Collaborative process cartography deduction based on collaborative ontology and model transformation

Wenxin Mu, Frederick Benaben, Hervé Pingaud

HAL Id: hal-01596362

https://hal.science/hal-01596362v1

Submitted on 26 Nov 2018

Collaborative process cartography deduction based on collaborative ontology and model transformation

Wenxin Mu [a,*], Frédérick Bénaben [b], Hervé Pingaud [c]

[a] *School of Economics and Management, Beijing Jiaotong University, 3 Changyuancun Xizhimenwai Beijing 100044, China*
[b] *Industry engineering center, Toulouse University – Ecole des mines d'Albi-Carmaux, Campus Jarlard, Allée des sciences, Albi, France*
[c] *Champollion University, Place Verdun, Albi, France*

ABSTRACT

With an increasing background in inter-enterprise collaboration and interoperability, the automatic deduction of collaborative business processes is increasingly being viewed as a crucial research subject. The common solution is believed to involve either semantic ontologies or metamodeling, which can be combined with semantic algorithms or transformation rules. However, from the viewpoint of enterprises, the modeling of collaborative processes should be software integrated and can be changed to workflow. The design of the targeted collaborative process model does not fit the need. This has led to a lack of concepts and transformation rules for the ontology or metamodel. In this paper, a new collaborative process model called collaborative process cartography is designed. Related collaborative ontology and its knowledge-expanding rules have also been updated. Collaborative ontology contains essential concepts for the collaborative situation and includes the rules and algorithms for process deduction. A prototype for the supporting tools is also provided.

1. Introduction

The ability of organizations (enterprises or institutions) to operate in collaboration is a key indicator of their level of competitiveness. However, organizations, whether recent or well established, rarely take considerable interest in their collaborative capacities (which are mainly empirical abilities that are informally built and used as needed). We propose to study the specificities and requirements of organizational collaboration from an IS (Information Systems) point of view and to define an engineering approach dedicated to facilitating the deployment of these emerging collaborative networks.

Enterprise interoperability can be seen as the capacity of enterprises or organizations to structure, formalize, and present their knowledge and to exchange or share it. Guedria et al. [14] introduce numerous methods, tools and languages, which have been developed to enhance enterprise interoperability. In Nicolle et al. [31], various architectures for the interoperation of information systems are introduced and compared, including the Peer-to-Peer [30,47], Standardization[1], Federation[2], Multi-base[3], Ontology [13,16] and Mediation [44] methods. Considering the difficulty in building common standards and initializing a collaborative environment, the combination of ontology and mediation is the most suitable way to solve interoperability problems.

---

* Corresponding author. Tel.: +8651683673; fax: +8651684925.
E-mail address: wxmu@bjtu.edu.cn, olinamu@126.com (W. Mu).
[1] Standardization uses a pivot model, canonical model or meta-model to reduce the number of translators (similar to the Peer-to-Peer method).
[2] Federation is derived from the standardization and uses a global, static federated schema.
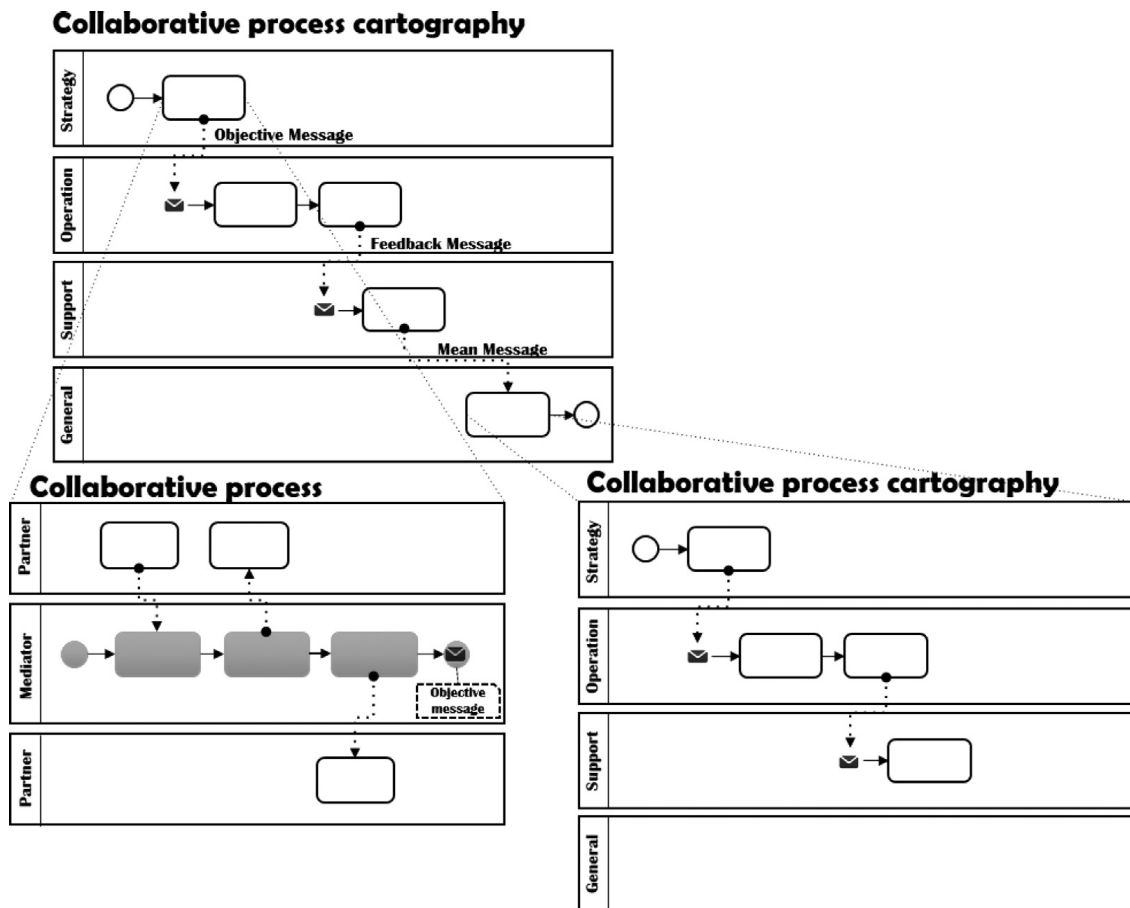[3] Multi-base uses a single language for many ISs.

Fig. 1. An example of collaborative process cartography.

Achieving sustainable interoperability among organizations in a networked environment is a crucial factor for successfully managing collaborations at all levels: abstract (business) and concrete (technology) [20]. In the field of IS (Information Systems) interoperability, the research topics and results are focused on either the business process or the technical workflow. This has led to the development of a huge gap between the business level and the technical level. Therefore, our research is focused on filling this gap by transferring business knowledge to a collaborative process.

To address the difficult task of constructing an automatically collaborative business process, our lab launched the Mediation Information System Engineering (MISE) project (2003-2014). Rajsiri et al. [39] defined the business level of MISE 1.0 (Mediation Information System Engineering version 1.0) by using the Model-driven Architecture (MDA) [29] and the Service Oriented Architecture (SOA) [23,42]. They defined collaborative ontology for gathering collaborative information and transferred this information to a collaborative business process. However, according to ISO 9000 [19], the business process is classified as a strategy, operation and support process. One single but complex collaborative process in MISE 1.0 leads to a problem: the collaborative process is difficult for partners to understand in different roles, e.g., managers, workers, and operators.

To solve this problem, a cartography of collaborative processes is required to classify the entire collaborative process into strategy, operation or support collaborative processes (as shown in Fig. 1). In 2009, our lab launched MISE 2.0 to generate the cartography of collaborative processes automatically. Fig. 1 shows an example of collaborative process cartography, which is based on BPMN (Business Process Modeling Notation) [34,35,37]. The collaborative process cartography first defined the collaborative process by its main functions, which are positioned in different pools (strategy, operation, support and general). Each main function is specified by another collaborative process cartography or a collaborative process, which depends on the complexity of the main function.

To deduce the collaborative process cartography, the following research questions should be answered:

- How should a collaborative ontology be defined? In ontology, all concepts involved in the collaborative situation, collaborative process, and process cartography should be included.
- What is the best way to search or transfer the useful process entities to fill the collaborative ontology?
- How should the deduction rules to be used to obtain the collaborative process cartography that will be defined?

In this paper, Section 2 gives the introduction of related works. Section 3 presents the global view of this research work and our working definition of collaborative ontology. Section 4 provides the transformation rules from gathered knowledge to deduced knowledge in the collaborative ontology. Section 5 presents the deduction rules used to deduce the cartography of collaborative

processes from collaborative ontology. Section 6 is a sketch of the knowledge transforming results and the implementation of software tools, and it includes a case study. Section 7 gives the evaluation results. Section 8 provides the comparisons with related works.

2. Related works

A BPM (Business Process Management) life cycle includes six steps: design, modeling, execution, monitoring, optimization and re-engineering [17]. Patig et al. [36] have conducted a worldwide survey of major public companies to elicit the requirements, which are grounded in the nature of the processes and the software usage. The analysis of 127 responses indicates that BPMs with software integration capabilities are the most urgently required. Our work is focused on "modeling", but with a special word ("automatic deduction") for solving the gap between process modeling and software integration capabilities, we are interested in determining how the software should be used to deduct the cartography of collaborative business processes.

To analyze the related works, we searched the web of science with the key words "business process" and "collaboration". We found 312 articles from 2013 to 2015. We narrowed these down to 14 strongly related articles, which were mainly published by journals under the information systems category, e.g., "Information Sciences", "Information Systems", "Enterprise Information Systems" and "IEEE Transactions on Systems Man Cybernetics-Systems".

Comuzzi et al. [9] propose an approach to flexibly design and enact cross-organizational business process monitoring based on a Product-Based Workflow Design. Antonia Martinez-Carreras et al. [1] describe the building of a context-aware application integrated in a Collaborative Working Environment (CWE) architecture, and they compared this new augmented application to existing applications. Bae et al. [2] introduce a methodology for incorporating business process semantics and alternative paths in the Business Process Management structure. Nie et al. [32] base their work on exchanged messages among processes. They consider behavior mismatch scenarios as the main feature of the message to identify occurrences in the mismatch patterns, leading to a corresponding process. Bianchini et al. [4] provide a detailed description of process-to-service, which is a computer-aided methodology used to enable the identification of services that create a collaborative business process. Norta et al. [33] define the reference architecture as a division of the functionality, with the exchanged data flow mapped onto the software elements that implement the required functionality for a specific context and the business goals of the stakeholders. Hoisl et al. [18] present an integrated model-driven approach for the specification and the enforcement of secure object flows in process-driven service-oriented architectures (SOA). A secure object flow ensures the confidentiality and integrity of important objects that are passed between difference participants in a SOA-based business process. Liptchinsky et al. [24] consider a collaboration process as the evolution of a network of collaborative documents along with a social network of collaborators. Yongchareon et al. [45] present a framework for modeling inter-organizational business processes and validating the changes to them. They develop algorithms to find the minimal public view for a given inter-organizational business process. Zeng et al. [46] built a Petri-net based RM_MF_Net model to provide systematic modeling and verification for the cross-department processes when considering different types of coordination patterns among different departments. Cortes-Cornax et al. [10] link the local and global perspectives of the inter-organizational business processes defined in BPMN2.0. They maintain a separation of concerns between the intentional level captured by the goal model and the organizational level captured by the process model. Rico et al. [40] propose to define the semantics for Business Process Management systems interoperability through the ontology of Electronic Business Documents (EBD) used to interchange the information required to perform cross-organizational processes. Li et al. [25] use SOA-based collaborative models and collaborative agents to solve problems with collaborative management for service convergence under the condition of process and data fragmentation. Meyer et al. [28] describe an entirely model-driven approach for BPMN by introducing a few concepts that model data retrieval, data transformation, message exchange, which are the four aspects of data exchange.

First, as shown in Table 1, the research topics all involve modeling or execution, but none of the papers combine both. Second, the research works design the methods from either message or function. Only one work exists that focuses on collaborative networks. Third, the future works can be summarized as follows: concept expansion, process abstraction, research method improvement and process verification.

To summarize, the research work in this paper helps to:

- Fill the gap between the business process and the technical workflow in the research field of IS interoperability by building a bridge from process modeling to workflow execution.
- Expand the collaborative concepts from a collaborative network viewpoint, including a Mediator, Partner, and Objective.
- Develop SaaS-based software tools to enhance the software integration capabilities of process modeling.
- The research results presented in the following sections provide the field of management and software engineering with relevant contributions in terms of mediation information system engineering.

3. Preliminaries

The main framework of the method presented in this article is shown in Fig. 2. This method is entirely based on collaborative ontology (Fig. 3).

The ontology is separated into three parts: the initial condition, function and message, and mediator concepts. The knowledge from the initial condition is gathered using the collaborative network model. The "function and message" portion of this process

Table 1
Summaries of related works.

Reference	Topic	Focus	Methods	Tool	Future work
Comuzzi et al. [9]	E	F*	Ant-colony optimization	ProM 5.2	Extend the type of monitoring functions
Antonia et al. [1]	M*	F*	Ontology* First order logic*	Protégé*, COCOS	Expand ontology with agent
Bae et al. [2]	E	F*	Genetic algorithm	-	Develop hybrid genetic algorithms
Nie et al. [32]	M*	M	Behavior mismatch patterns	BPMN*	Message semantics
Bianchinii et al. [4]	M*	F*	Service identification metrics	BPMN*	Business process abstraction
Norta et al. [33]	M*	M	E-sourcing reference architecture	External-internal modeling*	Collaboration concepts of trust management
Hoisl et al. [2014]	E	M	Enforcing secure object flows, MDA*	SoaML, BPMN*, UML	Extend business activities framework
Liptchinsky et al. [24]	M*	N*	Context dependency First order logic*	Petri net	Extend with the notion of group
Yongchareon et al. [45]	M*	F*	Artifact-centric meta-model	Petri net	Web services specification and abstraction
Zeng et al. [46]	M*	F*	RM_WF_NET modeling, cross-department coordination patterns	Petri net	Verification of the entire cross-department medical process
Cortes-Cornax et al. [10]	M*	F*	Intentional fragments	BPMN*	Develop automated support for verifying completeness
Rico et al. [40]	M*	F*	Method for ontology learning techniques	Ontology of Electronic Business Documents*	-
Li et al. [25]	E	F*, N*	Collaboration agents	ENI-P	Multi-cloud integration
Meyer et al. [28]	M*	M	BPMN and model-driven data dependency enactment approach	BPMN and UML based data model	Formal verification technique in process orchestrations and choreographies

Topic E: execution, M: modelling.
Focus F: function, M: message, N: network.
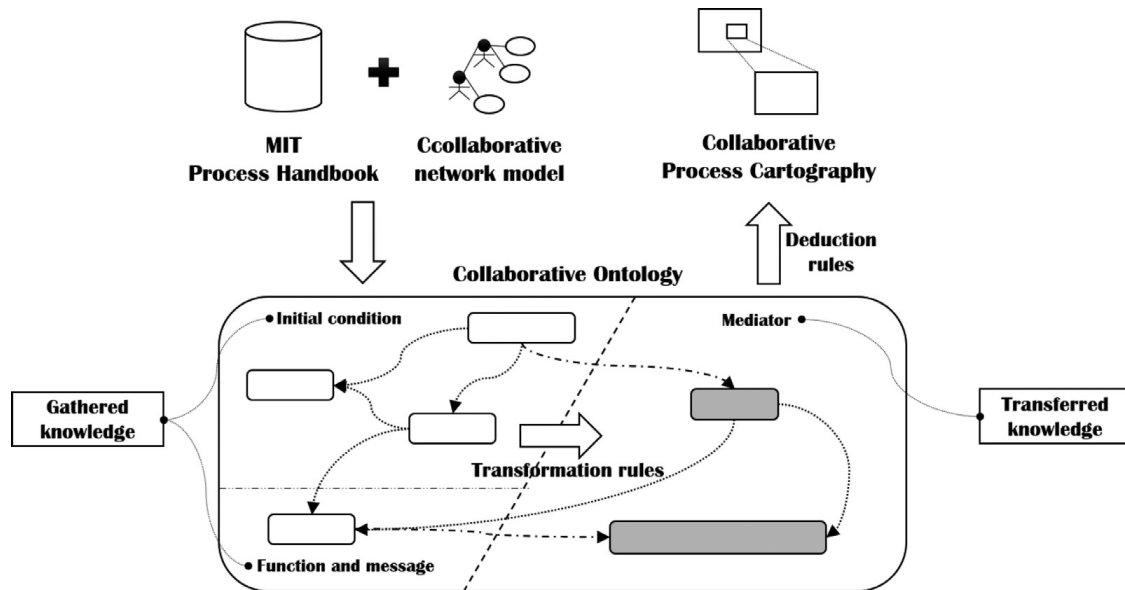*related or similar part to this paper.

Fig. 2. Knowledge in/out of collaborative ontology.

was derived from the MIT process handbook [27]. Both steps involve gathering knowledge. The transformation rules were defined to transfer the gathered knowledge to mediator concepts (transferred knowledge). Finally, by using the sum of the knowledge from the collaborative ontology, a model called collaborative process cartography is extracted and presented to the user.

3.1. Collaborative ontology – initial condition

From a collaborative business view, a collaborative situation is structured into five parts: the initial conditions; environmental factors; process, structure and governance; contingencies and constraints; and outcomes and accountabilities [7]. For the concept definition of the collaborative situation, the most important parts were the initial conditions and the process. According to our
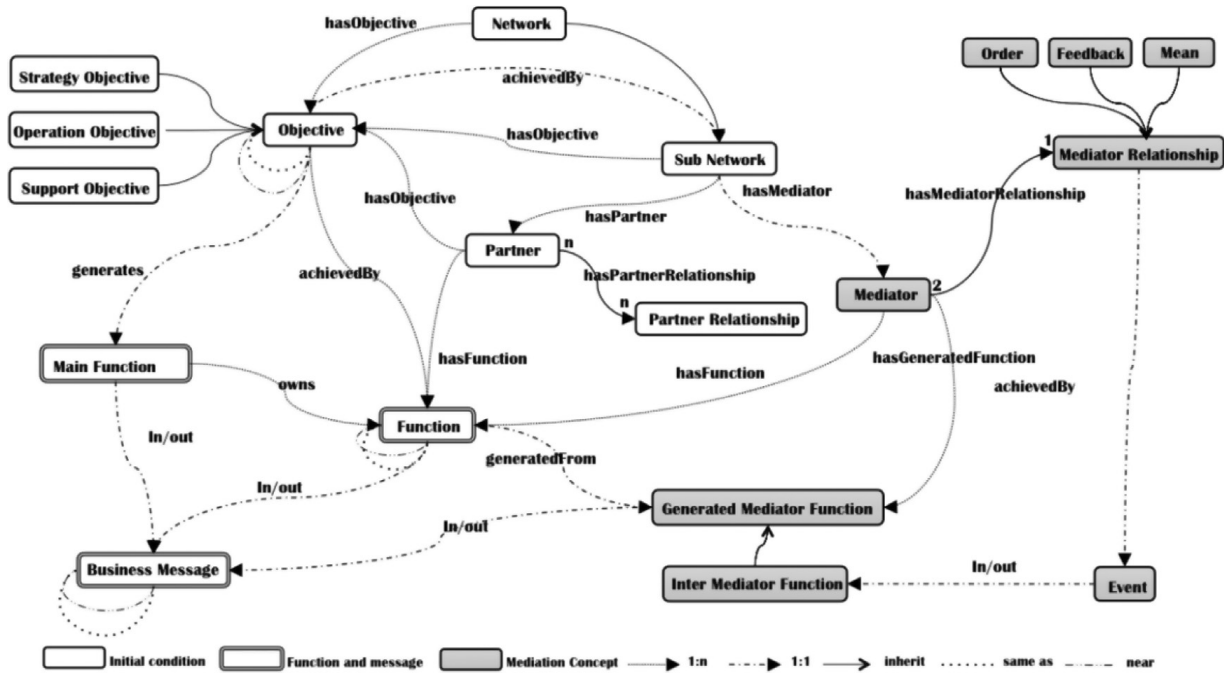
Fig. 3. Collaborative ontology of MISE 2.0.

experiment with the research projects (ISTA3[4], IsyCri[5], and OpenPaas[6] for the initial conditions), the key concepts for modeling include the objective, the collaborative network, and the partner. To demonstrate this idea, we assessed the state of art processes in the field of collaborative business and collaborative organization modeling.

As Dormady [12] mentioned, "organizations and firms in all sectors have long used partnerships, subsidiary corporations, divisions, or departments to adapt to the organizational environment in a manner consistent with the goals and objectives of the organization". Bouslimi et al. [6] define the collaborative network according to three levels of abstraction: the role model, organizational structures and concrete organizations. Rajsiri et al. [39] define the collaborative situation with the following concepts: network, participant, abstract service, topology, role and common goal. Jiang et al. [21] define an inter-organizational collaborative model with only two fundamental concepts: Role and Organization, which consider the role that encompasses a set of objectives.

For the above work, the objective and network are key concepts that are modeled on the same surface, which means that one objective is completed by one network, and thus, one network achieves several objectives. However, from our viewpoint, the relationship between the objective and network is stereoscopic, as shown in Fig. 4. In level 0, all partners are considered as the whole collaborative network. Then, they are defined as the objective of the whole network. In level 1, each objective in level 0 is expanded by a sub-network or by partners within the collaborative objectives. In level 2, the objective is expanded again by a sub-sub network with partners and their own objectives. The whole network is separated into groups (sub-networks) by collaborative objectives. This type of decomposition can continue to level n until the partners are the leaves of the tree.

In the collaborative ontology, the initial condition (Fig. 4) of the collaborative ontology is defined by the *network*, *objective*, *subnetwork*, *partner* and *partner relationship*. A *network* can have several *objectives* (*strategy*, *operation* or *support*). An *objective* can be obtained from the *sub-network*. A *sub-network* contains several *partners*. Among the *partners*, there are *partner relationships*.

3.2. Collaborative ontology – function and message

The very first study to specifically assess the Process Handbook project began in 1991 [27]. The project has been one of the primary projects of the MIT Center for Coordination Science. According to Bernstein et al. [3], the project includes: (i) collecting examples of how different organizations perform similar processes and (ii) representing these examples in an on-line "Process Handbook" that includes the relative advantages of the alternatives. To represent the large number of processes, the MIT process handbook ontology is defined. The Process Handbook ontology provides a specialized hierarchy of the processes and their interrelationships in the form of properties. All major parts of the Process Handbook, such as the Process, Bundle, Goal, Exception, Resource, Dependency, and Trade-offs, are represented as OWL classes. Rajsiri [38] analyzes the hierarchy of the processes in the
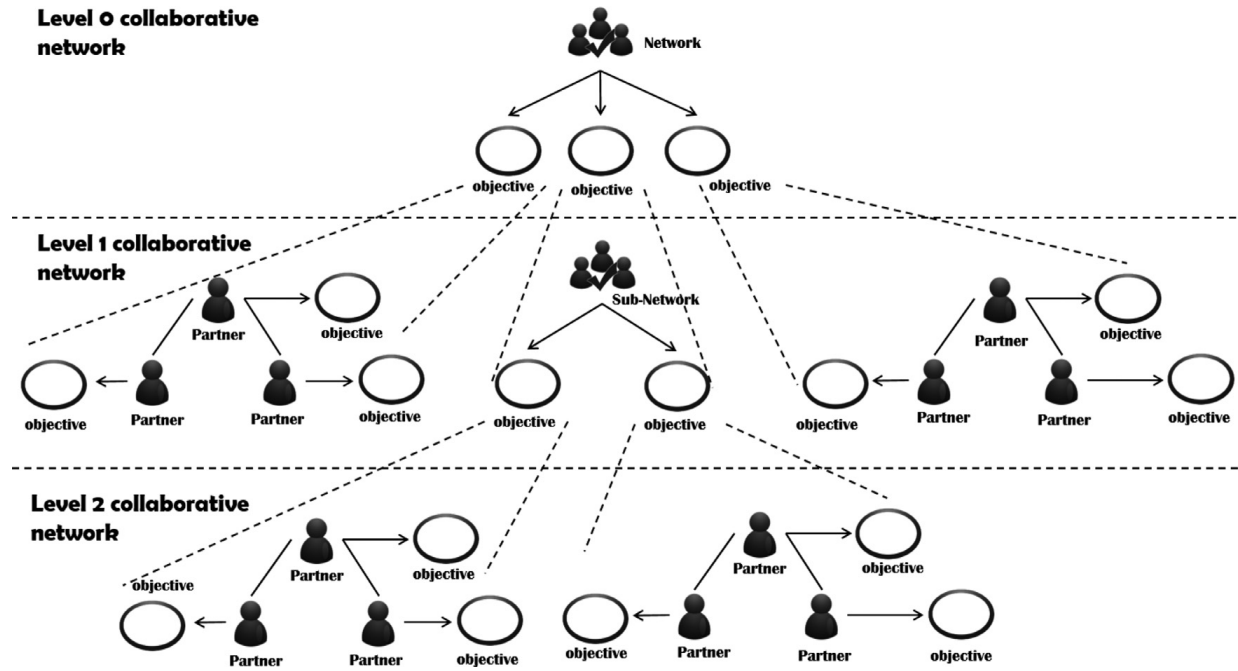
Fig. 4. Our vision of the collaborative networks and objectives.

ontology of process handbook to build the "function and message" part of the collaborative ontology of MISE 1.0. Here, we re-use her work to build our part of "function and message".

3.3. Collaborative ontology - process

This part is completely new for the collaborative ontology. It is different from the collaborative ontology presented in MISE 1.0, and the major difference is the introduction of the Mediator role.

- The *Mediator* is a participant in the collaboration and is a type of *Partner*. Both are actors in the collaboration, but the *Mediator* manages and orchestrates the collaborative process. From the viewpoint of the actors, the *Mediator* is at the center of all of the *Partners*.
- The *Mediator Relationship* defines the relationships among the *Mediators*. These include the objective, feedback and mean messages, which are transferred among different types of collaborative processes. Each collaborative process has one *Mediator*. Thus, the relationships among the *Mediators* are classified according to their *Order*, *Feedback* and *Mean*.
- The *Generated Mediator Function* is the function or business service of the *Mediator* because the *Mediator* orchestrates the functions of the partners. The functions of the *Mediator* can be generated from the functions of the partners. Thus, the functions of the *Mediator* are called *Generated Mediator Function*.
- The *Inter Mediator Function* is a special type of *Generated Mediator Function*. If one *Generated Mediator Function* sends or receives *Objective*, *Feedback* or *Mean* messages, the *Generated Mediator Function* is an *Inter Mediator Function*.
- The *Event* respects the *Mediator Relationship*. If the *Mediator Relationship* is in the abstract level of ontology, then the *Event* is in the concrete level of the ontology.

4. Transformation rules

The transformation rules of the collaborative ontology are defined in the first-order logic [41]. The transformation rules of collaborative ontology are specified in five groups (Table 2). These transformation rules aim to transfer the "initial condition" and "function and message" to "mediator" concepts.

Due to the particularity of the transformation rules, the first-order logic should still be expanded as follows:

- Class: X is instance of a *collaborative network* → collaborative network(X)
- Association: Y is an instance of an association *implement* that is between the *collaborative network* X1 and *objective* X2 → implement(Y) (collaborative network(X1), objective(X2))
- A set of instances: from X1, X2, X3 to Xn → X1 … Xn

The five groups of transformation rules address the deduction of the mediation concepts. Group 1, 2 and 3 create the *Mediator*, *Mediator Relationship* and *Generated Mediator Function*. Group 4 creates the relation *hasGeneratedFunction* to link the *Mediator* with the deduced *Generated Mediator Function*. Group 5 transfers the *Mediator Relationship* to the Event. With the *Event*, group 5 can recognize the *Inter Mediator Function* from the *Generated Mediator Function*.

Table 2
Transformation rules.

Group no.	Group name	Description	Rule	Total No.
1	Create *Mediator*	Deduce *Mediator* from *Sub Network*	(1)	1
2	Create *Mediator Relationship*	Deduce *Mediator Relationship* through *Sub Network*, *Objective*, *Main Function* and *Business Messages*	(2), (3) and (4)	3
3	Create *generated Mediator Function*	Deduce *Generated Mediator Function* and *In/out* relation through *Function*	(5) and (6)	2
4	Link *generated Mediator Function* to *mediator*	Deduce the relation: *hasGeneratedFunction* through *Mediator*, *Objective*, *Main Function* and *Function*. This group is based on the following relation: *owns* exists.	(7)	1
5	Create *inter Mediator Function*	Deduce *Event* from *Mediator Relationship*, Separate *Inter Mediator Function* from *Generated Mediator Function* and link it to *Event*	(8)	1

Table 3
Group 1: creating the mediator.

$\forall$Sub Network (X) ($\forall$hasPartner (Sub Network (X)Partner (X_1) $\wedge$ ($\forall$hasPartner (Sub Network (X), Partner (X_2))$\wedge$...$\wedge$ ($\forall$hasPartner (Sub Network (X), Partner (X_n)))	Rule(1)
-> $\exists$Mediator (X) $\wedge\exists$hasMediator (Sub Network (X), Mediator (X))	

Table 4
Group 2: Creating the mediator relationship.

$\forall$Strategy Objective (X_1) ($\forall$generates (Strategy Objective (X_1), Main Function (X_1))) $\wedge$ $\forall$Operation/Support Objective (X_2) ($\forall$generates (Operation/Support Objective (X_2), Main Function (X_2))) $\forall$Main Function (X_1) ($\forall$out (Main Function(X_1), Business Message (m))) $\wedge$ $\forall$Main Function (X_2) ($\forall$in (Main Function(X_2), Business Message (m))) -> $\exists$ Order (m)(hasMediatorRelationship (Mediator (X_1), Order (m))) $\wedge$ $\exists$ Order (m)(hasMediatorRelationship (Mediator (X_2), Order (m)))	Rule (2)
$\forall$Operation Objective (X1) ($\forall$generates (Operation Objective(X1), Main Function(X1))) $\wedge$ $\forall$Strategy/Support Objective (X2) ($\forall$generates (Strategy/Support Objective(X2), Main Function(X2))) $\forall$Main Function (X1) ($\forall$out (Main Function(X1), Business Message (m))) $\wedge$ $\forall$Main Function (X2) ($\forall$in (Main Function(X2), Business Message (m))) -> $\exists$ Feedback (m)(hasMediatorRelationship (Mediator (X1), Feedback (m))) $\wedge$ $\exists$ Feedback (m)(hasMediatorRelationship (Mediator (X2), Feedback (m)))	Rule (3)
$\forall$Support Objective (X1) ($\forall$generates (Support Objective(X1), Main Function(X1))) $\wedge$ $\forall$Strategy/Operation Objective (X2) ($\forall$generates (Strategy/Operation Objective(X2), Main Function(X2))) $\forall$Main Function (X1) ($\forall$out (Main Function(X1), Business Message (m))) $\wedge$ $\forall$Main Function (X2) ($\forall$in (Main Function(X2), Business Message (m))) -> $\exists$Mean (m)(hasMediatorRelationship (Mediator (X1), Mean (m))) $\wedge$ $\exists$ Mean (m)(hasMediatorRelationship (Mediator (X2), Mean (m)))	Rule (4)

4.1. Group 1: creating the mediator

Group 1 addresses the creation of instances of the *Mediator*. From a business perspective, the mediator orchestrates the collaborative process. The mediator has an important role in the collaborative network. The mediator manages the collaborative network to achieve the collaborative objective.

If one *Sub Network* contains several *Partners*, then for the *Sub Network*, there is one *Mediator* to manage the collaboration of the *Sub Network* (Rule 1, shown in Table 3). This group of transformation rules contains one equation. Table 3 presents rule (1) in extended first-order logic. The rule states that if there is an instance X of *Sub Network* and X links to several instances of *Partner* ($X_1, X_2 ... X_n$) by the relation *hasPartner*, then there is an instance X of *Mediator* that links to the instance of *Sub Network* by the relation *hasMediator*.

4.2. Creating the mediator relationship

This group concerns the deduction of instances of *Mediator Relationships* by recognizing the in/out of the business message of the main function. From the business perspective, in one collaborative situation, there may be several mediators that manage different collaborative processes and achieve different collaborative objectives. However, among different collaborative processes, there are communications. *Mediator Relationships* aim to communicate between the *mediators*.

Group 2 has three rules (rules 2, 3 and 4 in Table 4). In the collaborative ontology, the *Mediator Relationship* is defined to represent the communications among the collaborative processes. There are three types of *Mediator Relationships*. First, *Order* represents the objective information. *Feedback* represents the feedback information. *Mean* represents the mean information.

Table 5
Group 3: Creating the generated mediator function.

∀Function (X) (∀In (Function (X), Business Message (m))) -> ∃ Generated Mediator Function (X) (Out (Generated Mediator Function (X), Business Message (m))) ∧ ∃ Generated Mediator Function (X) (generatedFrom (Function (X), Generated Mediator Function (X)))	Rule (5)
∀Function (X) (∀In (Function (X), Business Message (m))) -> ∃ Generated Mediator Function (X) (Out (Generated Mediator Function (X), Business Message (m))) ∧ ∃ Generated Mediator Function (X) (generatedFrom (Function (X), Generated Mediator Function (X)))	Rule (6)

Table 6
Group 4: Linking the generated mediator function to the mediator.

∀Main Function (X) (∀owns (Main Function (X), Function (X_1)))	Rule (7)
-> ∃ Mediator (X) (∀hasGeneratedFunction (Mediator (X), Generated Mediator Function (X_1)))	

Rule (2) addresses the deduction of the *Order*. Rule (3) addresses the deduction of the *Feedback*. Rule (4) addresses the deduction of the *Mean*.

Rule (2) starts from the *Strategy* and *Operation Objective*. If an instance X_1 of the *Main Function* is transferred from an instance of the *Strategy Objective*, if an instance X_2 of the *Main Function* is transferred from an instance of the *Operation Objective*, and if an instance m of *Business Message* exists such that m is the output of X_1 and m is also the input of X_2, then an instance m of the *Order* is deduced and the relationship *hasMediatorRelationship* with *Mediator (X_1)* (this is the instance of *Mediator* that was generated from the *Sub Network* to achieve the *Main Function (X_1)*) and *Mediator (X_2)* (this is the instance of the *Mediator* that was generated from the *Sub Network* to achieve the *Main Function (X_2)*) is deduced. Rule (4) creates *Order* between the *Mediators*, which are transferred from the *Strategy* and *Support Objective*. Rule (5) creates *Feedback* between the *Mediators*, which are transferred from the *Operation* and the *Strategy Objective*. Rule (3) creates *Feedback* between the *Mediators*, which are transferred from the *Operation* and *Support Objective*. Rule (4) creates the *Mean* between the *Mediators*, which are transferred from the *Support* and *Strategy Objective*. Rule (8) creates the *Mean* between the *Mediators*, which are transferred from the *Support* and *Operation Objective*.

4.3. Group 3: creating the generated mediator function

The rules in this group are designed to deduce the instances of the *Generated Mediator Function* and *input/output* relations. The mediator needs the mediator functions to invoke the partner functions. The input and output messages between the mediator functions and the partner functions must be added.

This group contains two rules (Rules 5 and 6 in Table 5). Rule (5) defines how the *Generated Mediator Function* and *Out* relationship are created. Rule (6) defines how the *Generated Mediator Function* and the *In* relationship are created.

For rule (5), if Function (X) has the input Business Message (m), then the Generated Mediator *Function (X)* and the relationship *generatedFrom* between *Function (X)* and *Generated Mediator Function (X)* are created, and the *Generated Mediator Function (X)* has the output *Business Message (m)*. For rule (6), if *Function (X)* has the output *Business Message (m)*, then the *Generated Mediator Function (X)* and relationship *generatedFrom* between *Function (X)* and *Generated Mediator Function (X)* are created. The *Generated Mediator Function (X)* has the input *Business Message (m)*.

4.4. Group 4: linking the generated mediator function to mediator

This group is dedicated to deducing the instances of the relation: *hasGeneratedFunction* between the *Generated Mediator Function* (already transferred by group 3) and the *Mediator*. To deduce the collaborative process, we must know that the mediator functions belong to the mediator. The mediator can use its functions to invoke partner functions.

Rule (7) in Table 6 defines the deduction method. If *Main Function (X)* has *Function (X_1)*, then the *Mediator (X)*, which manages the *Sub Network* of *Main Function (X)*, has the *Generated Mediator Function (X_1)*, which is generated from *Function (X_1)*.

4.5. Group 5: creating the inter mediator function

This fifth group concerns the deductions of the instances of the *Inter Mediator Function*. As explained for group 2, one collaborative situation may have several mediators. There are communications among the mediators. This group helps to identify the specific mediator function, which is communicated with the other mediators. (Table 7)
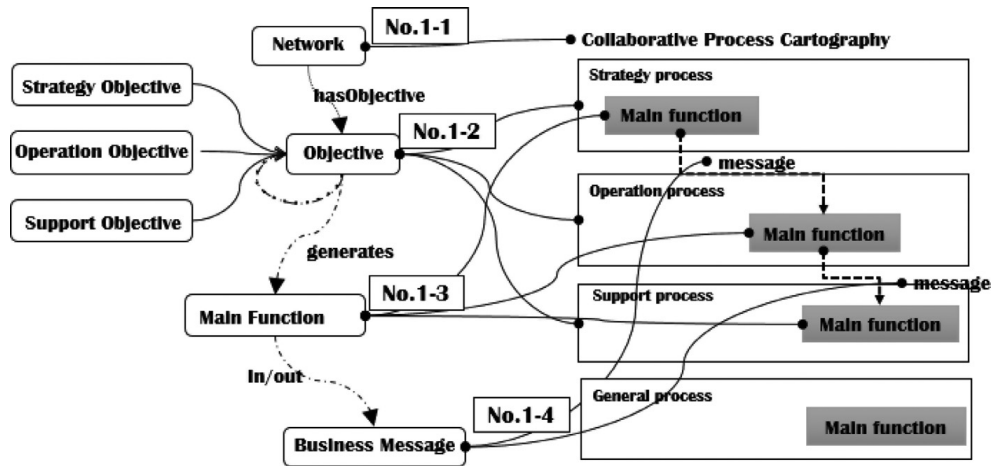
If the *Generated Mediator Function (X)* of *Mediator (Y)* has the *Business Message (m)* that generates *Mediator Relationship (m)* of *Mediator (Y)* by rule (2) of group 2, then the *Event (m)* is created to achieve the *Mediator Relationship (m)*. The *Generated Mediator*

Table 7
Group 5: creating the inter mediator function.

∀Generated Mediator Function (X)   Rule (8)
(In/out (Generated Mediator Function (X), Business Message (m))) ∧
∃ Mediator Relationship (m)
(hasMediatorRelationship (Mediator (Y), Mediator Relationship (m))) ∧
Mediator (Y) (hasGeneratedFunction (Mediator (Y), Generated Mediator Function (X)))
-> ∃ Event (m) (achievedBy (Mediator Relationship (m), Event (m))) ∧
∃ Inter Mediator Function (X) (In/out (Event (m), Inter Mediator Function (X)))

Table 8
Deduction rules.

Group no.	Group name	Description	Rule	Total no.
1	Deduction of the cartography	Deduce the cartography of the collaborative process	1–1, 1–2, 1–3 and 1–4	4
2	Deduction of the process	Deduce the collaborative process of each main task in the process cartography	2–1, 2–2 and 2–3	3
3	Deduction of the partner pool	Deduce the partner pool of the collaborative process	3–1 and 3–2	2
4	Deduction of the mediator pool	Deduce the mediator pool of the collaborative process	4–1, 4–2 and 4–3	3

Fig. 5. Deduction of the collaborative process cartography.

Function (X) is also an *Inter Mediator Function (X)*. If the Business *Message (m)* is the input message of the *Generated Mediator Function (X)*, then the *Event (m)* is also the input of the *Inter Mediator Function (X)*. If the *Business Message (m)* is the output message of the *Generated Mediator Function (X)*, then the *Event (m)* is also the output of the *Inter Mediator Function (X)*.

5. Deduction rules

The deduction rules of the collaborative process have been separated into four groups of rules. These four groups of rules are summarized in Table 8.

Part 1: deduction of the process cartography

The deduction of the process cartography demonstrates how the collaborative process cartography is extracted from the collaborative ontology. Fig. 5 shows an example of the deduction of the process of cartography. There are four rules in this part of the deduction.

- No.1–1: *Network/Sub Network* → Collaborative process cartography. If the *Network* or *Sub Network* has collaborative objectives, then a correspondence collaborative cartography exists for the *Network* or *Sub Network*.
- No.1–2: *Objective* → Strategy/Operation/Support/General pool. Based on rule No.1–1, rule No.1–2 fills the collaborative process cartography with different types of pools (strategy, operation, support and general). If the *Network* has a strategy objective, then there is one strategy pool. If the *Network* has an operation objective, then there is one operation pool. If the *Network* has a support objective, then there is one support pool. If the *Network* has an objective, then there is one general pool.
- No.1–3: *Main Function* → Task
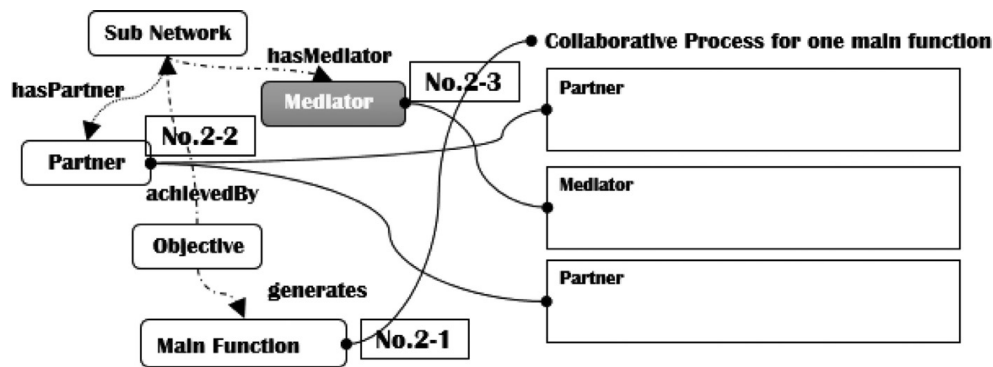- No.1–4: *Business Message* → Message flow

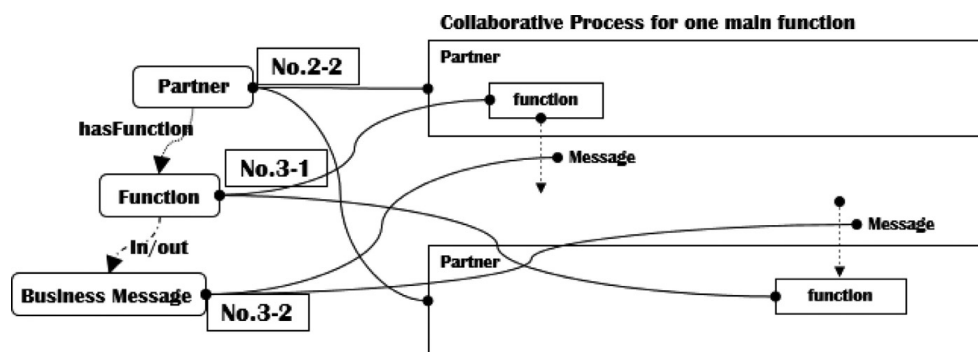**Fig. 6.** Deduction of the collaborative process.

Fig. 7. Deduction of the partner pool.

Part 2: deduction of the collaborative process

Part 2 defines the rules of deduction for the collaborative process. In the process cartography deduced as explained in Part 1, there are main tasks. For each task, there is a collaborative process. Part 2 shows how the sub-collaborative process is transferred. Fig. 6 shows an example of the deduction for the collaborative process. There are three rules in this part of the deduction.

- No.2–1: *Main Function* → Collaborative process. If there is a *Main Function* that has been transferred to task in the strategy/operation/support pool, then there is a detailed collaborative process. However, there is a special case if there is a *Main Function* that has been transferred to a task in the general pool. Then, there is a detailed cartography of the collaborative process, and the deduction goes back to part 1.
- No.2–2: *Partner* → Partner pool. Based on rule No.2–1, rule No.2–2 fills the collaborative process with partner pools. If the *Main Function* is generated from one *Objective*, which is achieved by *Sub Network*, the *Partners* of the *Sub Network* are transferred as partner pools in the collaborative process.
- No.2–3: *Mediator* → Mediator pool. If the *Sub Network* has one *Mediator*, then the *Mediator* is transferred into the mediator pool in the collaborative process.

Part 3: deduction of the partner pool

Rule No.2–2 identifies the partner pools of the collaborative process. However, there is no task in these pools. Part 3 of the deduction rules is thus dedicated to filling the partner pools with tasks and message flows. Fig. 7 shows an example of the deduction of the tasks. There are two rules in this part of the deduction.

- No.3–1: Function → The task in the partner pool. If there is one Function that is shared by one Partner, then the Function is transferred to the task of the partner pool.
- No.3–2: Business Message → Message flow. Based on rule No.3–1, rule No.3–2 adds message flows to the tasks. If one Function has the input/output Business Message, then the Business Message is transferred to the input/output message flows of the task, which is then transferred from the Function.

Part 4: deduction of the mediator pool

Rule No.2–3 determines the mediator pool. Part 4 fills the mediator pool with mediator tasks, message flows and events. Fig. 8 shows an example of the deduction of the mediator tasks, message flows and events. There are three rules in this part of the deduction.

- No.4–1: *Generated Mediator Function* → Task in the mediator pool. This rule transfers all of the functions of the mediator that are defined to achieve the collaborative objective. If the *Mediator* of the collaborative process has the *Generated Mediator Function*, then the *Generated Mediator Function* is transferred to the tasks in the mediator pool.
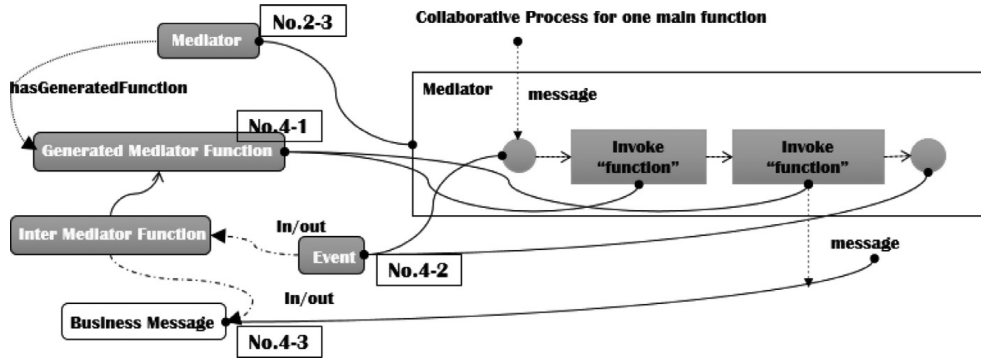
Fig. 8. Deduction of the mediator pool.

Table 9
Results of the knowledge filling from the MIT process handbook.

Concepts of MIT PH	Instances	Concepts of collaborative ontology	Instances
Cartography	9	Objective	131
Function	195	Function	195
Message	75	Message	75
		Main function	156
		Generated mediator function	195
		Inter mediator function	124
		Event	34
Total	279	Total	910

- No.4–2: *Business Message* → Message flow. Based on rule No.4–1, rule No.4–2 adds the message flows to the mediator functions. If one *Business Message* is the input/output of the *Generated Mediator Function*, then the *Business Message* is transferred to the input/output message flow of the mediator task, which is transferred from the *Generated Mediator Function*.
- No.4–3: *Event* → Start/Intermediate/End message event. If there is one *Event* that is the input or output of one *Inter Mediator Function*, then that *Event* is transferred to the start/intermediate/end message event before or after the task, which is then transferred from the *Inter Mediator Function*.

6. Results

To support the model definition and model transformation, Mediator modeling 2ool was developed. The developed method is based on SaaS [26]. The developed tools include GWT[7], GWT-EXT[8], Java, PetalsLink Geasy modeling API[9], PetalsLink Easy BPMN[10] and Protégé[11]. The main functions of Mediator modeling 2ool are (i) to define the collaborative network model, (ii) to link the objectives with ontology, (iii) to transfer the defined model to the cartography of the collaborative process and (vi) to update the collaborative ontology. The structure of the software is shown in Fig. 9.

The software has a server side and a client side. Between the server and the client, there are graphic, ontology and process interfaces to manage the communications. On the client side, the "modeling graphic module" is developed by using the PetalsLink Geasy graphic API, which defines the collaborative network model and sends the model to the server side through the "graphic inter". "Graphic<->XML exchanging engine" receives the graphic model through the "graphic inter" and then changes the model to an XML file containing all of the necessary information. The "Collaborative ontology engine" receives the XML file and updates the collaborative ontology. The "Deduction engine" reads the updated collaborative ontology and deduces the cartography of the collaborative processes in .BPM files. The .BPM files are sent to a BPMN modeling tool through the "process inter".

6.1. Implementation results

Table 9 shows the knowledge transfer results from the MIT process handbook to the collaborative ontology. We use Protégé to define the collaborative ontology. Jena, a tool inside Protégé, implements the transformation rules, which are defined in Section 4.

Fig. 10 shows the client interface of Mediator Modeling 2ool. It also shows the "Ontology" panel. The "Ontology" panel loads the collaborative ontology from the .owl file through the "Ontology inter" (Fig. 9).

7 http://www.gwtproject.org/.
8 http://gwt-ext.com/.
9 https://research.linagora.com/display/geasytools/GEasyTools+Overview.
10 http://bpmneditor.petalslink.com/.
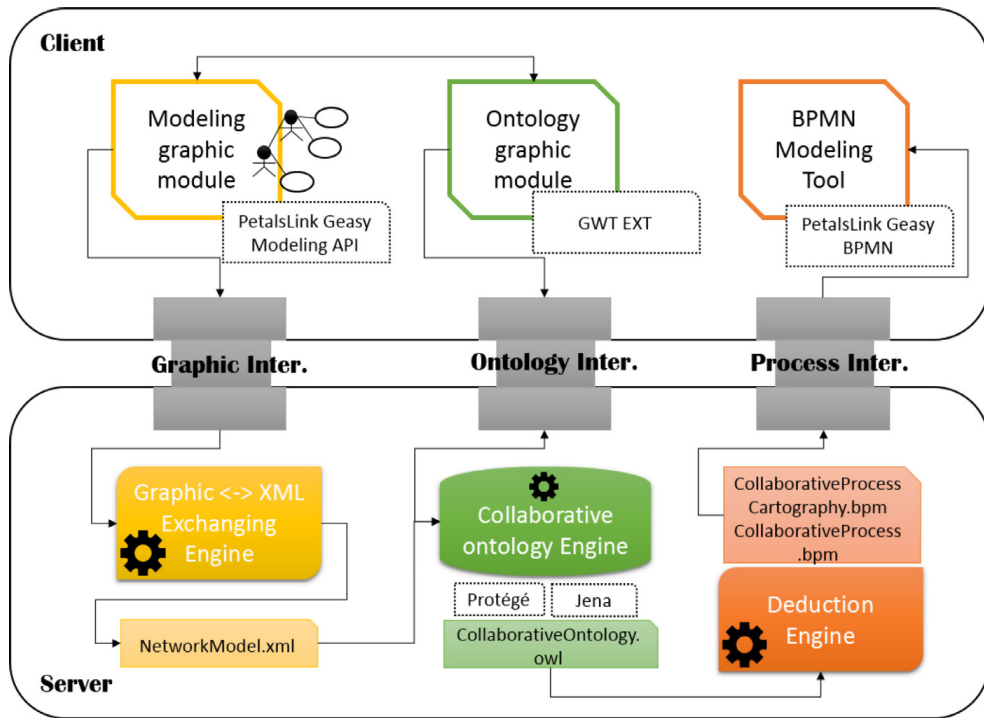11 http://protege.stanford.edu/.

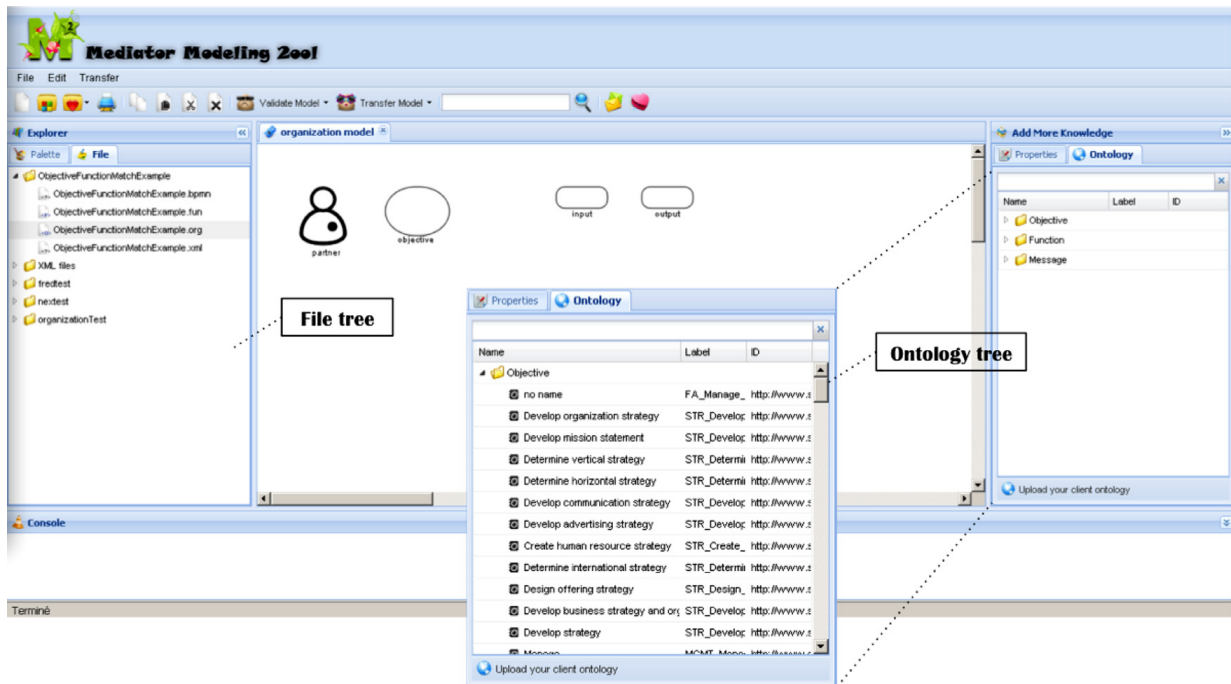**Fig. 9.** Structure of the supporting tools.

Fig. 10. Main interface (file and ontology).
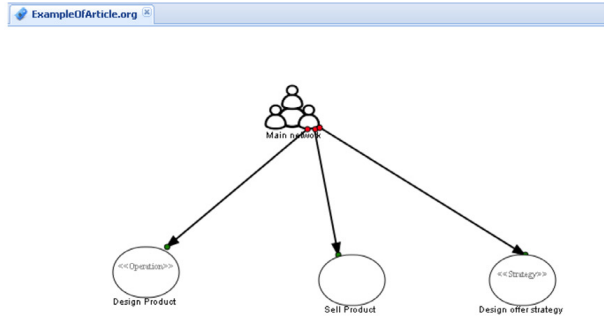
6.2. Case study

A case study of the transformation result of the collaborative process cartography was designed for two types of cases: a small case and a complex case.

Collaborative Situation - Collaborative network model

Both cases are based on the same collaborative objectives (Fig. 11): the Strategy objective, which is for the design offer strategy, the Operation objective, which is for the design product, and the General (Operation and support) objective, which sells the product.
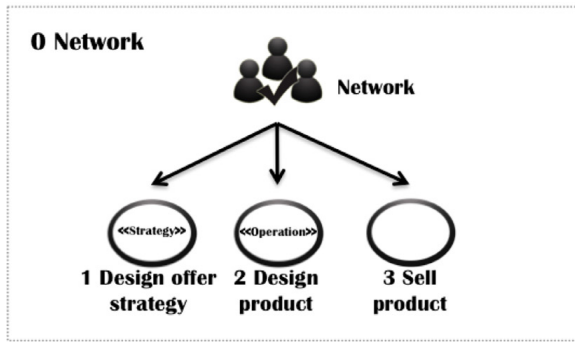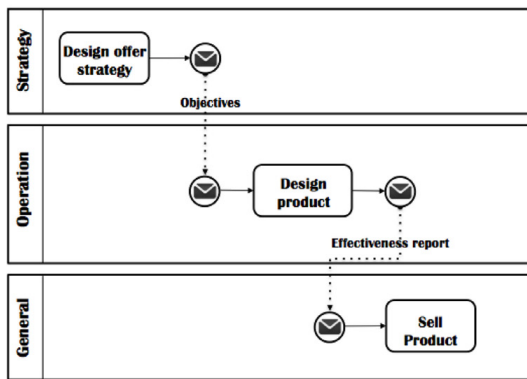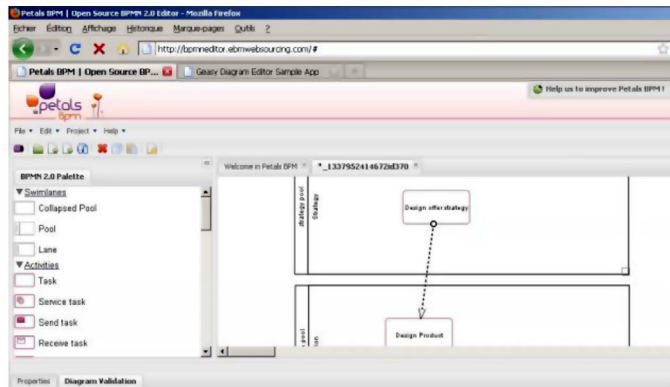
a) Design of the Collaborative network model   b) Implementation of the software tool

Fig. 11. Collaborative network model.
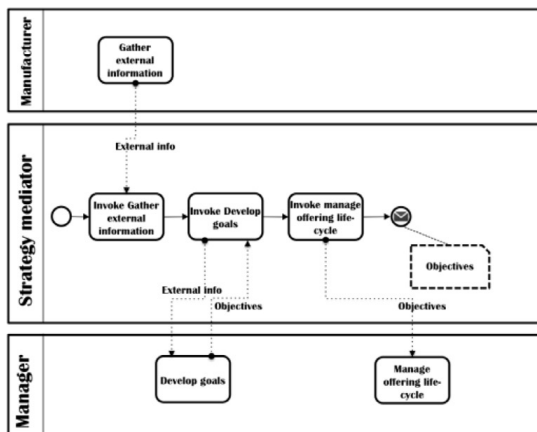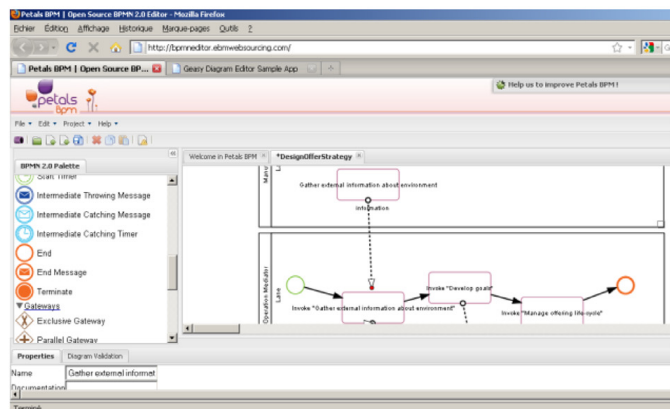

a) Ideal results   b) Transformation results

Fig. 12. Ideal results and transformation results of the process cartography.

a) Ideal results   b) Transformation results

Fig. 13. Ideal results and transformation results of the collaborative process.

Transformation results of the collaborative process cartography

Fig. 12 compares the ideal results and the transformation results: (i) the pools and functions are transformed without mistakes, (ii) the lanes are added automatically with the BPMN tool; and iii) all of the message events are replaced with message flows.

Transformation results of the collaborative process "Design offer strategy"

Fig. 13 compares the ideal results and the transformation results: (i) the pools and functions are transformed without mistakes, (ii) the lanes are added automatically because of the BPMN tool, and (iii) the message event at the end of the process is missing due to the BPMN tool.
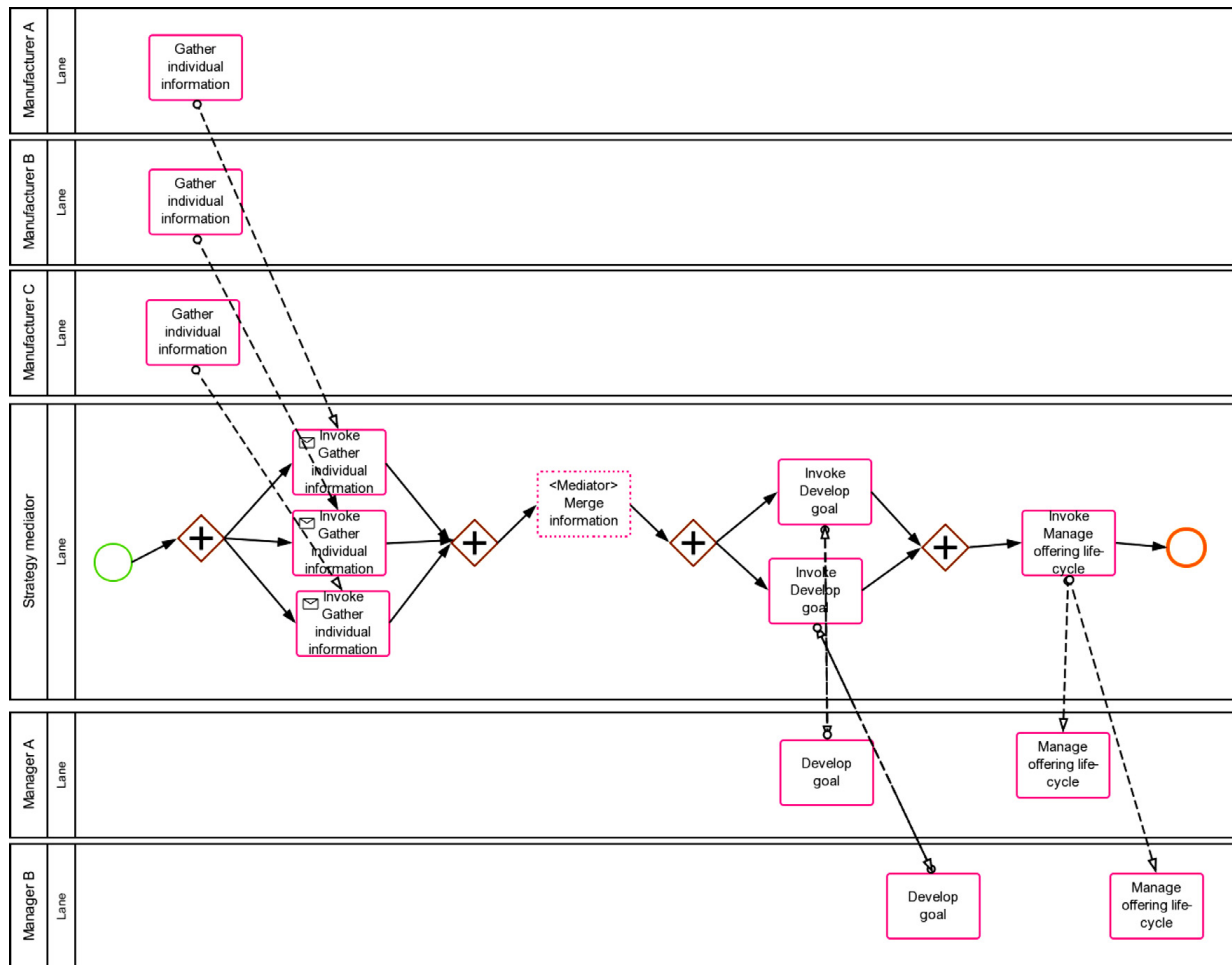
Fig. 14. Results of a complex case.

Several Partners are involved in the "Design offer strategy"
Fig. 14 shows the transformation results for a complex case with several partners involved.

7. Evaluation

Castela et al. [8] analyze their collaborative business process by developing case studies and calculating the numbers and percentages of the key elements, e.g., the number of processes, the number of annotated processes, and the percentage of annotated processes. Based on the numbers calculated, they completed an analysis with different functional or non-functional requirements. However, the requirements are defined when the maintenance of a collaborative business process does not fit our needs. Another paper [17] presents non-functional requirements: modifiability, integrality, interoperability, security, automation, flexibility, and usability. Because the cartography of the collaborative process is deduced by a computer and is not defined by humans, the modifiability is deleted. The final goal of the cartography of the collaborative process is to address any interoperability problems in the information systems; for this reason, we selected interoperability and removed integrality. In addition, no security problem was involved, and the security requirement was deleted. The flexibility is managed through the run time rather than the design time of the collaborative process (ongoing MISE 3.0), so we replaced the flexibility with complexity. Finally, we chose interoperability, automation, complexity and usability as the factors for evaluation.

Three cases were built to calculate the number of functions, message flows, sequence flows, gateways and events. "No mediator" means that the collaborative process goes from one partner to another, without any mediator to manage the process. The "Simple case" means that there is a mediator but only a simple mathematic calculation. For example, one partner function is invoked by one mediator function, and the number of partner functions and mediator functions should be equal. If we consider only the mathematics, the number of total functions is simply doubled. The "Complex case" is the real result of the deduction method presented in this paper.

As shown in Fig. 15, the number of functions can be calculated for the three cases. For the "complex case", by the increase in partners, the number of functions can be decreased and is infinitely close to the "no mediator" case. The cartography of the collaborative process shows the strong advantage of addressing a complex collaborative situation. The reasons for this are as follows (i) the cartography separates the collaborative process by sub-processes, (ii) the "Mediator" combines the same partners'
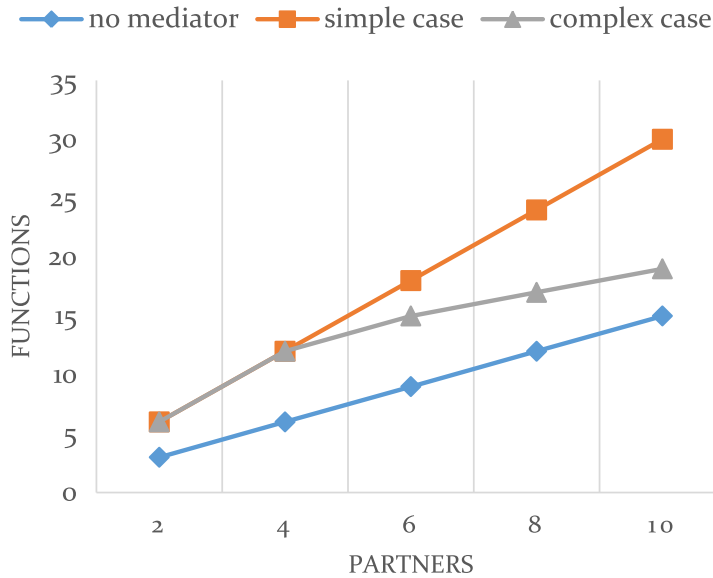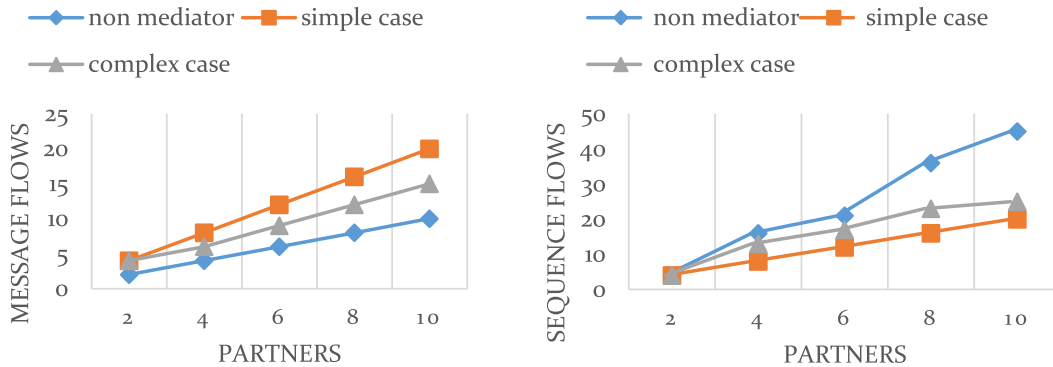
Fig. 15. Number of functions.

Fig. 16. Number of message flows and sequence flows.

functions as one invoking function (for example, the "Invoke Manage offering life cycle" function in Fig. 14), and (iii) the "Mediator" provides one function to merge message flows from different partners and help in decision-making (for example, the "Merge information" function with the "Mediator" tag in Fig. 14). We can trace back to the collaborative process modeling because we consider the workflow changeable in the design phase.

The left part of Fig. 16 shows the number of message flows for the three cases. Compared to the "no mediator" and "simple case", the degree of change for the "complex case" decreases slightly when the number of partners is two or four. This keeps the same degree of change as the other two when there are more than six partners. This shows that the cartography of the collaborative process does not strongly influence the number of message flows. The reason is that only the message-flow merging functions can affect the number of message flows. However, the merging functions are set up when the number of partners is low. These do not change as the number of partners increase. The right part of Fig. 16 shows the number of sequence flows. The degree of change for the "complex case" shows the opposite pattern as the number of message flows. The "Mediator" combines the same functions of partners into one invoking function. For the simple case, more partners lead to more sequence flows. For the complex case, with the invoking function, more partners lead to more sequence flows being saved. With the merging and invoking functions, the complexity of the collaborative process decreases.

Fig. 17 shows the number of events and gateways. On the left, the number of events is noticeably decreasing. In the "no mediator" case, one pool needs at least one start event, one end event and several intermediate events. For the "complex case" and the "simple case", only the mediator pool has one start event, one end event and several intermediate events. On the right, the number of gateways for the "complex case" and the "simple case" are less than those for the "no mediator" case. The reason is that only the mediator pool has gateways. However, for the "complex case", this keeps increasing. For the "simple case", it stays at the same level. This is because more gateways are needed to manage the invoking function compared to the "simple case".

Fig. 18 shows the analysis results with no mediator using the MISE 1.0 deducing method and the MISE 2.0 deducing method (this paper). To summarize:
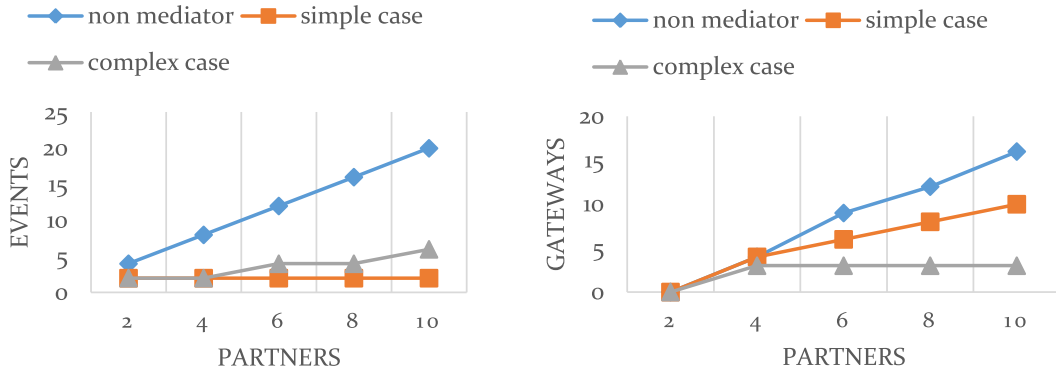
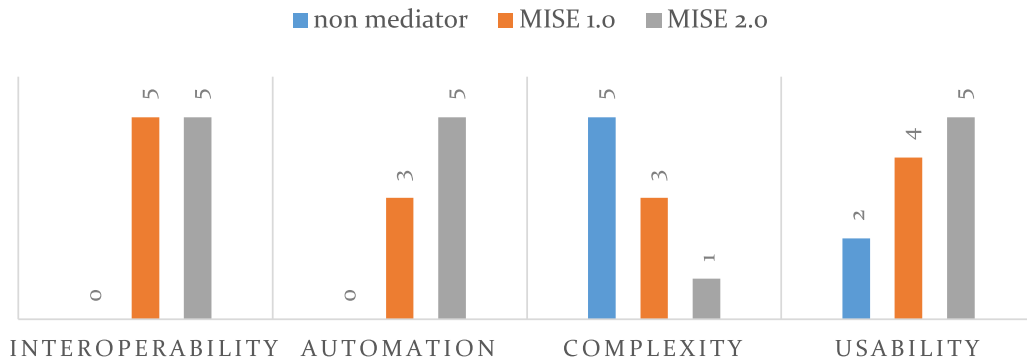**Fig. 17.** Numbers of events and gateways.

Fig. 18. analysis results.

- The methodology considers the workflow changeable to fill the gap between the business process and the technical workflow. It also leads to an unanticipated result: the decreasing complexity of the collaborative process.
- This methodology considers the software to be integrated. All of the transformations are implemented by the software tool. The tool leaves the ontology interface and the modeling interface. They help in the expansion of additional concepts and workflow transformation.

8. Comparison

Collaborative process modeling is a specific domain of business process modeling. To complete the comparison section, we did a survey on business process modeling to search for comparison elements.

Haddar et al. [15] compare business processes by domain dependency, covered business process perspectives, business specificities, modeling language and modeling context. Because all papers mentioned in this paper are general methods or approaches, domain dependency and business specificities are removed from the list.

In Haddar et al. [15], the covered business process perspectives are functional, behavioral, informational and organizational perspectives. de Oca et al. [11] mentioned that "business process models are required as a basis for knowledge transfer between internal and external collaborative partners and documentation in general. It allows us capture the organizational dimension in terms of actors, activities and workflows." We combine these two viewpoints and replace the covered business process perspectives with knowledge modeled. Considering the Specific characteristics of the collaborative process, the knowledge modeled is classified as network (both actor and organizational perspective), objective (both actor and behavioral perspective), function (both activities and functional perspective), message (informational perspective), and process (workflow).

The modeling language is an important element to compare. Kocbek et al. [22] say that "Choosing an appropriate language requires consideration because every modeling language is not suitable for all aspects of the process. This created the necessity for a standardization of modeling languages, which should satisfy the requirements of a high level of expressiveness and formality."

In the work of Haddar et al. [15], the modeling context means whether the artifact type is used in a design or redesign context in a collaborative way. Because all related works are collaborative business process modeling, the modeling context is removed.

As shown in Table 10, all related works mentioned in Section 2 (No. 1 to No. 14 in Table 10) are analyzed and compared with this paper (No. 15 in Table 10) by driving method, tool support, modeling language and knowledge involved. The knowledge involved is classified as network, objective, function, message, and process.

For paper nos. 1–5 and 10, their common points are as follows: (i) main modeled knowledge is function or message; ii) the supporting tool is medium automatic. Compare to Paper No. 15:

Table 10
Global comparisons of related works.

No	Name	General description				Knowledge modeled[1]				
		Driving method	Automa-tion[2]	Modeling language	Evaluation	N[3]	O[3]	F[3]	M[3]	P[3]
1	Comuzzi et al. [9]	PDM optimization algorithm	√√	Petri net	Program test			√		√
2	Antonia et al. [1]	Collaborative working environment structure	√√	Semantic web	Case study/Comparison			√√	√	√
3	Bae et al. [2]	Meta-heuristic programming	√	Petri net	Program test			√√		√
4	Nie et al. [32]	Enterprise integration patterns	√	BPMN 2.0	Program test			√	√√√	√
5	Bianchini et al. [4]	PIS methodology	√	BPNM 2.0	Case study			√√	√√√	
6	Norta et al. [33]	B2B collaborative architecture	√√√	UML Ex[4]	Case study/Comparison			√√√	√√√	√
7	Berhard et al. (2014)	Model driven architecture	√√√	UML Ex	Comparison			√√√	√√√	√
8	Liptchinsky et al. [24]	Graphical notation and formalization	√√	UML	Case study		√	√	√√	√
9	Yongchareon et al. (2014)	Artifact-centric view framework	√√	UML	-	√		√	√	√
10	Zeng et al. [46]	Cross-department coordination patterns	√	Petri Net Ex	-			√√√	√√	√
11	Cortes-Cornax et al. [10]	Intentional fragment, agent	√√√	BPMN 2.0	-	√	√√√	√√	√√	√√
12	Rico et al. [40]	Semantic ontology	√√	Ontology	-		√	√	√	√
13	Li et al. [25]	Fragmentation oriented enterprise network	√√√	UML	Comparison			√√√	√√	√√
14	Meyer et al. [28]	Model-driven architecture	√√√	BPMN 2.0 Ex	-			√	√√√	√√
15	This paper	Collaborative ontology, Model-driven architecture	√√√	BPMN 2.0 Ex	Case study/comparison	√√	√√	√√	√	√√√

Note 1: N = network, O = objective, F = function, M = message, P = process Note 2: √ = with tool support, √√ = well developed, √√√ = full automation
Note 3: √ = mentioned, √√ = modeled, √√√ = modeled well Note 4: Ex = Extension

- Advantages: (i) the research topic is specific and focused; and ii) the patterns of function or message are deeply analyzed and well developed.
- Disadvantages: (i) the network and objective knowledge is not or is slightly modeled; (ii) their result is one collaborative process on an operational level, which is hard to understand and difficult to transfer to BPEL (Business Process Execution Language) [35,43].
- For paper nos. 6 and 7, their common points are: (i) main modeled knowledge is function and message; and (ii) the supporting tool is well developed. Compare to Paper No. 15:
- Advantages: (i) the research topic is specific and focused; (ii) the patterns of function and message are deeply analyzed and well developed; and (iii) the deduction of the collaborative process is automatic and executable.
- Disadvantages: (i) the network and objective knowledge are not modeled; ii) their result is one collaborative process on an operational level, which is hard to understand.

For paper nos. 8, 9 and 12, their common points are: (i) nearly every modeling element is involved; and (ii) the knowledge modeled is additional semantic notation or classification tag. Compare to Paper No. 15:

- Advantages: (i) the modeling method can be combined with other methods to enrich collaborative knowledge and increase the automation of process deduction.
- Disadvantages: (i) the modeling of function and message is too simple; without combination with the other method, the result is not notable; and ii) their result is a UML based collaborative process and is hard to read.

For paper nos. 11, 13 and 14, their common points are: (i) two of the modeling elements are well modeled and related; and (ii) the sub-processes are considered. Compare to Paper No. 15:

- Advantages: (i) by collecting the relations between two key elements, the deduced collaborative process is separated into sub-processes.
- Disadvantages: (i) their result is a collaborative process and sub-processes on the operational level; no strategy and support are involved.

To summarize, Fig. 19. shows all related works that are located in a coordinate system with axes Automation and Knowledge. Compared to Paper nos.1–5 and 10, the detailed modeling of function or message in this paper (Paper No. 15) is not specific enough. This work may be improved by merging the methodologies presented in nos. 8, 9 and 12 to enrich collaborative knowledge. However, it collects almost all types of knowledge in a collaborative situation. It deduces a cartography of collaborative processes, which classify the collaborative processes into strategy, operation and support and separates the collaborative process into sub-processes. This paper shows higher automation than paper nos. 11, 13, and 14; the results are more readable and less complex than paper nos. 6 and 7.
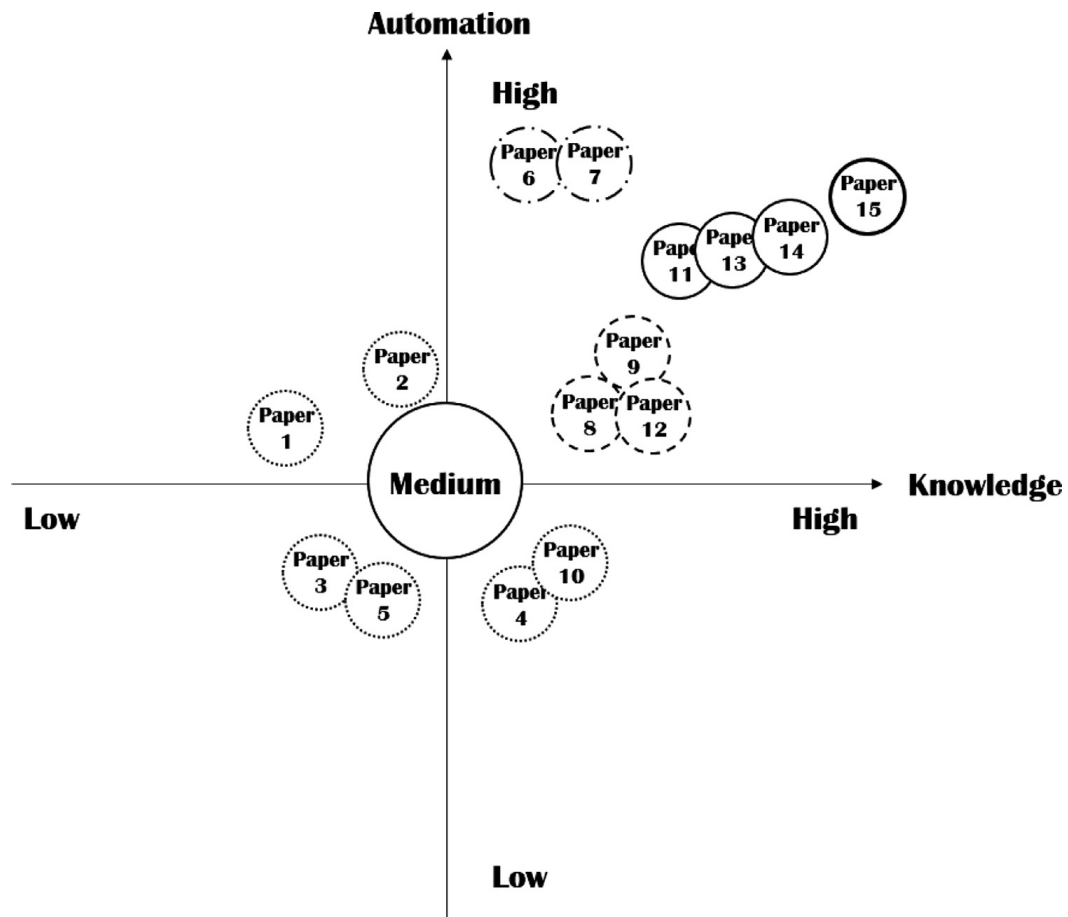
Fig. 19. Summary of comparison.

9. Conclusion

In a collaborative situation, all partners have both collaborative objectives and individual objectives to achieve and business services to share. They expect to combine their own business services with suitable services from other partners to work toward their common objectives. In addition, the collaborative business process is a combination of business functions, which are interlinked and filled with sequences and orders. With these needs, the selection of the business service, which is objective-oriented, and the creation of the business process are absolutely essential in the collaboration world. Considering the self-updating and re-building of the collaborative business process, we have designed an automatic way to address service selection and process creation at the design level. Furthermore, from the implementation viewpoint, the supporting software tool first addresses a collaborative situation, such that all of the partners may use the software at the same time or individually. Additionally, to interact with the other software tools, the software should be invoked. The results of this paper are reused by Boissel-Dallier et al. [5] as an input model. They transfer the collaborative process into an executable collaborative workflow by using semantic web ontologies.

The future works are as follows:

- Characterization with the real-time detection of events; for the whole design-time and run-time, the events can be detected and handled immediately.
- Performance indicator deduction: The indicators measure the performances of the business function and web services. For each collaborative function or web service, the indicator is added to the measure; for example, the launching time, the responding time, and the correction of the output message.
- Workflow monitoring: From both a functional and non-functional view, the user could observe the step in which the workflow is being executed.
- Cloud deploying: All MISE tools and the deployed MIS tools are integrated into the cloud platform. The MISE tools can be uploaded to the cloud in a PaaS (Platform as a Service) as a SaaS.
- Detection by performance monitoring: Using indicators of performance, the detection of agility management can be improved. The indicators can directly launch the detection by identifying the feedback of performance.
- Decision-making transition: For each step of the model transformation, a decision-making mechanism is added. This module suggests the best solution for the choice of functions, processes, and web services.

Acknowledgment

References

[1] M. Antonia Martinez-Carreras, A. Munoz, J. Botia, Building and evaluating context-aware collaborative working environments, Inf. Sci. 235 (2013) 224–241, doi:10.1016/j.ins.2013.02.009.

[2] H. Bae, S. Lee, I. Moon, Planning of business process execution in business process management environments, Inf. Sci. 268 (2014) 357–369, doi:10.1016/j.ins.2013.12.061.

[3] A. Bernstein, C. Dellarocas, T.W. Malone, J. Quimby, Software tools for a process handbook, Data Eng. 51 (1995) 41.

[4] D. Bianchini, C. Cappiello, V. De Antonellis, B. Pernici, Service identification in interorganizational process design, IEEE Trans. Serv. Comput. 7 (2014) 265–278, doi:10.1109/TSC.2013.26.

[5] N. Boissel-Dallier, F. Benaben, J.-P. Lorre, H. Pingaud, Mediation information system engineering based on hybrid service composition mechanism, J. Syst. Softw. 108 (2015) 39–59, doi:10.1016/j.jss.2015.05.064.

[6] I. Bouslimi, K. Ghédira, C. Hanachi, An agent-based organizational model for cooperative information gathering, Adv. Internet Based Syst. Appl. 4879 (2009) 180–189.

[7] J.M. Bryson, B.C. Crosby, M.M. Stone, The design and implementation of cross-sector collaborations: propositions from the literature, Public Admin. Rev. 66 (2006) 44–55, doi:10.1111/j.1540-6210.2006.00665.x.

[8] N. Castela, P. Dias, M. Zacarias, J.M. Tribolet, Collaborative method to maintain business process models updated, Comput. Sci. Inf. Syst. 11 (2014) 461–480, doi:10.2298/CSIS130117031C.

[9] M. Comuzzi, I. Vanderfeesten, T. Wang, Optimized cross-organizational business process monitoring: design and enactment, Inf. Sci. 244 (2013) 107–118, doi:10.1016/j.ins.2013.04.036.

[10] M. Cortes-Cornax, A. Matei, S. Dupuy-Chessa, D. Rieu, N. Mandran, E. Letier, Using intentional fragments to bridge the gap between organizational and intentional levels, Inf. Softw. Technol. 58 (2015) 1–19, doi:10.1016/j.infsof.2014.09.010.

[11] I.M.-M. de Oca, M. Snoeck, H.A. Reijers, A. Rodriguez-Morffi, A systematic literature review of studies on business process modeling quality, Inf. Softw. Technol. 58 (2015) 187–205, doi:10.1016/j.infsof.2014.07.011.

[12] N.C. Dormady, The political economy of collaborative organization, Admin. Soc. 45 (2013) 748–772, doi:10.1177/0095399712438373.

[13] N. Guarino, P. Giaretta, Ontologies and Knowledge bases - towards a terminological clarification, in: Towards Very Large Knowledge Bases, Amsterdam, The Netherlands, IOS Press, 1995, p. 32.

[14] W. Guedria, Y. Naudet, D. Chen, Maturity model for enterprise interoperability, Enterpr. Inf. Syst. 9 (2015) 1–28, doi:10.1080/17517575.2013.805246.

[15] N.Z. Haddar, L. Makni, H. Ben Abdallah, Literature review of reuse in business process modeling, Softw. Syst. Model. 13 (2014) 975–989, doi:10.1007/s10270-012-0286-4.

[16] H.H. Hoang, J.J. Jung, C.P. Tran, Ontology-based approaches for cross-enterprise collaboration: a literature review on semantic business process management, Enterpr. Inf. Syst. 8 (2014) 648–664, doi:10.1080/17517575.2013.767382.

[17] A.Ter Hofstede, W. Van Der Aalst, M. Weske, Business process management: a survey, in: M. Weske (Ed.), Business Process Management, Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, 2003 1019–1019.

[18] B. Hoisl, S. Sobernig, M. Strembeck, Modeling and enforcing secure object flows in process-driven SOAs: an integrated model-driven approach, Softw. Syst. Model. 13 (2014) 513–548, doi:10.1007/s10270-012-0263-y.

[19] ISO 9000, 2005. ISO 9000 Quality management [WWW Document]. URL http://www.iso.org/iso/home/store/publications_and_e-products/publication_item.htm?pid=PUB100224 (accessed 1.19.12).

[20] R. Jardim-Goncalves, C. Coutinho, A. Cretan, C.F. da Silva, P. Ghodous, Collaborative negotiation for ontology-driven enterprise businesses, Comput. Ind. 65 (2014) 1232–1241, doi:10.1016/j.compind.2014.01.001.

[21] J. Jiang, V. Dignum, Y.H. Tan, S. Overbeek, A context-aware inter-organizational collaboration model applied to international trade, Electron. Govern. (2011) 308–319.

[22] M. Kocbek, G. Jost, M. Hericko, G. Polancic, Business process model and notation: the current state of affairs, Comput. Sci. Inf. Syst. 12 (2015) 509–539, doi:10.2298/CSIS140610006K.

[23] D. Krafzig, K. Banke, D. Slama, Enterprise SOA: service-oriented architecture best practices, Prentice Hall PTR, 2005.

[24] V. Liptchinsky, R. Khazankin, S. Schulte, B. Satzger, H.-L. Truong, S. Dustdar, On modeling context-aware social collaboration processes, Inf. Syst. 43 (2014) 66–82, doi:10.1016/j.is.2013.05.007.

[25] Q. Li, Z. Wang, Z. Cao, R. Du, H. Luo, Process and data fragmentation-oriented enterprise network integration with collaboration modelling and collaboration agents, Enterpr. Inf. Syst. 9 (2015) 468–498, doi:10.1080/17517575.2013.879742.

[26] T. Mäkilä, A. Järvi, M. Rönkkö, J. Nissilä, How to define software-as-a-service – an empirical study of finnish saas providers, in: P. Tyrväinen, S. Jansen, M.A. Cusumano, W. Aalst, J. Mylopoulos, M. Rosemann, M.J. Shaw, C. Szyperski, W. Aalst, J. Mylopoulos, M. Rosemann, M.J. Shaw, C. Szyperski (Eds.), Software Business, Springer, Berlin Heidelberg, 2010, pp. 115–124.

[27] T.W. Malone, K. Crowston, G.A. Herman, Organizing business knowledge: the MIT process handbook, the MIT Press, 2003.

[28] A. Meyer, L. Pufahl, K. Batoulis, D. Fahland, M. Weske, Automating data exchange in process choreographies, Inf. Syst. 53 (2015) 296–329, doi:10.1016/j.is.2015.03.008.

[29] J. Miller, J. Mukerji, others, MDA Guide Version 1.0. 1. Object Management Group 234, 51 (2003).

[30] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu, Peer-to-peer computing, Citeseer (2002).

[31] C. Nicolle, J.C. Simon, K. Yetongnon, Interoperability of information systems, Database 1651 (2014) 1650.

[32] H. Nie, X. Lu, H. Duan, Supporting BPMN choreography with system integration artefacts for enterprise process collaboration, Enterpr. Inf. Syst. 8 (2014) 512–529, doi:10.1080/17517575.2014.880131.

[33] A. Norta, P. Grefen, N.C. Narendra, A reference architecture for managing dynamic inter-organizational business processes, Data Knowl. Eng. 91 (2014) 52–89, doi:10.1016/j.datak.2014.04.001.

[34] OMG, Business Process Model And Notation (BPMN) Version 2.0, Object Management Group, 2011.

[35] C. Ouyang, M. Dumas, A.H. Ter Hofstede, W.M. Van Der Aalst, From BPMN process models to BPEL web services, (2006).

[36] S. Patig, V. Casanova-Brito, B. Vögeli, IT Requirements of business process management in practice – an empirical study, in: R. Hull, J. Mendling, S. Tai (Eds.), Business Process Management, Lecture Notes in Computer Science, Springer, Berlin / Heidelberg, 2010, pp. 13–28.

[37] R.M. Pillat, T.C. Oliveira, P.S.C. Alencar, D.D. Cowan, BPMNt: a BPMN extension for specifying software process tailoring, Inf. Softw. Technol. 57 (2015) 95–115, doi:10.1016/j.infsof.2014.09.004.

[38] V. Rajsiri, Knowledge-based system for collaborative process specification, (2010).

[39] V. Rajsiri, J.-P. Lorré, F. Bénaben, H. Pingaud, Knowledge-based system for collaborative process specification, Comput. Ind. 61 (2010) 161–175, doi:10.1016/j.compind.2009.10.012.

[40] M. Rico, M. Laura Caliusco, O. Chiotti, M. Rosa Galli, An approach to define semantics for BPM systems interoperability, Enterpr. Inf. Syst. 9 (2015) 324–347, doi:10.1080/17517575.2013.767381.
[41] R.M. Smullyan, First-order logic, Dover Publications, 1995.
[42] F. Vernadat, Interoperable enterprise systems: principles, concepts, and methods, Annu. Rev. Control 31 (2007) 137–145, doi:10.1016/j.arcontrol.2007.03.004.
[43] S. White, Using BPMN to model a BPEL process, BPTrends 3 (2005) 1–18.
[44] G. Wiederhold, Mediators in the architecture of future information systems, Computer 25 (1992) 38–49.
[45] S. Yongchareon, C. Liu, Y. Jian, X. Zhao, A view framework for modeling and change validation of artifact-centric inter-organizational business processes, Inf. Syst. 47 (2015) 51–81, doi:10.1016/j.is.2014.07.004.
[46] Q. Zeng, F. Lu, C. Liu, H. Duan, C. Zhou, Modeling and verification for cross-department collaborative business processes using extended petri nets, IEEE Trans. Syst. Man Cybern.-Syst. 45 (2015) 349–362, doi:10.1109/TSMC.2014.2334276.
[47] W. Zhang, S. Zhang, F. Qi, M. Cai, Self-organized P2P approach to manufacturing service discovery for cross-enterprise collaboration, IEEE Trans. Syst. Man Cybern.-Syst. 44 (2014) 263–276, doi:10.1109/TSMCC.2013.2265234.