rabat IT Center
Rabat, Morocco
mahmoud.nassar@um5.net.ma

***Abstract*—Design of complex systems goes through a multi-view paradigm in which separate teams, from different business viewpoints, build partial models describing the system. As they are expressed in different languages, these partial models are called heterogeneous models. To maintain the global system's consistency, we propose a collaborative approach that combines Group Decision Making (GDM) and Model-Based Engineering. This paper presents a metamodel for collaborative decision elaboration via a set of decision policies which are instances of GDM patterns. Our approach is illustrated with a hospital Emergency Department case study and is supported by a tool allowing models alignment through GDM based processes.**

***Keywords*—collaboration, group decision-making, pattern, heterogeneous models, model-based engineering, model alignment, model matching.**

I. INTRODUCTION

Complex systems modeling involves designers from distinct business domains. These designers generally produce partial models, according to their viewpoints, using domain specific languages (DSL) [1]. This leads to heterogeneous models that are conform to different metamodels. (We do not consider semantics or concrete syntax aspects, so a DSL is seen here as a metamodel). For example, modeling a car may implies electronic, mechanic and software models. Working with these models separately can lead to some inconsistencies. For instance, some contradicting design choices or redundant concepts among models may raise inconsistencies if the partial models are not updated together. To avoid such a problem, and to ensure a global coherence among the partial models of a system, a solution may be to capture the inter-model *correspondences*, also called *model alignment* by analogy to *ontology alignment* [2]. Actually, a *correspondence* is a semantic relationship relating at least two elements. For example, a *similarity* between a concept ”a” from model ”A” and a concept ”b” from model ”B”. *Model alignment* allows to first establish correspondences among models (also called *model matching*) and second to manage the global consistency when models evolve. Actually, the validity of a correspondence might be questioned whenever a model evolves and thus allows to detect and repair the inconsistencies. There are several approaches for heterogeneous models alignment. Since model alignment in itself is not the purpose of this paper, we briefly mention these approaches limitations. Actually, either the studied approaches allow a set of frozen relationships to relate models [3]–[5], or suppose that a single actor (i.e. system's expert) can perform alone the alignment [6]–[9]. 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. Indeed, no matter how expert in the application domain the actor performing the alignment is, he cannot grasp technical and functional concerns of all involved viewpoints, especially in the case of strongly heterogeneous models. So, involving all concerned actors allows the capture of wider knowledge and preoccupations, and facilitates model alignment, while ensuring consistency and reliability.

Furthermore, although industrial practices favour collaborative design, the collaborative alignment of heterogeneous models is still done, in practice, informally which is fastidious and error-prone. To cope with this need of collaboration, we proposed an approach for semi-automating the Collaborative Alignment of Heterogeneous Models [10]. It combines Model-Based Engineering (MBE) and Group Decision-Making (GDM) to establish and maintain correspondences among heterogeneous models. It is based on a metamodel of collaboration, called MMCollab and introduced in [10]. In this paper, we propose an extension of MMCollab by integrating co-decision policies. For that purpose, we describe a set of GDM Patterns. This paper also presents a Decision Making Tool (DMT) which has been added to our prototype to allow co-decision elaboration.

The rest of this paper is structured as follows. We give in Section II an overview of the related work addressing GDM. Section III presents the proposed GDM modeling, specifically the CollectiveDecision package, and five decision policies instantiated from the GDM patterns. In Section IV, the proposed approach is enacted on an Emergency Department management system to validate its applicability to collabora-

DOI 10.1109/WETICE.2019.00032

tive models matching. Section V presents the architecture of the Decision Making Tool. Finally, we conclude and give some perspectives in Section VI.

II. RELATED WORK

The approach described in this paper essentially brings together two strands of work: *model alignment* and GDM. Each of them comes with its own background and related literature. Since this paper deals with GDM modeling, we devote this section to approaches describing GDM knowledge.

A GDM process is a collaborative work where stakeholders aim to produce a co-decision. It usually goes through five stages as defined in [11,12] : (i) Define the problem, (ii) Identify problem parameters, for instance *alternatives*, *selection criteria*. Notice that a *Selection criterion* can be any type of information that enables the evaluation of alternatives and their comparison, e.g. intrinsic characteristics, stakeholders' opinions, potential consequences of alternatives. (iii) Establish evaluations, i.e. estimate alternatives according to all criteria, (iv) Select decision making method, and (v) Aggregate evaluations (provide a final aggregated evaluation allowing decision).

Several approaches deal with GDM modeling. Collaboro [13], OntoGDSS [14], DMO [15] and DSO [16] provide features including concepts and relationships for GDM description. Cited approaches facilitate the management of co-decision processes, from alternatives generation, evaluation and opinions interactions to decision aggregation. To compare these approaches, we analyzed how they manage the following aspects: *Organization of Alternatives* (OA), *Selection Criteria of alternatives* (SC), *Method of alternatives Aggregation* (MA) and *existence of a Support Tool* (ST):

- OA: does the approach support dependencies between alternatives, if any?
- SC: does the approach specify criteria to evaluate alternatives?
- MA: does the approach support several aggregation method to come up to a collective decision?
- ST: does the approach provide a supporting tool?

OntoGDSS, DSO and DMO are ontologies supporting the definition of at least a selection criterion. However, they do not provide any tool for enacting the GDM process. DSO was developed independently of the decision making aggregation method. Collaboro's main goal is to collaboratively define new DSLs. 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, thus actors need to agree on all of their proposals.

Table I sums-up the features proposed per approach. None of them covers all of the aspects defined above. DMO and Collaboro stand out, but the former does not provide a supporting tool nor a way to organize dependencies among alternatives, whereas in the latter there are no criteria set for selection and it offers a unique method of alternatives aggregation (i.e. consensus).

TABLE I
COMPARING RELATED WORK IN GDM MODELING

Approach\Criterion	**OA**	**SC**	**MA**	**ST**
Collaboro [13]	✓	∅	✓	✓
OntoGDSS [14]	?	✓	✓	∅
DMO [15]	∅	✓✓	✓✓	∅
DSO [16]	∅	✓✓	∅	∅

∅: Not supported, ✓: Supported, ?: No information found

III. MODELING GDM

To remedy the shortcomings previously identified, we propose the metamodel MMCollab. It can be instantiated to describe each collaborative session where stakeholders make proposals, evaluate or refine them to come up with a collective decision. A description of the kernel of MMCollab was done in [10]. Since this release, we have structured MMCollab in packages and enriched it by adding the *CollectiveDecision* package which is dedicated to GDM. Each package covers a part of the collaborative decision making modeling, namely: actors organization *(package Actors)*, proposals organization *(Proposals)*, proposals evaluation *(Evaluation)*, collective decision elaboration *(CollectiveDecision)* and a package for core concepts *(CoreConcepts)*. In Section III.A, we give an overview of MMCollab, then we present the *CollectiveDecision* package in Section III.B. Section III.C presents five decision policies that are instances of *GDMPattern*; the core concept of *CollectiveDecision* package.

A. Overview of MMCollab

Collaboration is the focal point of MMCollab, described in Figure 1. It is a specialization of SPEM's Activity [17] and includes a set of *Proposal*s. A Collaboration is enacted via a *GDMPattern* (this will be detailed in section III.B) and according to this latter, a *finalDecision* is associated to each Proposal at the end of the collaboration.

A Collaboration implies a set of *involvedUsers*, including a *moderator* (*isModerator* attribute of *InvolvedUser*). The role of a moderator is to choose the decision policy (the GDMPattern to be adopted: *adoptedGDMPattern*). By default, the stakeholder who made the first proposal is considered as the collaboration's moderator. A list of eligible decision-makers (*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* (EP) that are either approved or rejected together. Each EP 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 *EP*, he has to justify his choice by a *Comment*. In case he thinks an EP needs to be refined,

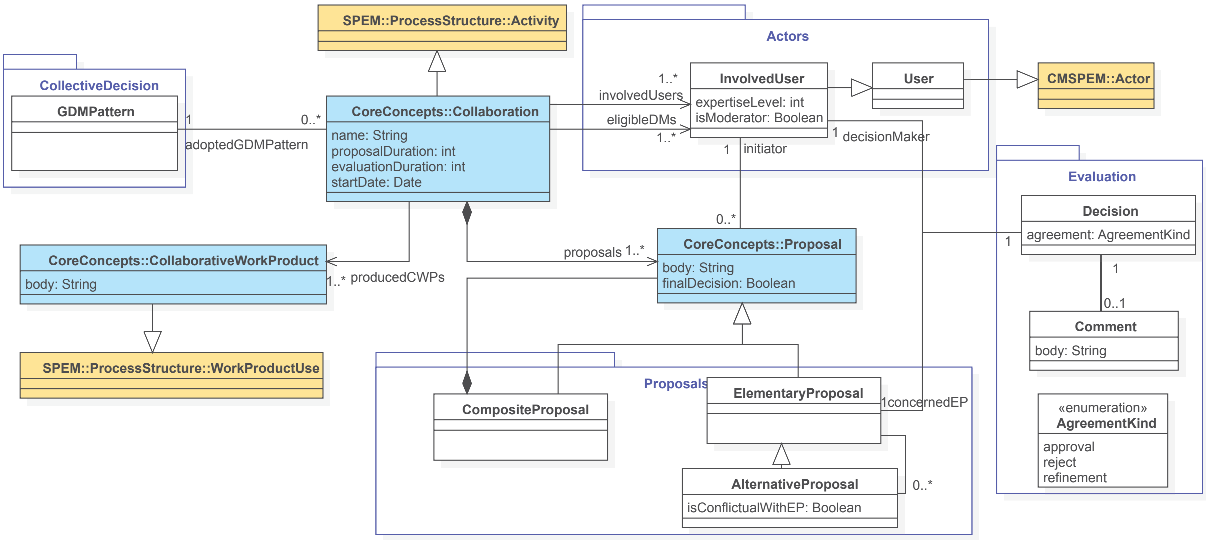

Fig. 1. Overview of metamodel of collaboration (MMCollab)

he provides an *AlternativeProposal* (AP). The attribute *isConflictualWithEP* of an *AP* specifies if this AP is conflicting with the EP to which it is attached.

The value of *finalDecision* attribute of a Proposal is set by aggregating the individual *Decisions* according to the *adoptedGDMPattern*; Considering *EP* having an associated *AP*, this AP has also to be evaluated before setting the finalDecision of its *EP*. In case this AP is conflicting with its associated EP, it is either EP or AP that is maintained. A Collaboration produces *CollaborativeWorkProduct*(s) that gathers the set of approved proposals.

B. The CollectiveDecision package

The *CollectiveDecision* package, shown in Figure 2, gathers concepts needed to describe the elaboration of a collective decision in a GDM context. The main concept of this package is *GDMPattern*. It is a specialization of Pattern. This latter is defined according to the structure widely used in software design to describe patterns [18], i.e., an *intent*, a set of *application* contexts, a set of *known uses*, a *solution* and a reflexive *parent-child* relationship. Besides its inherited characteristics, a *GDMPattern* consists of a *ParticipationMethod* and a *CodecisionMethod*.

ParticipationMethod specifies how stakeholders participate on the decision-making. It is specified via the enumeration *ParticipationType*. It is *democratic* when all stakeholders are involved and *restricted* when only a subset of them is involved. For each *ParticipationMethod*, some *parameter*s could be specified (i.e., *ParameterKind*: stakeholders anonymity and confidence). In case of a restricted participation, the criterion behind stakeholders selection should be specified (either *disponibility* or *expertise*).

CodecisionMethod is determined by three attributes: (i) Thresholds ease group decision making. Indeed, groups may use agreement *threshold* ranges for proposals validation. A *strict* threshold means that a 100% agreement is required whereas *low*, *medium*, *high* thresholds avoid to be contracted by a strict agreement. (ii) The *processKind* specifies the process of proposals evaluation. Since stakeholders may be in different locations, even consensual or negotiation processes give rise to a final vote to capture opinions. Thus, we propose three decision processes stored in the *DecisionProcessKind* enumeration: *directVote*, *consensus2vote* (requires a strict threshold) and *negotiation2vote* (a low, medium or high threshold). (iii) The *preferenceKind* specifies how proposals are evaluated: *rating* or a *yesNo*.

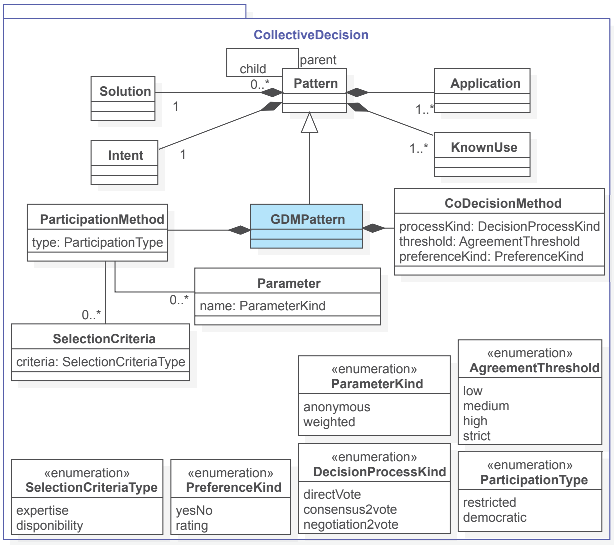

Fig. 2. MMCollab's *CollectiveDecision* package

C. Decision policy as an instance of GDM pattern

Given the description of GDMpatterns, we consider now their instances, which we call *Decision Policy* (DP). Actually, a DP is a combination of instances of elements which compose a GDM pattern (i.e., *ParticipationMethod* and *CoDecisionMethod*) and by transitivity a combination of instances of elements that characterize both of them, namely, type of participation (*type*), decision process (*processKind*), agreement threshold (*threshold*) and preference kind (*preferenceKind*). Combination of these elements allowed us to define five Decision Policies that describe the commonly used policies in GDM (highlighted classes on Figure 3). These five DP can be classified according to their type of participation: Restricted (*RestrictedDP*) vs Democratic (*DemocraticDP*) and also according to the number of turns needed to come up with a decision: *SingleElectionDP* vs *IterativeDP*.

MajorityDeciding is a DemocraticDP. It inherits also from SingleElectionDP since it is performed in a single round. Meaning, if stakeholders did not reach the defined threshold at the end of the collaboration, they either adjust the threshold or have to re-evaluate the proposals. *ConsentingTogether* and *NegotiatingTogether* are IterativeDP, which means they may be repeated until reaching the fixed threshold. *ConsentingTogether* requires a strict threshold (100% agreement) while *NegotiatingTogether* works with a low, medium or high threshold. *Delegating* and *TakingAdvice* are RestrictedDP thus the criteria of stakeholders' selection need to be specified.

These decision policies are not frozen and can be extended as application contexts require by exploring the possible combinations of the elements that compose them.

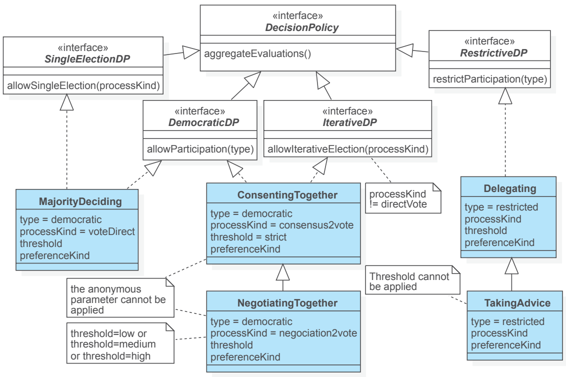

Fig. 3. Decision policies (*GDMPattern* instances) and their dependencies

IV. APPLICATION TO COLLABORATIVE MODEL MATCHING

We apply the proposed GDM modeling on the collaborative model matching of a hospital Emergency Department system (ED) which is a representative example of a complex system. Partial models describing this system were defined in cooperation with emergency doctors of a french hospital. We first present the ED, then recall the collaborative matching process proposed in [10] before applying it to ED system.

A. Emergency Department case study

An ED system is a critical and complex system that affects the daily lives of citizens. Design of such a system implies heterogeneous models associated to different viewpoints. In this paper, due to space constraint, we consider only three of them:

- Software Design (SD): This is an object-oriented model of the system. It describes the ED system as classes having attributes and operations.
- Business Protocol (BP): a model describing the system as a workflow of activities and flows among roles.
- Examination Report (ER): It represents the digital mock-ups of an emergency report as a set of fields.

Models associated to these viewpoints have been elaborated by separate design teams as part of a case study involving several research teams [8]. Figure 4 presents small extracts of these metamodels and their respective models. Complete models and metamodels are available at [19]. SD model contains classes concerning patients, their medical history and diagnostics. Roles and their respective Activities are described in BP model. In ER model, fields that form the medical report are described. (e.g., socialSecurityNumber, clinicalObservations). These models are heterogeneous since they are expressed in distinct DSLs that correspond to different business uses. However, these models may include some common or dependent elements that need to be orchestrated to ensure the system's consistency. In the following, we recall the collaborative matching process used to relate these models.

B. Collaborative matching process overview

This process aims to collaboratively produce *correspondences* among heterogeneous models. Actually, we defined a *correspondence* as a set of elements linked through a *relationship*. Correspondences are defined first at metamodel level (they are called *High Level Correspondence (HLC)*) and then at model level (*Low Level Correspondence (LLC)*). This process involves the following actors: (i) a designer from each concerned viewpoint (called *local coordinator*), (ii) a tool, called HMCS (for Heterogeneous Matching and Consistency management Suite) and (iii) a *semantics expert* who associates a semantics to relationships newly defined with HMCS. This process goes through three main activities:

(1) Set the relationships to be used in correspondences definition. For the examples in Figure 4, there are 3 defined relationships: *Similarity*, *Generalization* and *Induction* (induction indicates a behavioral connection of giving rise).

(2) Produce HLCs: Each local coordinator proposes correspondences at metamodel level that involve meta-elements from his metamodel. For each HLC, he specifies the involved meta-element(s) (i.e. meta-elements from his metamodel and the other ones) and the relationship which links them. The proposed HLCs are later collaboratively evaluated. In Figure 4, three HLCs are emphasized. For example HLC1: *Similarity [ER:Field ↔ SD:Attribute]* means that a similarity relationship exists between the meta-element Field from ER metamodel

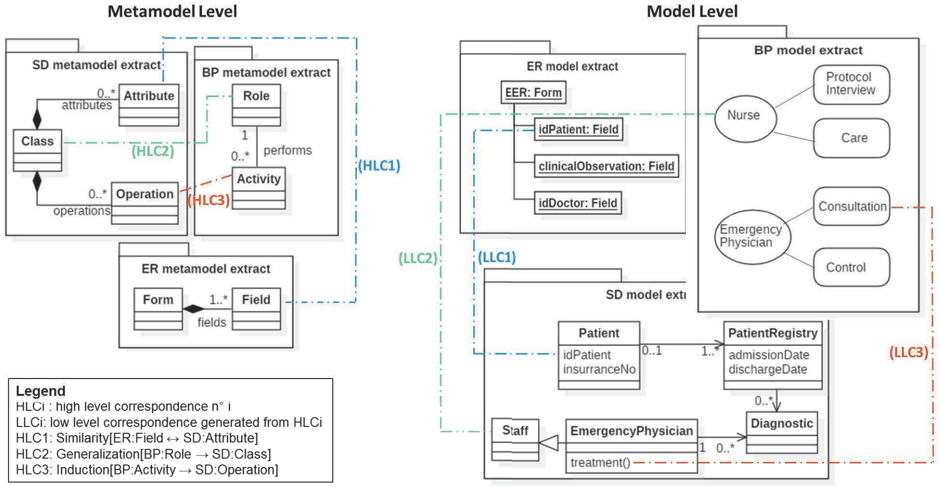

Fig. 4. Extracts of Emergency Department partial models, their respective metamodels and examples of HLCs and their respective LLCs

and the meta-element Attribute from SD metamodel. Likewise, HLC2 means that a Class from SD metamodel can be a generalization of a Role from BP metamodel.

(3) Generate LLCs: Each LLCi is automatically derived from HLCi. In Figure 4, we show an example of valid LLCs. HLC1 generates 12 correspondences, but only LLC1 is valid in regard to the semantics of the Similarity relationship. Thus, HMCS tool will keep only LLC1. In a same manner, LLC2 and LLC3 are kept at the end of the automatic process (for more information about this process, see [10]).

C. Application to ED system

In our model matching process, we have identified two collaborative activities where local coordinators need to elaborate a co-decision: (1) set relationships and (2) produce HLCs. Here, to simplify, we assume that four potential semantic relationships have been set to describe the ED system's correspondences, namely: Similarity, Generalization, Induction, Deduction. Thus, the collaboration we are interested in is the production of HLCs.

SD_{LC}, BP_{LC} and ER_{LC} respectively refer to SD, BP and ER local coordinators. Table II summarises the proposed meta-correspondences, their initiator and decision makers (DMs). A HLC is represented using the following syntax (where → is used for asymmetric relationships and ↔ for symmetric ones):

Relationship "[" metamodel ":" meta-element (→ or ↔) metamodel ":" meta-element "]"

Once HLCs have been proposed, they undergo evaluations by the eligible decision makers (*eligibleDMs*). BP_{LC} is considered to be the collaboration moderator since he is the first actor to initiate a proposal. He chooses to adopt an iterative decision policy. He has thus to choose between *ConsentingTogether* and *NegotiatingTogether*. Let's suppose he opts for the latter. We detail the evaluation process of HLC3 (Figure 5) since HLC1, HLC2, HLC4 and HLC5 are binary (so evaluated by a sole decision maker).

TABLE II
PROPOSED HLCs

N	Initiator	High Level Correspondence	DM(s)
1	BP_{LC}	Similarity [BP:Role ↔ SD:Class]	SD_{LC}
2	SD_{LC}	Similarity[ER:Field ↔ SD:Attribute]	ER_{LC}
3	BP_{LC}	Induction[BP:Activity, SD:Operation → ER:Field]	SD_{LC}, ER_{LC}
4	BP_{LC}	Generalization[BP:Role → SD:Class]	SD_{LC}
5	ER_{LC}	Deduction[ER:Field → SD:Attribute]	SD_{LC}

DM(s): Decision-maker(s)

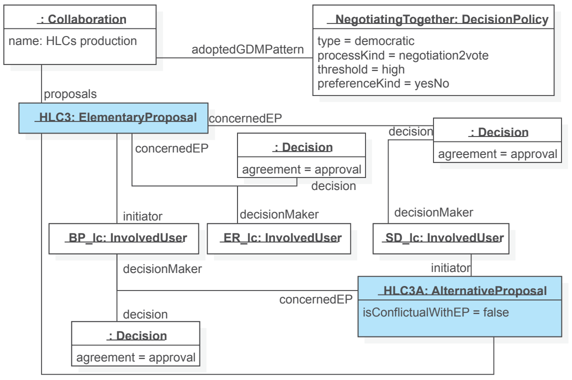

Fig. 5. Instantiation of MMCollab's package Evaluation

HLC3 is initiated by BP_{LC}. It has two decision makers (SD_{LC} and ER_{LC}). ER_{LC} gives his agreement about HLC3 whereas SD_{LC} refines it by an AlternativeProposal (HLC3A): *Induction[BP:Activity SD:Operation]*. SD_{LC} is thus the initiator of HLC3A and BP_{LC} the decision maker. Since HLC3A and HLC3 are not conflictual (stated by SD_{LC} when he defined HLC3A), both HLC3 and HLC3A may be maintained. At the end of this activity, all the proposed

HLCs were approved. Due to space constraints, the model of correspondences (i.e. valid LLCs produced from these HLCs) is not given here, but can be found in [19].

V. TOOL SUPPORT: DECISION MAKING TOOL MODULE

HMCS is a set of modules ensuring matching, consistency management and model transformation. To support the collaborative alignment of models, we added two modules: *Collaboration Tool (CollabT)* and *Decision Making Tool (DMT)*. The global architecture of the collaborative version of HMCS is presented in [10]. In this section we put the focus on DMT module which is dedicated to GDM. DMT module (Figure 6) allows producing a collaborative decision for a given proposal by exploiting users data (*UDB*), implemented decision-making policies (*DMP*), proposals (*PDB*) and their evaluations (*EDB*). These four data storage are accessed by four managers. *UDB extractor* extracts for each proposal (1.a), the list of concerned users (1.b). Then, this list is transferred to *Notification Center* (2.a) that notifies concerned users (2.b). Afterward, users individually assess proposals and provide decisions (3.a) by *Decision Assessment* service. These decisions modify 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).

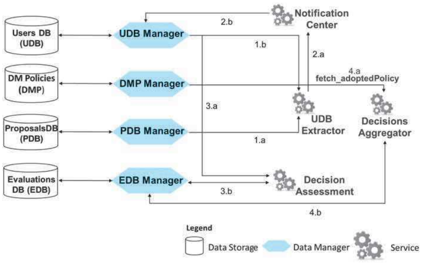

Fig. 6. Decision Making Tool (DMT)

VI. CONCLUSION AND PERSPECTIVES

We have been working on Group Decision Making processes via a conceptual metamodel of collaboration (*MMCollab*). In this paper, we have described the new package *CollectiveDecision* which supports GDM patterns. These latter, once instantiated, give rise to various decision policies that are extensible and customized according to the application context. MMCollab also provides features to organize proposals and allows their evaluation according to several decision criteria. It is also tooled by a Decision Making Tool (DMT). We have applied MMCollab to conduct the collaborative matching process on models of a hospital Emergency Department.

Some work still needs to be done. Indeed, we are finalizing the implementation of the collaborative modules (DMT and CollabT). We also plan to reduce the moderator's intervention. This could be done by (i) defining other GDM patterns and their associated decision policies and (ii) developing a recommendation system to infer the appropriate policies for a given system, by learning experiences from the previous studied systems. Besides, we aim to complete the collaborative alignment process by formalizing the detection and collaborative handling of inconsistencies once the correspondences are set and the partial models evolve.

REFERENCES

[1] M. Broy, M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu, "Seamless model-based development: From isolated tools to integrated model engineering environments," *Proceedings of the IEEE*, vol. 98, no. 4, pp. 526–545, 2010.

[2] P. Shvaiko and J. Euzenat, "Ontology Matching: State of the Art and Future Challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2013.

[3] M. Bräuer and H. Lochmann, "An ontology for software models and its practical implications for semantic web reasoning," in *European Semantic Web Conference*, pp. 34–48, Springer, 2008.

[4] R. Shosha, C. Debruyne, and D. O'Sullivan, "Towards an adaptive tool and method for collaborative ontology mapping," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 319–328, Springer, 2015.

[5] A. V. Zhdanova and P. Shvaiko, "Community-driven ontology matching," in *European Semantic Web Conference*, pp. 34–49, Springer, 2006.

[6] F. R. Golra, A. Beugnard, F. Dagnat, S. Guerin, and C. Guychard, "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), pp. 206–211, ACM, 2016.

[7] M. El Hamlaoui, S. Bennani, M. Nassar, S. Ebersold, and B. Coulette, "A MDE Approach for Heterogeneous Models Consistency," in *International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), Madeira*, SciTePress, 2018.

[8] M. El Hamlaoui, B. Coulette, S. Ebersold, S. Bennani, M. Nassar, A. Anwar, A. Beugnard, J.-C. Bach, Y. Jamoussi, and H. N. Tran, "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), pp. 18–27, 2016.

[9] H. Brunelière, J. G. Perez, M. Wimmer, and J. Cabot, "Emf views: A view mechanism for integrating heterogeneous models," in *International Conference on Conceptual Modeling*, pp. 317–325, Springer, 2015.

[10] S. Bennani, M. El Hamlaoui, S. Ebersold, B. Coulette, and M. Nassar, "Collaborative model-based matching of heterogeneous models," in *International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanjing, China*, IEEExplore, 2018.

[11] D. Baker, D. Bridges, R. Hunter, G. Johnson, J. Krupa, J. Murphy, and K. Sorenson, "Guidebook to decision-making methods. developed for the department of energy," tech. rep., WSRC-IM-2002-00002, 2001.

[12] I. P. Levin, M. E. Huneke, and J. D. Jasper, "Information processing at successive stages of decision making: Need for cognition and inclusion–exclusion effects," *Organizational behavior and human decision processes*, vol. 82, no. 2, pp. 171–193, 2000.

[13] J. L. C. Izquierdo and J. Cabot, "Collaboro: A collaborative (meta) modeling tool," *PeerJ Computer Science*, vol. 2, p. e84, 2016.

[14] J. Chai and J. N. Liu, "An ontology-driven framework for supporting complex decision process," in *World Automation Congress (WAC), 2010*, pp. 1–6, IEEE, 2010.

[15] E. Kornyshova and R. Deneckère, "Decision-making ontology for information system engineering," in *International Conference on Conceptual Modeling*, pp. 104–117, Springer, 2010.

[16] J. A. Rockwell, I. R. Grosse, S. Krishnamurty, and J. C. Wileden, "A decision support ontology for collaborative decision making in engineering design," 2009.

[17] OMG, "Software & systems process engineering meta-model specification," *OMG Std., Rev*, vol. 2, 2008.

[18] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design patterns: Elements of reusable object-oriented software addison-wesley," *Reading, MA*, p. 1995, 1995.

[19] M. El Hamlaoui et al., "Emergency Department case study." https://github.com/sawa91/Wetice/blob/master/ED-case-study.pdf, 2019. [Online; accessed 28-February-2019].