A Methodology Proposal for Collaborative Business Process Elaboration using a Model-Driven Approach

Wenxin Mu, Frederick Benaben, Hervé Pingaud

▶ **To cite this version:**

Wenxin Mu, Frederick Benaben, Hervé Pingaud. A Methodology Proposal for Collaborative Business Process Elaboration using a Model-Driven Approach. Enterprise Information Systems, 2015, 9 (4), pp.349-383. 10.1080/17517575.2013.771410 . hal-01207436

HAL Id: hal-01207436

https://imt-mines-albi.hal.science/hal-01207436

Submitted on 1 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Methodology Proposal for Collaborative Business Process Elaboration using a Model-Driven Approach

Wenxin Mu, Frédérick Bénaben and Hervé Pingaud

Industrial Centre, Toulouse university - Albi-Carmaux, Albi,France

{mwenxin, frederick.benaben, herve.pingaud}@enstimac.fr

Campus Jarlard, Route de Teillet, CT Cedex 09, 81013, Albi

A Methodology Proposal for Collaborative Business Process Elaboration using a Model-Driven Approach

Business Process Management (BPM) principles are commonly used to improve processes within an organization. But they can equally be applied to supporting the design of an Information System (IS). In a collaborative situation involving several partners, this type of BPM approach may be useful to support the design of a Mediation Information System (MIS), which would ensure interoperability between the partners' Information Systems (which are assumed to be service-oriented). To achieve this objective, the first main task is to build a collaborative-business-process cartography. The aim of this article is to present a method for bringing together collaborative information and elaborating collaborative business processes from the information gathered (with the help of a collaborative situation framework, an organizational model, an informational model, a functional model, and a metamodel, and by using model transformation rules).

Keywords: Business process management, model-driven engineering, interoperability, mediation, service-oriented architecture

Subject classification codes: include these here if the journal requires them

Section 1: Introduction

In today's competitive global market, the capacity of enterprises to collaborate with their partners is a critical factor in their development and in their ability to survive [1]–[3]. Most enterprise architectures are now starting to consider collaborative issues [4], [5]. Touzi et al. define four levels of collaborative capacities [6]: (i) communicating: the ability to exchange and share information, (ii) open: the ability to share business services and functionalities with others, (iii) federated: the ability to work with others by following collaborative processes in pursuit of a common objective, as well as the objective of the enterprise itself, and (iv) interoperable: the ability to work with others

without the need for a special effort; the enterprises involved are seen as a seamless system. The concept of interoperability first appeared in the domain of computer science in the early 1990s and has been developed continuously and extensively in many fields, such as military, medical, transportation, software, etc. Several definitions of this concept have been proposed. The most quoted definition was proposed by [7] which defines interoperability as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged". According to [8], "interoperability is the capability of systems, natively independent from each other, to interact in order to build harmonious and finalized collective behaviors without any deep modification of their own structure or behavior".

The InterOp NoE proposes an enterprise interoperability framework [9], which defines three interoperability barriers: conceptual, technological and organizational. Considering that enterprise information systems are the practical and operational parts of an enterprise, one crucial requirement is to break down technological barriers between information systems. The possibility of breaking down organizational and conceptual barriers by breaking down technological barriers is also considered.

As shown in Figure 1, to break organizational, technological and conceptual barriers in a collaborative situation, a Mediation Information System (MIS) provides a solution that to be both practicable and suitable. The concept of mediation was first presented in [10]. Based on this concept, the MIS can deal with process orchestration, data conversion and service selection. For process orchestration, collaborative business processes must first be built. The methodology chosen to elaborate collaborative business processes is based on Model Driven Architecture (MDA) [11]. In MDA, modeling and transformation play important roles. Metamodeling is the underlying feature of MDA. The OMG Modeling Infrastructure [12] illustrates the traditional four-

layer infrastructure (user data, user concepts, Unified Modeling Language concepts and Meta-Objective Facility) that underpins the first generation of Model Driven Development (MDD) technologies. Bezivin proposes a framework of 3+1 metamodeling layers [13]. At the bottom level, the M0 layer is the real system. A model represents this system at level M1. This model conforms to its meta-model defined at level M2 and the metamodel itself conforms to the meta-meta-model at level M3. The meta-meta model conforms to itself. The MISE 2.0 (Mediation Information System Engineering Version 2.0) business process elaboration methodology structure is similar to the OMG modeling infrastructure. The general principle of MISE is that it is structured in two steps between three levels (Figure 2):

(1) The first step concerns the transition from the "characterization of the situation" level to the "collaborative process models" level. By gathering structured knowledge concerning the collaboration under consideration (partners, roles, goals, abilities, etc.), a specific ontology is instantiated to draw up a global characterization of the collaborative situation. Then, by applying elaboration rules to this knowledge, collaborative process models are inferred.

(2) The second step concerns the transition from the "collaborative process models" level to the "MIS deployment" level. The knowledge embedded in these collaborative process models is semantically analyzed in order to apply model-transformation mechanisms dedicated to matching business components (such as business activities from the "collaborative process models" level) with technical components (such as web-services from the "MIS deployment" level). Once technical services have been chosen (to implement the collaborative process from a technical point of view), semantic reconciliation of data should also be performed in order to add transformation of data "on the fly", so that each

technical service will receive the appropriate inputs. The service-oriented MIS structure that is obtained, as described in [14] and [15], can be deployed on the technological target platform, which is an Enterprise Service Bus (ESB). An ESB is a middleware component able to carry messages efficiently between connected services and, by extension, potentially able to orchestrate workflows between connected services (if a workflow engine is plugged to exploit the communication facilities of the ESB).

The first step is considered as the abstract level, *i.e.* the "*collaborative business process elaboration level*", and is the global topic of this article. The second one, "*MIS technical deployment*", is dedicated to the concrete level and is fully described in [16].

The collaborative business process elaboration methodology structure is divided into two main parts (Figure 3): gathering collaborative knowledge and translating the knowledge into collaborative business processes.

In the "gathering collaborative knowledge" part, the models (second level: user concepts in traditional OMG modeling infrastructure) are defined to model user data (first level: user data in traditional OMG modeling infrastructure). In the knowledge-gathering phase, users provide models according to the modeling definitions in this article. In the "elaborate collaborative processes" phase, the metamodel in UML (third level: UML concepts in traditional OMG modeling infrastructure), the transformation rules and the collaborative business process models are defined.

Section 2 of this paper explains the models that present gathered collaborative knowledge. Section 3 presents a detailed methodology for collaborative business process elaboration. Section 4 presents an example, which covers the whole elaboration approach, to help readers understand the methodology better.

Section 2: Introduction to MISE 2.0 Models

In the knowledge-gathering phase of MISE 2.0, the models are defined and chosen to present and collect knowledge. The requirements of these models are focused on collaborative networks, business functions and dynamic description. For collaborative networks, organizational models should be used to describe the knowledge [17]. For the list of functions, a function model is necessary to collect functional knowledge. For dynamic description, the two main components are sequence/order (process model) and flows (informational model). This section answers the question: what are the models chosen for knowledge gathering? Why are these models chosen? And how to use them?

MISE 2.0 defines an organizational model, a functional model, an informational model and a process model. To manage knowledge in these models and transfer knowledge from the organizational model, the functional model and the informational model to the process model, a metamodel is also defined (Figure 4). In section 2.1, the organizational model/collaborative network model is introduced. In section 2.2, the functional model/IDEF0-based functional main model and functional table are presented. In section 2.3, the informational model/IDEF1 is briefly represented. Section 2.4 presents the BPMN-based collaborative process model. Finally, the overseeing metamodel, matching rules from the above models to metamodel elements and transformation rules inside the metamodel, is detailed in section 3.

Section 2.1: Collaborative Network Model

Organizational modeling is not a new subject in enterprise modeling. But most of the organizational models only define the organization chart of enterprises, in terms of responsibilities, departments and workers. In a collaborative situation, the structure is a graph (in discrete mathematics terms) rather than a tree. Based on the concept of

topology [18], the organizational model – a collaborative network model is defined.

The collaborative network model is an objective-oriented organizational model. This model is defined to bring together: (i) the collaborative network partners and partner relationships and (ii) the objectives of the main network, sub-networks and partners. "*An objective model is required to facilitate: (i) identification, communication and structuring of business objectives, and (ii) measurement of the level of success in achieving objectives. But individual modeling methodologies focus primarily on selected aspects of objectives representation and measurement* [19]". Rajsiri et al. [15] have proposed a definition of a collaborative network model, which models the collaborative network and the collaborative main goal. But to cater for individual needs, the present collaborative network model should collect both the collaborative main goals and the partner individual objectives. For each collaborative goal, partners are grouped in a sub-collaborative network. But partners have also their own objectives. A real collaborative situation is like a multi-level pyramid: each level can be broken down from the whole collaborative network into several sub-networks until reaching the end-nodes, i.e. the individual partners.

As shown in the left-hand part of Figure 5, there are four main elements in the collaborative network model: partner, collaborative network, objective and relationship. In the right-hand part, the table defines the possible connections among different elements. The explanation of each element is listed as follows:

- A collaborative network can either represent the whole collaborative network, (which is made up of all the partners) or it can represent a sub-network (a part of the whole network, which involves several partners). The whole network contains the main objectives. The objectives can be implemented by a sub-network.

- “Partner” means organizations, persons, enterprises, etc., which are involved in a collaborative situation.
- “Objective” means a goal, which is a desired result for a partner, a collaborative network or a part of the network, a plan and commitments to be achieved - a personal or organizational desired end-point in some sort of assumed development in the collaborative situation. “Objective” is classified into three types: Strategy Objective, Operation Objective and Support Objective.
- “Relationship” contains two parts: Objective Relationship and Partner Relationship (Strategy Relationship, Operation Relationship and Support Relationship). If a Partner or a Collaborative Network has an Objective, then an Objective Relationship is created between them. If one Partner co-works with another Partner under the same Strategy Objective, then the two Partners own a Strategy Relationship (ibid for Operation Relationship and Support Relationship).

The right-hand part of Figure 5 shows the relationships that are used among the modeling elements. Partner, Collaborative Network, Objective, Operation Objective, Strategy Objective and Support Objective fill the first line and the first column. Objective Relationship, Operation Relationship, Strategy Relationship and Support Relationship fill other cells of the matrix, if the correspondence can be made between a first-column modeling element and a first-line modeling element. For example, Operation Relationship, Strategy Relationship or Support Relationship may relate Partner in the first column to Partner in the first line.

Organizational model definition rules are summarized as follows (using the collaborative network model as an example):

- The first step in building a collaborative network model is to define the objectives of the whole collaborative network. A collaborative network can have one general objective and three types of objectives: strategy, operation and support ([20] explains that business processes contain strategy, operation and support processes. As the goal of this article is to elaborate business processes, the objectives are separated into strategy, operation and support objectives).
- A sub-network implements a general objective. The general objective may also contain small goals. So a sub-network can have several small goals/objectives.
- For a strategy objective, an operation objective or a support objective, the objective can be implemented by a set of partners. All the objectives of the partners must be of the same type as this objective (strategy, operation or support).
- A partner can have a relationship (strategy, operation or support) with other partners. A partner can only have strategy objectives, operation objectives or support objectives (not general objectives).

To explain these model-definition rules, an example of a collaborative-network model is provided in section 3.1.

Section 2.2:IDEF0-Based Functional Model

The requirements for the functional model are to obtain partner functions, to simplify user modeling tasks and to decrease user workload. The functional model only collects functions that partners want to share and which can be published to other partners. The functional model also needs to model input/output/controlling messages of partner functions. IDEF0 (Integration Definition for Function Modeling) is a function-modeling methodology suitable for describing manufacturing functions, which offers a functional

modeling language for the analysis, development, re-engineering and integration of information systems, business processes or software engineering analysis [21]. As mentioned in [21], IDEF0 is based on SADT (Structured Analysis and Design Technique), developed by Douglas T. Ross in 1977. In its original form, IDEF0 includes both a definition of a graphical modeling language (syntax and semantics) and a description of a comprehensive methodology for developing models.

The standard modeling unit in IDEF0 is shown in Figure 6, on the left-hand side. The modeling unit of IDEF0 is presented in [21].

The box in the middle represents a function in the model. A sub-function model may precisely represent the function. The box meanings are named with verbs or verb phrases. Arrows represent messages, which are transferred among functions. Arrow segments are labeled with nouns or noun phrases. Each side of the function box has a standard meaning in terms of box/arrow relationships. Arrows entering the left-hand side of the box are inputs. Inputs are transformed or consumed by the function to produce outputs. Arrows entering the top of the box are controls. Controls specify the conditions required for the function to produce correct outputs. Arrows leaving a box on the right-hand side are outputs. Outputs are the data or objects produced by the function. Arrows connected to the bottom side of the box represent mechanisms. These arrows identify some of the means that support the execution of the function. Other means may be inherited from the parent box. Mechanism arrows that point downwards are ‘call arrows’. Call arrows enable the sharing of detail between models (linking them together) or between portions of the same model. The ‘called’ box provides detail for the ‘caller’ box.

Based on these explanations of IDEF0, it appears that: i) IDEF0 is well defined as a de facto standard and is widely used. This fits the requirement to decrease user

workload. ii) IDEF0 can model input/output/control messages, iii) IDEF0 can be used in MISE 2.0 with small modifications. IDEF0 was therefore chosen as the base model of the MISE 2.0 functional model. In the MISE 2.0 functional model, IDEF0 is reused in two ways: as the functional main model (Figure 6, middle part) and as a functional table (Figure 6, right-hand part). The functional main model mainly represents the main functions and exchanges messages among the main functions in the whole collaborative situation. In the functional main model, the 'control message' box in IDEF0 is reused. An example of a functional main model is detailed in section 4.2. In the functional table, partners and partner objectives are reused as columns to separate out the whole table. The functions to reach objectives are placed in different columns of the table. Section 4.2 presents an example of a functional table.

Section 2.3: IDEF1 Informational Model

The basic need for the informational model of MISE 2.0 is to model messages, which are transferred among business functions, and to model the properties of each message, which are reused in the BPEL transformation. IDEF1 [22] and UML class diagrams are both suitable for modeling informational. Both of them cover the needs of MISE 2.0 and also provide a modeling method with a well-defined standard and a large number of potential users. However, considering message-relationship and information modeling, IDEF1 is more specific for database design and message modeling. In the case of MISE 2.0, UML provides excessive modeling elements and modeling information. IDEF1 already provides sufficient informational modeling constructs. Therefore, IDEF1 was selected for modeling the informational knowledge in MISE 2.0.

As explained in [22], IDEF1 can be viewed as a method for both analysis and communication in establishing modeling requirements. However, IDEF1 is primarily

focused on supporting the task of establishing the requirements for what information is or should be managed by an enterprise. IDEF1 is generally used to: 1) identify what information is currently managed in the organization, 2) identify which of the problems identified during the needs analysis are caused by lack of management of appropriate information, and 3) specify what information will be managed in implementation.

As shown in Figure 7, entities and relationships among entities are the basic constructs of IDEF1. Entities have characteristic attributes associated with them. Attributes record the values of property entities. The term 'attribute class' refers to the set of attribute-value pairs formed by grouping the name of the attribute and the values of that attribute for individual entities. A collection of one or more attribute classes, which distinguishes one individual entity of an entity class from another, is called a 'key class'. A 'relation' in IDEF1 is an association between two individual entities. The existence of a relation is discovered or verified by noting that the attribute classes of one entity class contain the attribute classes of the key class of the referenced individual entity. A relation class can be thought of as the template for associations that exist between entity classes.

Section 2.4: BPMN-Based Collaborative Process Model

In the process modeling domain, a number of models have been defined, such as flow charts IDEF3, Petri nets, Event Process Chains of ARIS, activity diagrams of UML and, more recently, BPMN. But, as mentioned by Touzi [23], using advanced formalisms to model a process can cover several aspects of processes, including actors (Organizational View) and information (Informational View). Furthermore, one of the objectives of MISE 2.0 is to derive a BPEL file, which is deployed on the ESB to execute the technical process. Both [24] and [25] introduced methods to translate BPMN models into a BPEL. They both agree that, with the goal of BPEL derivation, BPMN is an

effective way to model business processes. Rajsiri et al. [15] even provided a BPMN-based Collaborative process model with several partner pools and one mediator pool. BPMN (Business Process Model and Notation) 2.0 [26] was developed by OMG (Object Management Group) in 2010. In BPMN 2.0, semantic annotations are added to tasks and messages. For example, the tasks are classified as service tasks, send tasks, receive tasks, user tasks, manual tasks, business-rule tasks etc. MISE 2.0 aims to deduce a BPEL file and select web services with the help of semantic annotations. Clearly, BPMN has become a good choice to express collaborative process models.

In this paper, the use of the so-called BPMN-based collaborative process, as introduced in [15], is preferred. But it still needs to be adapted to fit with MISE 2.0. As shown in Figure 8, the collaborative process model has one mediator pool and several partner pools. The mediator pool can be a strategy mediator, an operation mediator or a support mediator. Different mediators can communicate through data objects, which are linked to a 'start event' message or other event messages. Tasks in mediator pools invoke tasks in the partner pool according to a defined collaborative process. Consequently, the target metamodel is not BPMN (or BPMN 2.0) itself, but the specific collaborative process metamodel presented in [6], [15] and [23]. This is a very important point, because translating classical BPMN models directly into BPEL files is not always feasible. Börger [27] has clearly addressed "*an unmediated gap between conceptual and executable BPMN model (in particular if obtained through compilation to more detailed languages like BPEL or to code)*". The considered target metamodel has been built especially for the purpose of restricting BPMN expressivity to a sub-space, specifically dedicated to a collaborative situation, and which can be translatable into BPEL.

Section 3: Collaborative Metamodel and Transformation Rules

The transformation part of MISE 2.0 business process elaboration methodology can be separated into the followed main steps: 1) matching gathered modeling elements of organizational, functional and informational models to metamodel Organizational, Functional and Informational Views by using matching rules; 2) transferring metamodel Organizational, Functional and Informational Views into metamodel Process View by using transformation rules; 3) extracting the collaborative process model by using matching rules from the metamodel Process View to a BPMN-based business process collaborative model. In this section, the collaborative metamodel is first presented. Secondly, all the matching rules from the modeling elements of the collaborative network model, IDEF0-based functional model, IDEF1 and BPMN-based process models to the collaborative metamodel are combined together and introduced. Finally, five groups of transformation rules are presented. They help to transfer the collaborative network model, IDEF0- based functional model and IDEF1 into a collaborative process model. These transformation rules cover functional model and organizational model initial rules and collaborative process model transformation rules.

Section 3.1: Collaborative Metamodel

The collaborative metamodel of MISE 2.0 is shown in Figure 13. In the metamodel, there are four packages (Organizational View, Functional View, Informational View and Process View).

Each package manages one model. The Organizational View (Figure 9) mainly stores the information concerning the collaborative network model (e.g. collaborative network, partners and objectives). The Functional View (Figure 10) mainly manages activities, tasks or functions provided by partners and mediator. The Informational View

(Figure 11) is defined to confirm modeling elements in the IDEF1-based informational model. The Process View (Figure 12) is used to present collaborative process model knowledge. The associations among packages are also defined. These packages are used to describe which functions are used to implement an objective, which messages are transferred among different functions, which mediator activities constitute collaborative process, and so on.

Section 3.1.1 Classes of the Organizational View

First of all, classes in the Organizational View (Figure 9) of the collaborative metamodel are introduced as follows:

- The class *collaborative network* is used to define the organization network in the collaborative situation. One *collaborative network* can have several *objectives*. A *sub collaborative network* can implement one *objective*, which is defined in a higher-level *collaborative network*.
- The class *partner* is used to define a *partner* who is involved in a collaboration situation.
- The class *partner relationship* is linked to the association *partner relationship*. This class is used to store and define the partner relationship value. The *partner relationship* can be a *strategy partner relationship*, an *operation partner relationship* or a *support partner relationship*.
- The class *objective* is used to define the objectives of the partners. The *objective* can be a *strategy objective*, an *operation objective* or a *support objective*.

Section 3.1.2 Classes of the Functional View

In the Functional View (Figure 10), the *partner activities* are functions provided by the

partners. The *collaboration activities* are functions provided by a mediator. The *collaboration activities* are deduced from the *partner activities* by transformation rules (section 3.3).

Partner activity:

- The class *strategy activity* is used to provide strategy or decision service.
- The class *operation activity* is used to provide operational service.
- The class *support activity* is used to provide support activity.

Collaboration activity:

- The class *invoking activity* is used to receive a *message* from a *partner*, send a *message* to a *partner* or send and receive a *message*. These three activities are class *receiving activity*, class *calling activity* and class *receiving and calling activity*.
- The class *added value activity* does not send or receive any *message*. The activity is a service provided by the mediator (e.g. providing a required function that partners do not). The class *translating activity* is used to pass a *message* or change the format of a *message*.

Section 3.1.3 Classes of the Informational View

The exchanged business messages and process communication messages are managed in the Informational View (Figure 11). The exchanged business message supports communication among partners in one type of collaborative process (strategy, operational or support) while the process communication message supports communication among different types of collaborative processes (strategy, operation and support).

The class *message* has got one or more *message relationship* associations with other *messages*. The *message relationship* refers to the *message relationship* association.

The class *exchange business message* contains:

- The class *strategy message*, which is one kind of *exchange business message*. The *strategy message* is transferred from one *strategy activity* to another *strategy activity*.
- The class *operation message*, which is one kind of *exchange business message*. The *operation message* is transferred from one *operation activity* to another *operation activity*.
- The class *support message*, which is one kind of *exchange business message*. The *support message* is transferred from one *support activity* to another *support activity*.

The class *process communication message* contains:

- The class *objective message*, which can be transferred from *strategy process* to *operation process* or from *strategy process* to *support process*.
- The class *feedback message*, which can be transferred from *operation process* to *support process* or from *operation process* to *strategy process*.
- The class *mean message*, which can be transferred from *support process* to *strategy process* or from *support process* to *operation process*.

Section 3.1.4 Classes of the Process View

As shown in Figure 12, the *collaborative process* contains three parts: *strategy process*, *operation process* and *support process*. Each type of process contains activities. Inside

each process, the activities are organized through a *sequence flow*. The class s*equence flow* links two activities. The *sequence flow* can also be linked to *event* or *gateway*. Outside each process, the *message flow* is used to communicate.

The class *collaborative process*:

- The class *strategy process* is used to define a strategy part of a collaborative process. One *strategy process* contains one or more *collaborative strategy activities*.
- The class *operation process* is used to define an operational part of the collaborative process. One *operation process* contains one or more *collaborative operation activities*.
- The class *support process* is used to define a support part of the collaborative process. One *support process* contains one or more *collaborative support activities*.

The class *process communication message flow*:

- The class *objective message flow* is used to send an objective information message from a *strategy process* to an *operation process* or from a *strategy process* to a *support process*.
- The class *feedback message flow* is used to send a feedback information message from an *operation process* to a *support process* or from an *operation process* to a *strategy process*.
- The class *mean message flow* is used to send a mean message from a *support process* to a *strategy process* or from a *support process* to an *operation process*.

Section 3.1.5 Associations between the Organizational View and the Functional View

Associations between the Organizational View and the Functional View (Figure 13):

- The association *implement* from *collaborative network* to *functional model*: with this association, *functional main model* and *functional table* can be initialized.
- The association *implement* from *objective* to *activity*: one *partner activity* achieves a goal, which is described by the *objective*.

Section 3.1.6 Associations between the Functional View and the Informational View

Associations between the Functional View and the Informational View (Figure 13) are used to define to input messages and output messages to each function:

- The association *in* from *message* to *receiving activity*: one *receiving activity* only receives input message without output message.
- The association *out* from *message* to *calling activity*: one *calling activity* only sends one output message without input message.
- The association *in* and *out* from *message* to *receiving and calling activity*: one *receiving and calling activity* has to send and receive messages.
- The association *in* and *out* from *strategy message* to *strategy activity*: one *strategy activity* may have one input *strategy message*, one output *strategy message* or both.
- The association *in* and *out* from *operation message* to *operation activity*: one *operation activity* may have one input *operation message*, one output *operation message* or both.
- The association *in* and *out* from *support message* to *support activity*: one

support activity may have one input *support message*, one output *support message* or both.

The associations *in* and *out* from *process communication messag*e to *partner activity* are used to identify messages which are transferred between different types of activities (for example, between *strategy activity* and *operation activity*).

Section 3.1.7 Associations between the Functional View and the Process View

Associations between the Functional View and the Process View (Figure 13) are used to identify supporting partner and mediator activities in a collaborative process:

- The association *represent* from *collaboration activity* to c*ollaboration strategy/operation/support activity*: one *collaboration strategy/operation/support activity* can refer to one *collaboration activity*. The *collaboration strategy/operation/support activity* defines the size, position and symbol of *collaboration activity* in the process model.
- The association *contain* from *collaboration activity* to *collaboration process*: one *collaborative activity* can have one *sub-collaborative process*.

Association *transferred by* from the Informational View to the Process View:

- One *objective message* is transferred by one *objective message flow*.
- One *feedback message* is transferred by one *feedback message flow*.
- One *mean message* is transferred by one *mean message flow*.

Section 3.2: Matching Rules between Metamodel Concepts

The collaborative network model, IDEF0-based functional model, IDEF1-based

informational model and BPMN-based collaborative model have been defined. The collaborative metamodel has also been introduced. But what about the connections between metamodel concepts? Which class in the metamodel presents a concept in these models? This section answers these questions.

The matching rules from the metamodel concepts of the collaborative network model to the collaborative metamodel Organizational View are presented in Figure 14. On the left-hand side are the concepts defined in the collaborative network model. On the right-hand side are the classes defined in the metamodel. In Figure 14, for "objective relationship", there is the association *implement* between *collaborative network* and *objective*, the association between *objective* and *collaborative network* and the association between *partner* and *objective*. The first association represents a sub collaborative network, which achieves one of the main objectives of the whole collaborative network. The second association shows that a collaborative network has several objectives. The third association shows that a partner in a collaborative network has several objectives.

For the IDEF0-based functional model and the IDEF1 informational model, some part of the knowledge in these two models is shared (for example, input/output messages in IDEF0 and entity in IDEF1). Thus, it is better to introduce the matching rules of the functional and informational metamodel concepts together. The matching rules from the functional and informational metamodel concepts to the collaborative metamodel Functional View and Informational View are listed in Figure 15.

For the BPMN-based collaborative process metamodel concepts, the matching classes cover the Organizational, Functional, Informational and Process Views. The matching rules are listed in Figure 16.

Section 3.3: Transformation Rules inside the Metamodel

To define initial rules formally, the rules have been defined with first-order logic [28]. Because of the specialization of model transformation, first-order logic still needs to be expanded. The expanded rules are listed as follows:

(3) Class: X is collaborative network → collaborative network (X)

(4) Association: Y is association *implement* which is between collaborative network X1 and objective X2 → implement (Y) (collaborative network (X1), objective (X2))

(5) If-then-else: if (X) → then (Y), else if (X1) → then (Y1), else → then (Y2)

(6) A set of variables: from X_1, X_2, X_3 to X_n→ X_1 … X_n

The transformation rules are defined in six groups. As shown in Figure 17, according to the matching rules in section 3.2, the classes in white present the knowledge gathered by the collaborative network model, IDEF0-based functional model and IDEF1 model. The gray and black classes need to be deduced by the transformation rules.

Group 1: *collaborative network →functional model*. The Group 1 Transformation Rules are used to initialize the *functional model* (Equations no.1 and no.2). For any collaborative network with a sub-network, one *functional main model* is initialized. For any collaborative network without a sub-network, one *functional table* is initialized.

$$\exists collaborativenetwork(\mathrm{x})\left(\exists implement\left(collaborativenetwork(\mathrm{x}), objective(\mathrm{x}_0)\right)\right)$$
$$\wedge \nexists contain\left(collaborativenetwork(\mathrm{x}), partner(\mathrm{x}_1)\right)$$
$$\rightarrow \exists functionalmainmodel(\mathrm{y})$$
$$\wedge \exists implement\left(collaborativenetwork(\mathrm{x}), functionalmainmodel(\mathrm{y})\right)(\mathbf{1})$$

$$\exists collaborativenetwork(\mathrm{x})\left(\exists implement\left(collaborativenetwork(\mathrm{x}), objective(\mathrm{x}_0)\right)\right)$$
$$\wedge \exists contain\left(collaborativenetwork(\mathrm{x}), partner(\mathrm{x}_1)\right)$$
$$\rightarrow \exists functionaltable(\mathrm{y})$$
$$\wedge \exists implement\left(collaborativenetwork(\mathrm{x}), functionaltable(\mathrm{y})\right)(\mathbf{2})$$

Group 2: *partner activity* →*strategy/operation/support activity* (Equations no.3, no.4 and no.5). This group of transformation rules helps the classification of partner activities. If one *partner activity* links to a *strategy objective*, then the *partner activity* is a *strategy activity*. If one *partner activity* links to an *operation objective*, then the *partner activity* is an *operation activity*. If one *partner activity* links to a *support objective*, then the *partner activity* is a *support activity*.

$$\exists implement\left(strategy\ objective(\mathrm{x}), partner\ activity(\mathrm{x}_0)\right)$$
$$\rightarrow \exists implement\left(strategyobjective(\mathrm{x}), strategyactivity(\mathrm{x}_0)\right)(\mathbf{3})$$

$$\exists implement\left(operationobjective(\mathrm{x}), partneractivity(\mathrm{x}_0)\right)$$
$$\rightarrow \exists implement\left(operationobjective(\mathrm{x}), operationactivity(\mathrm{x}_0)\right)(\mathbf{4})$$

$$\exists implement\left(supportobjective(\mathrm{x}), partneractivity(\mathrm{x}_0)\right)$$
$$\rightarrow \exists implement\left(supportobjective(\mathrm{x}), supportactivity(\mathrm{x}_0)\right)(\mathbf{5})$$

Group 3: *exchanged business message*→*strategy/operation/support message and process communication message*→*objective/feedback/mean message* (Equations

no.6 to no.11). This group of transformation rules is defined to classify *exchanged business messages* and *process communication messages*. If one *exchanged business message* is an input or output message for a *strategy activity*, then the *exchange business message* is a *strategy message*. If one *exchanged business message* is an input or output message for an *operation activity*, then the *exchanged business message* is an *operation message*. If one *exchanged business message* is an input or output message for a *support activity*, then the *exchanged business message* is a *support message*. If a *process communication message* is an output message of a *strategy activity* and an input message of an *operation* or *support activity*, then the *process communication message* is an *objective message*. If a *process communication message* is an output message of an *operation activity* and an input message of an *objective activity* or *support activity*, then the *process communication message* is a *feedback message*. If a *process communication message* is an output message of a *support activity* and an input message of a *strategy activity* or a *support activity,* then *the process communication message* is a *mean message*.

$$\exists exchangedbusinessmessage(x)\left(\exists in(m)\begin{pmatrix} strategyactivity(x_1), \\ exchangedbusinessmessage(x)\end{pmatrix}\right) \rightarrow \exists strategymessage(x)(\mathbf{6})$$

$$\exists exchangedbusinessmessage(x)\left(\exists in(m)\begin{pmatrix} operationactivity(x_1), \\ exchangedbusinessmessage(x)\end{pmatrix}\right) \rightarrow \exists operationmessage(x)(\mathbf{7})$$

$$\exists exchangedbusinessmessage(x)\left(\exists in(m)\begin{pmatrix} supportactivity(x_1), \\ exchangedbusinessmessage(x)\end{pmatrix}\right) \rightarrow \exists supportmessage(x)(\mathbf{8})$$

$$\exists processcommunicationmessage(x)\left(\exists out(m_1)\begin{pmatrix} strategyactivity(x_1), \\ processcommunicationmessage(x)\end{pmatrix}\right.$$
$$\wedge \exists in(m_2)\begin{pmatrix} operationactivity(x_2), \\ processcommunicationmessage(x)\end{pmatrix}$$
$$\left.\vee \exists in(m_3)\begin{pmatrix} supportactivity(x_3), \\ processcommunicationmessage(x)\end{pmatrix}\right) \rightarrow \exists objectivemessage(x)(\mathbf{9})$$

$$\exists processcommunicationmessage(x)\left(\exists out(m_1)\begin{pmatrix} operationactivity(x_1), \\ processcommunicationmessage(x)\end{pmatrix}\right.$$
$$\wedge \exists in(m_2)\begin{pmatrix} strategyactivity(x_2), \\ processcommunicationmessage(x)\end{pmatrix}$$
$$\left.\vee \exists in(m_3)\begin{pmatrix} supportactivity(x_3), \\ processcommunicationmessage(x)\end{pmatrix}\right)$$
$$\rightarrow \exists feedbackmessage(x)\ (\mathbf{10})$$

$$\exists processcommunicationmessage(x)\left(\exists out(m_1)\begin{pmatrix} supportactivity(x_1), \\ processcommunicationmessage(x)\end{pmatrix}\right.$$
$$\wedge \exists in(m_2)\begin{pmatrix} strategyactivity(x_2), \\ processcommunicationmessage(x)\end{pmatrix}$$
$$\left.\vee \exists in(m_3)\begin{pmatrix} operationactivity(x_3), \\ processcommunicationmessage(x)\end{pmatrix}\right) \rightarrow \exists meanmessage(x)(\mathbf{11})$$

Group 4: *partner activity*→*collaboration activity* (Equations no.12 to no.14). The transformation rules of Group 4 are used to create collaboration activities in the Functional View. If a *partner activity* has got one input message, then a *calling activity* and an *association out* to the message are created. An association *invoked by* from a *partner activity* to a *calling activity* is created. If a *partner activity* has one output message, then a *receiving activity* and an *association in* to the message are created. An association *invoked by* from a *partner activity* to a *receiving activity* is created. If a *partner activity* has both input message and output message, then a *receiving* and *calling activity* and an *association in/out* to the messages are created. An association *invoked by* from a *partner activity* to a *calling* and *receiving activity* is created.

$$\exists partner\ activity(\mathrm{x})\begin{pmatrix}\exists in(\mathrm{x}_1)\big(partneractivity(\mathrm{x}), message(\mathrm{x}_2)\big)\\ \wedge\ \exists out(\mathrm{x}_3)\big(partneractivity(\mathrm{x}), message(\mathrm{x}_4)\big)\end{pmatrix}$$

$$\rightarrow \exists receiving\ and\ calling\ activity(\mathrm{y})\begin{pmatrix}\exists in(\mathrm{y}_1)\begin{pmatrix}receiving\ and\ calling\ activity(\mathrm{y}),\\ message(\mathrm{x}_2)\end{pmatrix}\\ \wedge\ \exists out(\mathrm{y}_2)\begin{pmatrix}receiving\ and\ calling\ activity(\mathrm{y}),\\ message(\mathrm{x}_4)\end{pmatrix}\end{pmatrix}$$

$$\wedge \exists invoked\ by(\mathrm{z})\big(partner\ activity(\mathrm{x}), receiving\ and\ calling\ activity(\mathrm{y})\big)(\mathbf{12})$$

$$\exists partner\ activity(\mathrm{x})\begin{pmatrix}\exists in(\mathrm{x}_1)\big(partneractivity(\mathrm{x}), message(\mathrm{x}_2)\big)\\ \wedge\ \nexists out(\mathrm{x}_3)\big(partner\ activity(\mathrm{x}), message(\mathrm{x}_4)\big)\end{pmatrix}$$

$$\rightarrow \exists calling\ activity(\mathrm{y})\begin{pmatrix}\exists in(\mathrm{y}_1)\big(calling\ activity(\mathrm{y}), message(\mathrm{x}_2)\big)\\ \wedge\ \nexists out(\mathrm{y}_2)\big(calling\ activity(\mathrm{y}), message(\mathrm{x}_4)\big)\end{pmatrix}$$

$$\wedge \exists invokedby(\mathrm{z})\big(partner\ activity(\mathrm{x}), calling\ activity(\mathrm{y})\big)(\mathbf{13})$$

$$\exists partner\ activity(\mathrm{x})\begin{pmatrix}\nexists in(\mathrm{x}_1)\big(partner\ activity(\mathrm{x}), message(\mathrm{x}_2)\big)\\ \wedge\ \exists out(\mathrm{x}_3)\big(partner\ activity(\mathrm{x}), message(\mathrm{x}_4)\big)\end{pmatrix}$$

$$\rightarrow \exists receiving\ activity(\mathrm{y})\begin{pmatrix}\nexists in(\mathrm{y}_1)\big(receiving\ activity(\mathrm{y}), message(\mathrm{x}_2)\big)\\ \wedge\ \exists out(\mathrm{y}_2)\big(receiving\ activity(\mathrm{y}), message(\mathrm{x}_4)\big)\end{pmatrix}$$

$$\wedge \exists invoked\ by(\mathrm{z})\big(partner\ activity(\mathrm{x}), receiving\ activity(\mathrm{y})\big)(\mathbf{14})$$

Group 5: *collaborative activity* →*collaborative strategy/operation/support activity* and *sequence flow* (Equations no.15 to no.17). The Transformation Rules of Group 5 are used to create sequence flows in Process View. This group of transformation rules is implemented by a breadth-first traversal algorithm graph [29]. A functional model can be analyzed as a graph. The function boxes can be seen as nodes. The input/output messages can be seen as arrows in a graph. The algorithm is summarized as follows: (i) If one pre-node has one post-node, then create a *sequence flow* between the pre-node and post-node (pre-node and post-node are *partner activities*, but they are linked to *collaborative strategy/operation/support activity* by a *collaborative activity* through an association *represented by* and an association *invoked by*, so the sequence flow is created between collaborative strategy/operation/ support

activities). (ii) If one pre-node has several post-nodes, *parallel gateway* and *sequence flows* are created among the pre-node and post-nodes. (iii) If one post-node has several pre-nodes, *sequence flows* and *parallel gateway* are created among the pre-nodes and post-nodes. The first logic equation for this group is also defined. The *strategy activity* is used as an example to present the first logic equation.

$$\exists \text{strategy activity}(x_1), \text{strategy activity}(x_2)$$
$$\wedge \exists \text{out}(y_1)\big(partner\ activity(x_1), message(x_3)\big)$$
$$\wedge \exists \text{in}(y_2)\big(partner\ activity(x_2), message(x_4)\big)$$
$$\rightarrow \exists collaborative\ strategy\ activity(z_1)$$
$$\wedge \exists collaborative\ strategy\ activity(z_2)$$
$$\wedge \exists sequenceflow(s)\big(collaborative\ strategy\ activity(z_1),$$
$$collaborative\ strategy\ activity(z_2)\big) \mathbf{(15)}$$

$$\exists strategy\ activity(x_0), strategy\ activity(x_1) \dots strategy\ activity(x_n) \wedge$$
$$\exists out(y_0)\big(partner\ activity(x_0), message(m)\big) \wedge$$
$$\exists in(y_1)\big(partner\ activity(x_1), message(m)\big) \dots \wedge$$
$$\exists in(y_n)(partner\ activity(x_n), message(m) \rightarrow$$
$$\exists collaborative\ strategy\ activity(z_0) \wedge$$
$$\exists collaborative\ strategy\ activity(z_2) \dots \wedge$$
$$\exists collaborative\ strategy\ activity(z_n) \wedge \exists gateway(g) \wedge$$
$$\exists sequenceflow(s_0)\big(collaborative\ strategy\ activity(z_0), gateway(g)\big) \wedge$$
$$\exists sequenceflow(s_1)\big(collaborative\ strategy\ activity(z_1), gateway(g)\big) \dots \wedge$$
$$\exists sequence\ flow(s_n)(collaborative\ strategy\ activity(z_n), gateway(g)) \mathbf{(16)}$$

$$\exists strategy\ activity(x_0), strategy\ activity(x_1) \dots strategy\ activity(x_n)$$
$$\wedge \exists out(y_1)\big(partner\ activity(x_1), message(m)\big)$$
$$\wedge \exists out(y_n)\big(partner\ activity(x_n), message(m)\big)$$
$$\wedge \exists in(y_0)\big(partner\ activity(x_0), message(m)\big)$$
$$\rightarrow \exists collaborative\ strategy\ activity(z_0)$$
$$\wedge \exists collaborative\ strategy\ activity(z_2) \dots$$
$$\wedge \exists collaborative\ strategy\ activity(z_n) \wedge \exists gateway(g)$$
$$\wedge \exists sequenceflow(s_1)\big(collaborative\ strategy\ activity(z_1), gateway(g)\big) \dots$$
$$\wedge \exists sequence\ flow(s_n)\big(collaborative\ strategy\ activity(z_n), gateway(g)\big)$$
$$\wedge \exists sequence\ flow(s_0)\big(collaborative\ strategy\ activity(z_0), gateway(g)\big) \quad (\mathbf{17})$$

Group 6: *collaborative activity* →*collaborative strategy/operation/support activity* and *sequence flow* (Equations no.18 to no.20). The Transformation Rules of Group 6 are used to create message flows in the Process View. If there is a process communication message, which is *in/out* to *partner activity*, then *process communication message flow*, which is *in/out* to *collaborative strategy/operation/support activity*, is created.

$$\Big(\exists objective\ message(\mathrm{x})\ \Big(\exists out(\mathrm{m}_1)\big(strategy\ activity(\mathrm{x}_1), objective\ message(\mathrm{x})\big)$$

$$\wedge \exists in(\mathrm{m}_2)\big(operation\ activity(\mathrm{x}_2), objective\ message(\mathrm{x})\big)$$

$$\vee \exists in(\mathrm{m}_3)\big(support\ activity(\mathrm{x}_3), objective\ message(\mathrm{x})\big)\Big)\Big)$$

$$\wedge \exists \mathrm{x}_1\ \Big(\exists invoked\ by\big(\mathrm{x}_1, invoking\ activity(\mathrm{X}_1)\big)\Big)$$

$$\wedge \exists \mathrm{x}_2\ \Big(\exists invoked\ by\big(\mathrm{x}_2, invoking\ activity(\mathrm{X}_2)\big)\Big)$$

$$\wedge \exists \mathrm{x}_3\ \Big(\exists invoked\ by\big(\mathrm{x}_3, invoking\ activity(\mathrm{X}_3)\big)\Big)$$

$$\wedge \exists \mathrm{X}_1\ \Big(\exists represent\ \Big(\mathrm{X}_1, collaborative\ strategy\ activity\big(\mathrm{y}_1\big)\Big)\Big)$$

$$\wedge \exists X_2\ \Big(\exists represent\ \Big(X_2, collaborative\ operation\ activity\big(y_2\big)\Big)\Big)$$

$$\wedge \exists X_3\ \Big(\exists represent\ \Big(X_3, collaborative\ support\ activity\big(y_3\big)\Big)\Big)$$

$$\rightarrow \exists objective\ message\ flow(\mathrm{y}) \wedge \exists out\big(\mathrm{y}_1, \mathrm{y}\big) \wedge \exists\ \Big(in\big(\mathrm{y}_2, \mathrm{y}\big) \vee in\big(\mathrm{y}_3, \mathrm{y}\big)\Big) \quad (18)$$

$$\Big(\exists feedbackmessage(x)\Big(\exists out(m_1)\big(operationactivity(x_1), feedbackmessage(x)\big)$$
$$\wedge \exists in(m_2)\big(strategyactivity(x_2), feedbackmessage(x)\big)$$
$$\vee \exists in(m_3)\big(supportactivity(x_3), feedbackmessage(x)\big)\Big)\Big)$$
$$\wedge \exists x_1\Big(\exists invokedby\big(x_1, invokingactivity(X_1)\big)\Big)$$
$$\wedge \exists x_2\Big(\exists invokedby\big(x_2, invokingactivity(X_2)\big)\Big)$$
$$\wedge \exists x_3\Big(\exists invokedby\big(x_3, invokingactivity(X_3)\big)\Big)$$
$$\wedge \exists X_1\Big(\exists represent\Big(X_1, collaborativeoperationactivity\big(y_1\big)\Big)\Big)$$
$$\wedge \exists X_2\Big(\exists represent\Big(X_2, collaborativestrategyactivity\big(y_2\big)\Big)\Big)$$
$$\wedge \exists X_3\Big(\exists represent\Big(X_3, collaborativesupportactivity\big(y_3\big)\Big)\Big)$$
$$\rightarrow \exists feedbackmessageflow(y) \wedge \exists out\big(y_1, y\big) \wedge \exists\Big(in\big(y_2, y\big) \vee in\big(y_3, y\big)\Big) \quad (19)$$

$$\Big(\exists meanmessage(x)\Big(\exists out(m_1)\big(supportactivity(x_1), meanmessage(x)\big)$$
$$\wedge \exists in(m_2)\big(operationactivity(x_2), meanmessage(x)\big)$$
$$\vee \exists in(m_3)\big(strategyactivity(x_3), meanmessage(x)\big)\Big)\Big)$$
$$\wedge \exists x_1\Big(\exists invokedby\big(x_1, invokingactivity(X_1)\big)\Big)$$
$$\wedge \exists x_2\Big(\exists invokedby\big(x_2, invokingactivity(X_2)\big)\Big)$$
$$\wedge \exists x_3\Big(\exists invokedby\big(x_3, invokingactivity(X_3)\big)\Big)$$
$$\wedge \exists X_1\Big(\exists represent\Big(X_1, collaborativesupportactivity\big(y_1\big)\Big)\Big)$$
$$\wedge \exists X_2\Big(\exists represent\Big(X_2, collaborativeoperationactivity\big(y_2\big)\Big)\Big)$$
$$\wedge \exists X_3\Big(\exists represent\Big(X_3, collaborativestrategyactivity\big(y_3\big)\Big)\Big)$$
$$\rightarrow \exists objectivemessageflow(y) \wedge \exists out\big(y_1, y\big) \wedge \exists\Big(in\big(y_2, y\big) \vee in\big(y_3, y\big)\Big) \quad (20)$$

Section 4: Illustrative Example

In this section, an example is presented that covers the whole collaborative process

elaboration. This example helps to understand the elaboration methodology.

As shown in Figure 18, the example covers a collaborative network which has four partners: client, assembler, supplier1 and a group of suppliers (supplier 2 and supplier 3). The client buys product from an assembler. The assembler cooperates with suppliers 1, 2 and 3. Among them, supplier 1 is a long-term and stable supplier. Suppliers 2 and 3 provide the same components. The assembler has to choose one suitable supplier from suppliers 2 and 3.

Section 4.1: Organizational Model – Collaborative Network Model – Example

To explain the collaborative network model, a collaborative network model for the example is defined. As shown in Figure 19, the collaborative network model has four levels. The first level is to define the main *objectives* of the whole network. The main objectives can be strategy objectives, operation objectives, support objectives or objectives of no specific type. The first level of the collaborative network model in Figure 19 – the 0 network, defines the strategy objective: *choose partner*, the objective: *sell product* and the operation objective: *sell component*. Each objective has a sub-network to achieve a collaborative goal. For the strategy objective, *choose partner* and for the operation objective, *sell component*, which are defined as strategy or operation objectives, it appears that their sub-network can only contain sub strategy or sub operation objectives. For the objective: *sell product*, the objective has not been sorted, so the sub-network can have different kinds of objectives.

As shown in Figure 19, in the second level, *choose partner* and *sell component* have been enlarged into a sub-network which already contains partners. But for *sell product*, the objective of the sub-network has been sorted into three different kinds of objectives: *place order*, *deliver product* and *pay product*. With itemized objectives,

these objectives can be enlarged directly by a sub-network, which is composed of partners. For these sub-networks, objective types must be the same as for objectives in the previous level.

Section 4.2: IDEF0-Based Functional Model – Example

The functional model in MISE 2.0 is an IDEF0-based functional model. IDEF0 has been reused in two styles: as a main functional model and as a functional table. The organizational model is made of network elements and objective elements (e.g. *0Network* and *2 Sell Product* in Figure 19) and reused to initialize the main functional model. The organizational model, made up of partner elements and objective elements (e.g. *1 Choose Partner*, *3 Sell Product*, *2.1 Place Order* etc. in Figure 20), is reused to initialize the functional table. In this section, *0Network* and *1Choose Partner* are taken to illustrate how the functional model can be initialized.

As shown in Figure 20, the organizational model *0 Network* is initialized into a main functional model *A0 network*. The network objectives in the organizational model are reused as main functions in the functional model. Users have to add control messages among these functions to complete the main functional model. As shown in Figure 20, *1 Choose Partner* is initialized to the functional table: *A1 Choose Partner*. The Functional table reuses partners and partner objectives, separating the functional model into several columns. Users have to complete each column of the functional table by functions and exchanged messages, which can be provided by the partners and support the objective in the column.

A0 Network

As shown in Figure 21, the *A0 Network* is completed. The control messages are added to the main functional model. The function *Choose Partner* triggers *Sell Product* by

sending a control message named *Wait for order trigger*. If *Wait for order trigger* equals true, it means that a partner has been chosen from Supplier 2 and Supplier 3, and the whole collaborative network has been settled and the assembler can start to take orders from the client. To complete the *Sell Product* Function, a message is needed: *Component sold feedback*. If *Component sold feedback* is equal to true, it means that all needed components have been bought, and the assembler can start to assemble products and deliver them.

A1 Choose Partner

Figure 22 shows the completed *A1 Choose Partner*. For each objective, the functions and exchanged messages among functions have been provided by each partner.

In *A1 Choose Partner*, the assembler asks for application reports from Supplier 2 and Supplier 3. After receiving the application reports, the assembler makes a decision to choose one partner from Supplier 2 and Supplier 3 with the help of the application reports. The assembler has to send the final decision to Supplier 2 and Supplier 3. In addition, the assembler needs to send a *Wait for order trigger* to launch the *Sell Product* function.

Section 4.3: IDEF1 Informational Model – Example

In MISE 2.0, input/output messages in the functional model are extracted to create IDEF1 entities. As shown in Figure 23, the functional model *A1 Choose Partner* is selected as an example to show initial results. All the business input/output messages in black and bold are selected and the entities in IDEF1 are created (see bottom of Figure 23).

As shown in Figure 23, the input/output messages in bold black line (*application report require*, *supplier 2 application report*, *supplier 3 application report* and *partner*

chosen decision) are extracted out and presented as entities in the IDEF1 model. In the next step, the user has to add attributes and relationships for these entities (Figure 24).

Section 4.4: BPMN-Based Collaborative Process Model – Example

This section presents the final results for the transformation rules in section 3.3 and the matching rules in section 3.2. As shown in Figure 25, the collaborative process main model has three collaborative process pools (strategy process, operation process and support process). The main process is transferred from the functional main model (*A0 network* and *A2 sell product*). In the main collaborative process, the sub collaborative processes can detail each collaborative task. Among the different collaborative processes, process communication messages (*order taken* trigger, *component sold* feedback and *product delivered* feedback) are defined to trigger another task. In the following sections, the sub collaborative processes, *choose partner*, *send payment* and *deliver product* are presented. Each of them presents a type of collaborative process (strategy, operation, support) and owns one strategy/operation/support mediator.

Choose Partner

As shown in Figure 26, this collaborative strategy process is the sub process of Choose Partner in Figure 25. The sub collaborative strategy process reorganized functions and messages, which are defined in functional table *A1 Choose Partner*.

Send Payment

As shown in Figure 27, this collaborative operation process is the sub process of *send payment* in Figure 25. The sub collaborative operation process reorganized functions and messages, which are defined in the functional table *A6 Send Payment*.

Deliver Product

As shown in Figure 28, this collaborative support process is the sub process of *deliver product* in Figure 25. The sub collaborative operation process reorganized functions and messages, which are defined in functional table *A5 Deliver Product*.

Section 5: Conclusion

The aim of MISE 2.0 is to develop a Mediation Information System, which manages process orchestration, data conversion and service selection in enterprise information systems. To do so, the first problem is to define or elaborate a collaborative business process. This paper presents a methodology to elaborate collaborative business process models. The organizational model, collaborative network model, IDEF0-based functional model and IDEF1 informational model are defined or reused to gather useful and necessary knowledge in a collaborative situation. A metamodel and several groups of transformation rules are defined to transfer these three models into a BPMN-based collaborative process model.

The strong points of the process elaboration methodology are: (1) an organizational model to define objectives and sub-networks; this can be used easily to verify the small group of partners to complete the task; (2) a functional table separated by columns of partners and objectives; this allows each partner to fill in their column independently; (3) an informational model to provide detailed attributes and relations of messages, this will help the semantic selection of web services; and (4) a collaborative process model, elaborated automatically, which can save time and effort to perform repetitive and tedious tasks.

However, any system has its weak points. These are summarized as follows: (1) with regard to the knowledge gathering phase, models are provided by users. This phase

is completed manually. However, some current research works are dealing with this specific point. Event-Driven Architectures are able to provide a technical infrastructure that allows devices, sensors and other services to publish their messages (as events). These events may then be used to feed the situational modeling editor. This would finally be a way to link the Internet of things (devices) with the Internet of knowledge (ontology) to drive the Internet of services (web-services); (2) as regards the gateways in BPMN, the transformations of exclusive, inclusive and parallel gateways are completed. These are enough for BPEL transformation. But all the gateways defined in BPMN have to be covered; (3) as regards events, the start/end events and the start/end message events are used. Thus, all the BPMN-defined events are not covered.

Once the model, metamodel and transformation-rules design has been accomplished, a software tool will be developed to support model building and transformation-rule implementation [30].

The MISE 2.0 abstract level software tool should implement the following three main functions: (i) creation of the organizational model, functional model, informational model and process model (using GWT: Google Web Toolkit [31] and Java 2D graphical design to implement); (ii) transformation from the organizational, functional and informational to the process model (using JDOM, Java or ATL to implement); and (iii) extraction of the BPMN collaborative process cartography (using JDOM and Java to implement).

The whole BPMN collaborative process cartography is provided to MISE 2.0 concrete level. The semantic gap between business services and web services is fixed at concrete level. The BPMN-based collaborative process cartography is transferred to BPEL [32] file and deployed in ESB [33]. The concrete-level global structure is presented in [34].

References

[1] M. Zdravković, H. Panetto, M. Trajanović, and A. Aubry, “An approach for formalising the supply chain operations,” *Enterprise Information Systems*, Vol. 5, No. 4, pp. 401–421, 2011.

[2] S. Li, L. Xu, X. Wang, and J. Wang, “Integration of hybrid wireless networks in cloud services oriented enterprise information systems,” *Enterprise Information Systems*, Vol. 6, No. 2, pp. 165–187, 2012.

[3] G. editors L. Li and J. N. Warfield, “Perspectives on quality coordination and assurance in global supply chains,” *International Journal of Production Research*, Vol. 49, No. 1, pp. 1–4, 2011.

[4] L. Li, “Effects of enterprise technology on supply chain collaboration: analysis of China-linked supply chain,” *Enterprise Information System*, Vol. 6, No. 1, pp. 55–77, 2012.

[5] W. Tan, W. Xu, F. Yang, L. Xu, and C. Jiang, “A framework for service enterprise workflow simulation with multi-agents cooperation,” *Enterprise Information Systems*, Vol. 0, No. 0, pp. 1–20, 2012.

[6] J. Touzi, F. Benaben, H. Pingaud, and J. P. Lorré, “A model-driven approach for collaborative service-oriented architecture design,” *International Journal of Production Economics*, Vol. 121, No. 1, pp. 5–20, Sep. 2009.

[7] IEEE, “IEEE: Standard Computer Dictionary - A Compilation of IEEE Standard Computer Glossaries,” 1990.

[8] S. Truptil, F. Benaben, P. Couget, M. Lauras, V. Chapurlat, and H. Pingaud, “Interoperability of information systems in crisis management: Crisis modeling and metamodeling,” *Enteriprise Interoperability III: New Challenges and Industrial*, pp. 583–594, 2008.

[9] D. Chen, G. Doumeingts, and F. Vernadat, “Architectures for enterprise integration and interoperability: Past, present and future,” *Computers in Industry*, Vol. 59, No. 7, pp. 647–659, Sep. 2008.

[10] G. Wiederhold and M. Genesereth, “The conceptual basis for mediation services,” *IEEE Expert*, Vol. 12, No. 5, pp. 38–47, 1997.

[11] J. Miller, J. Mukerji, and others, “MDA Guide Version 1.0. 1,” *Object Management Group*, Vol. 234, p. 51, 2003.

[12] C. Atkinson and T. Kuhne, “Model-driven development: a metamodeling foundation,” *IEEE Softw.*, Vol. 20, No. 5, pp. 36–41, Sep. 2003.

[13] J. Bézivin, “In search of a basic principle for model driven engineering,” *Novatica Journal, Special Issue*, Vol. 5, No. 2, pp. 21–24, 2004.

[14] F. Bénaben, W. Mu, S. Truptil, H. Pingaud, and J. P. Lorré, “Information Systems design for emerging ecosystems,” in *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, 2010, pp. 310–315.

[15] V. Rajsiri, J.-P. Lorré, F. Bénaben, and H. Pingaud, “Knowledge-based system for collaborative process specification,” *Computers in Industry*, Vol. 61, No. 2, pp. 161–175, Feb. 2010.

[16] F. Benaben, N. Boissel-Dallier, H. Pingaud, and J.-P. Lorre, “Semantic issues in model-driven management of information system interoperability,” *International Journal of Computer Integrated Manufacturing*, Vol. 0, No. 0, pp. 1–12, 2012.

[17] X. V. Wang and X. W. Xu, “DIMP: an interoperable solution for software integration and product data exchange,” *Enterprise Information Systems*, Vol. 6, No. 3, pp. 291–314, 2012.

[18] J. L. Kelley, *General topology*. Springer Verlag, 1975.

[19] D. Neiger, L. Churilov, and A. Flitman, “Business Objectives Modelling,” in *Value-Focused Business Process Engineering : a Systems Approach*, Vol. 19, Boston, MA: Springer, 2009, pp. 1–26.
[20] ISO 9000, “ISO 9000 Quality management,” Sep-2005. [Online]. Available: http://www.iso.org/iso/home/store/publications_and_e-products/publication_item.htm?pid=PUB100224
[21] C. Menzel and R. J. Mayer, “The IDEF Family of Languages,” in *Handbook on Architectures of Information Systems*, 2006, pp. 215–249.
[22] R. J. Mayer, “IDEF1 information modeling,” *Knowledge Based Systems, Inc., College Station, Texas*, 1992.
[23] J. Touzi, “Aide à la conception de Système d’Information Collaboratif support de l’interoperabilité des entreprise,” Ph.D. Thesis (in French), INPT-ENSTIMAC, Albi, France, 2007.
[24] S. Truptil, “Etude de l’approche de l’interopérabilité par médiation dans le cadre d’une dynamique de collaboration appliquée à la gestion de crise,” Ph.D. Thesis (in French), INPT-ENSTIMAC, Albi, France, 2011.
[25] S. White, “Using BPMN to model a BPEL process,” *BPTrends*, Vol. 3, No. 3, pp. 1–18, 2005.
[26] OMG, “Business Process Model and Notation (BPMN) Version 2.0.” 2011.
[27] E. Börger, “Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL,” *Software & Systems Modeling*, Sep. 2011.
[28] R. M. Smullyan, *First-order logic*. Dover Publications, 1995.
[29] G. Valiente, *Algorithms on trees and graphs*. Springer, 2002.
[30] S. Hallé, “Cooperative runtime monitoring,” *Enterprise Information Systems*, Vol. 0, No. 0, pp. 1–29, 2012.
[31] V. Gupta, *Accelerated GWT: Building Enterprise Google Web Toolkit Applications*. Apress, 2008.
[32] S. Lee, T.-Y. Kim, D. Kang, K. Kim, and J. Y. Lee, “Composition of executable business process models by combining business rules and process flows,” *Expert Systems with Applications*, Vol. 33, No. 1, pp. 221–229, Jul. 2007.
[33] D. A. Chappell, *Enterprise Service Bus*. O’Reilly Media, Inc., 2004.
[34] F. Benaben, N. Boissel-Dallier, J.-P. Lorre, and H. Pingaud, “Semantic Reconciliation in Interoperability Management through Model-Driven Approach,” in *Collaborative Networks for a Sustainable World*, Vol. 336, L. Camarinha Matos, X. Boucher, and H. Afsarmanesh, Eds. Berlin: Springer-Verlag Berlin, 2010, pp. 705–712.

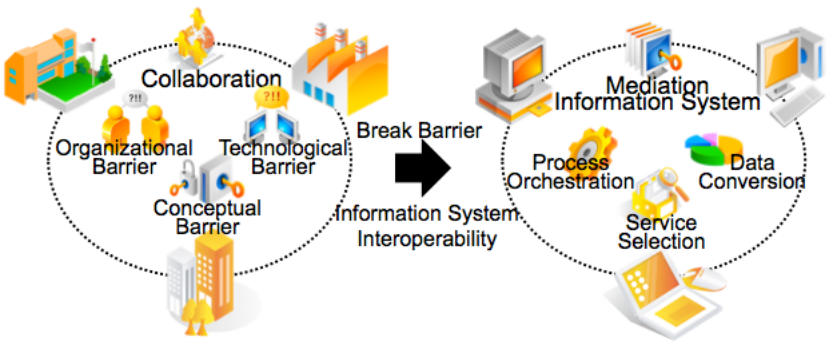

Figure 1. Introduction to Global Situation

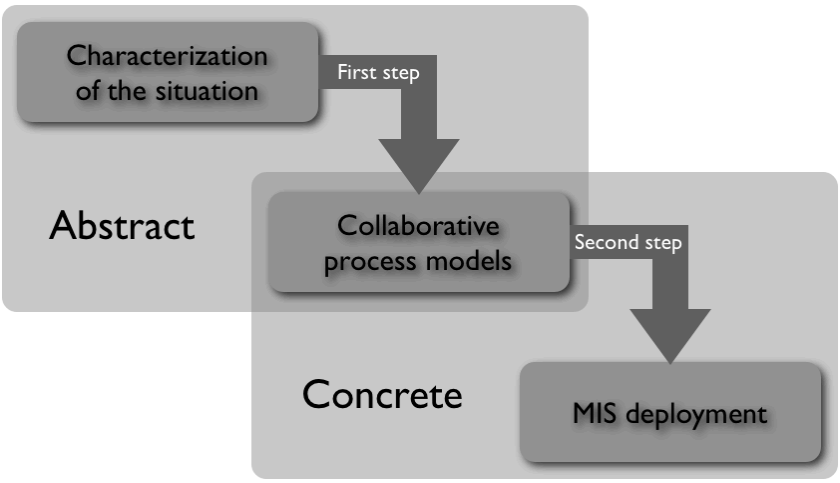

Figure 2. Overview of the MISE approach

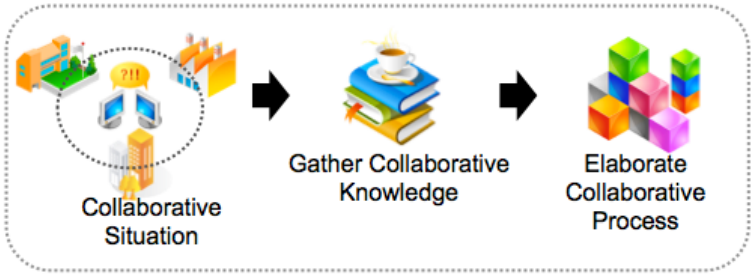

Figure 3. Collaborative business process main elaborating structure

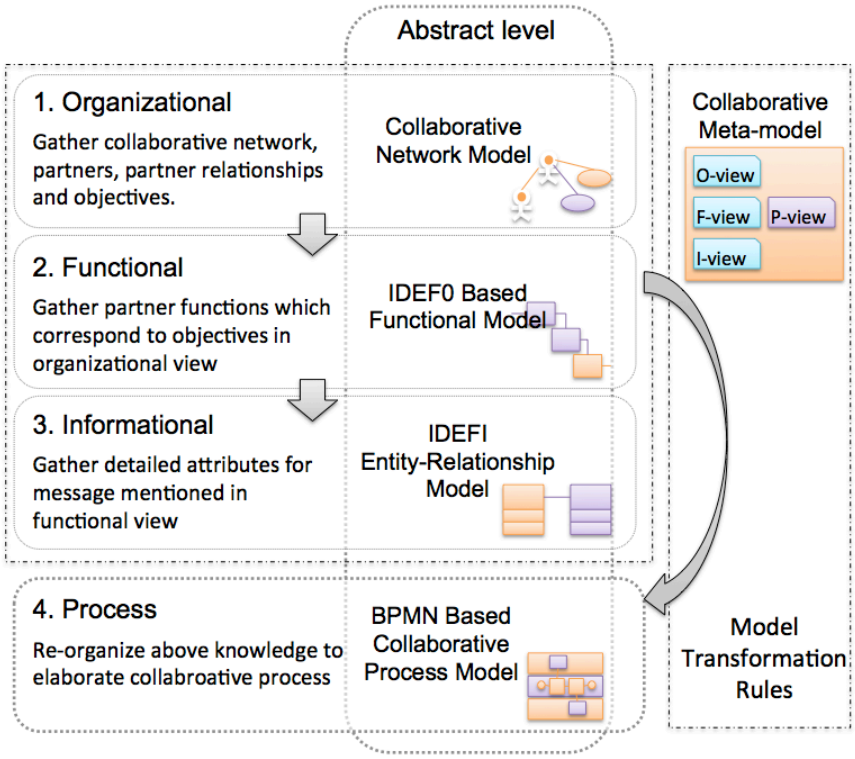

Figure 4. Collaborative process elaboration methodology

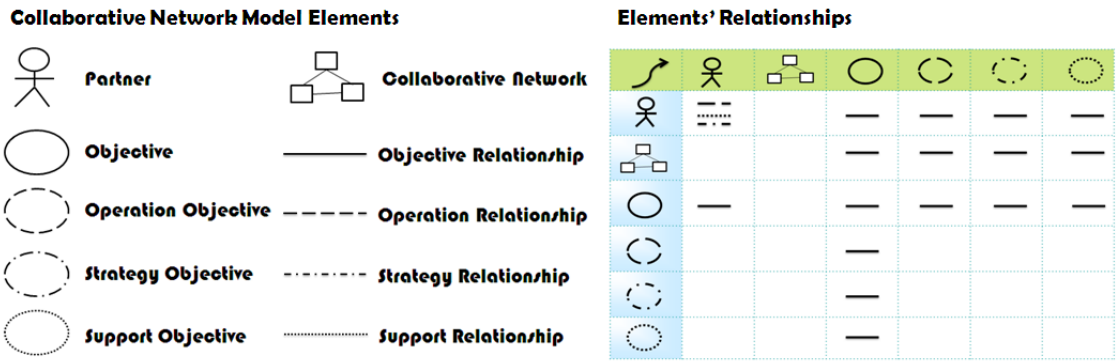

Figure 5. Collaborative network model elements and relations

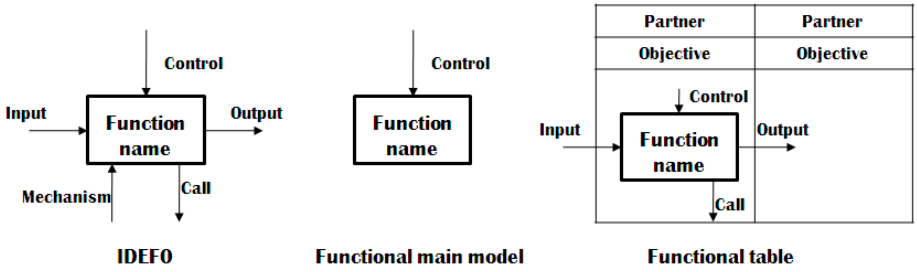

Figure 6. IDEF0 syntax and extensions

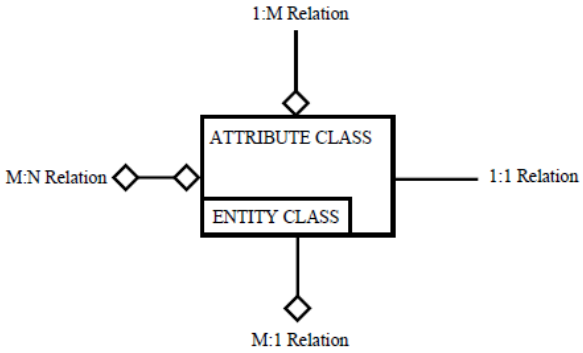

Figure 7. IDEF1 model unit [22]

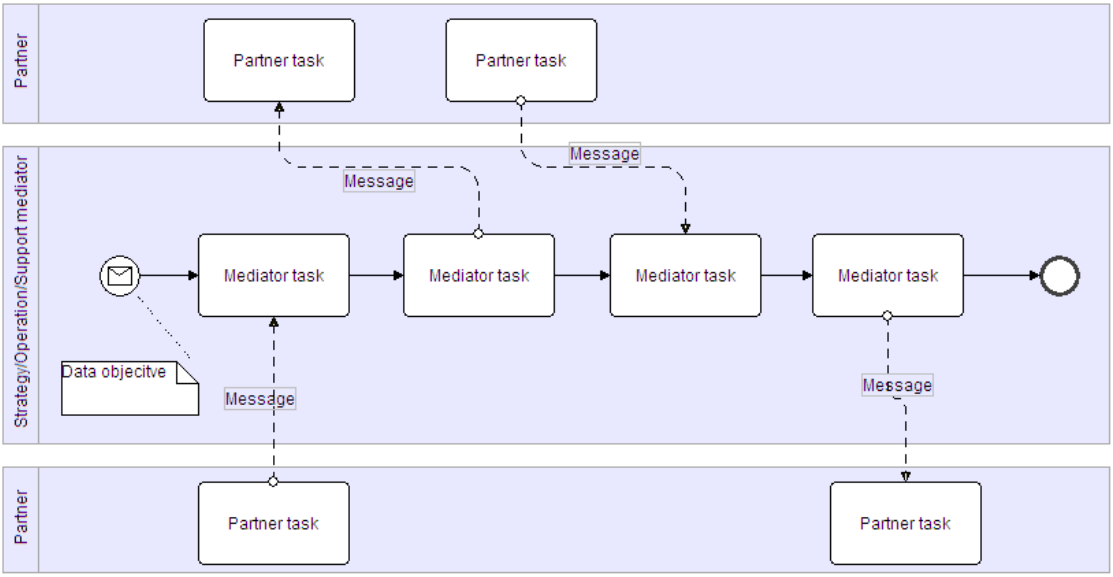

Figure 8. Collaborative Process Model – Simple example

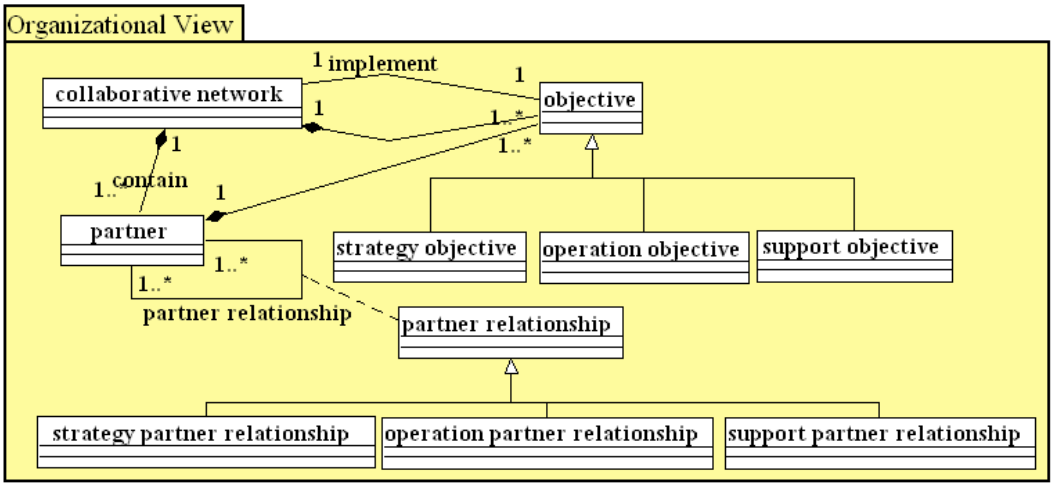

Figure 9. Metamodel – Organizational View

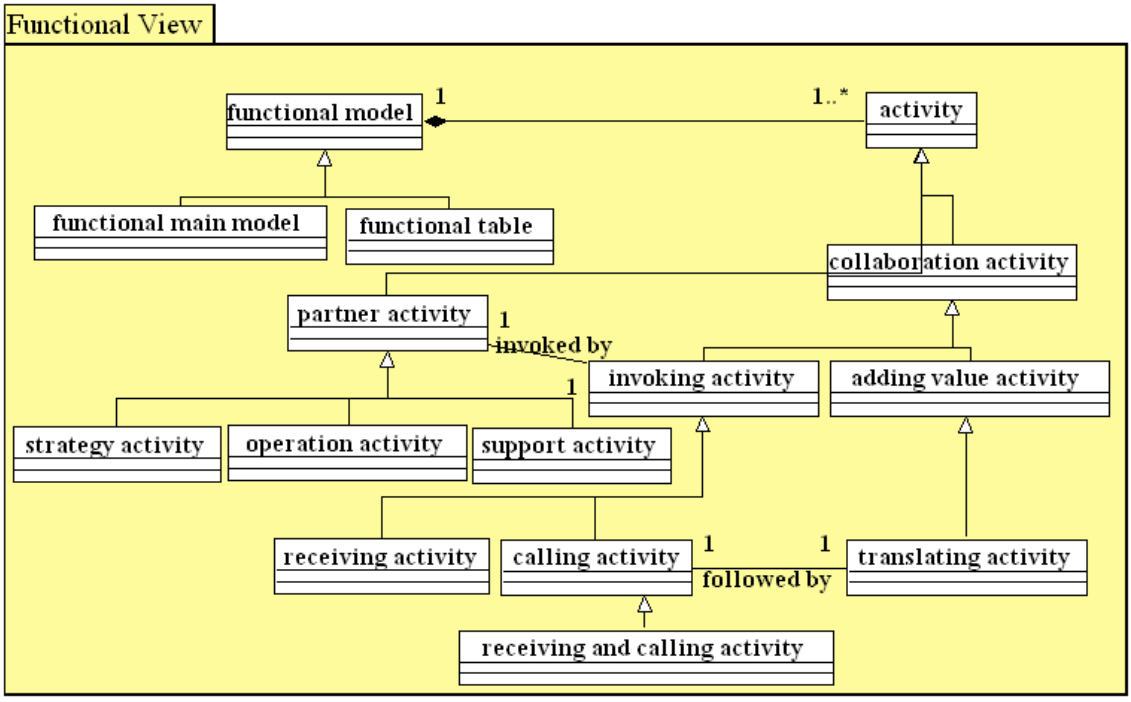

Figure 10. Metamodel – Functional View

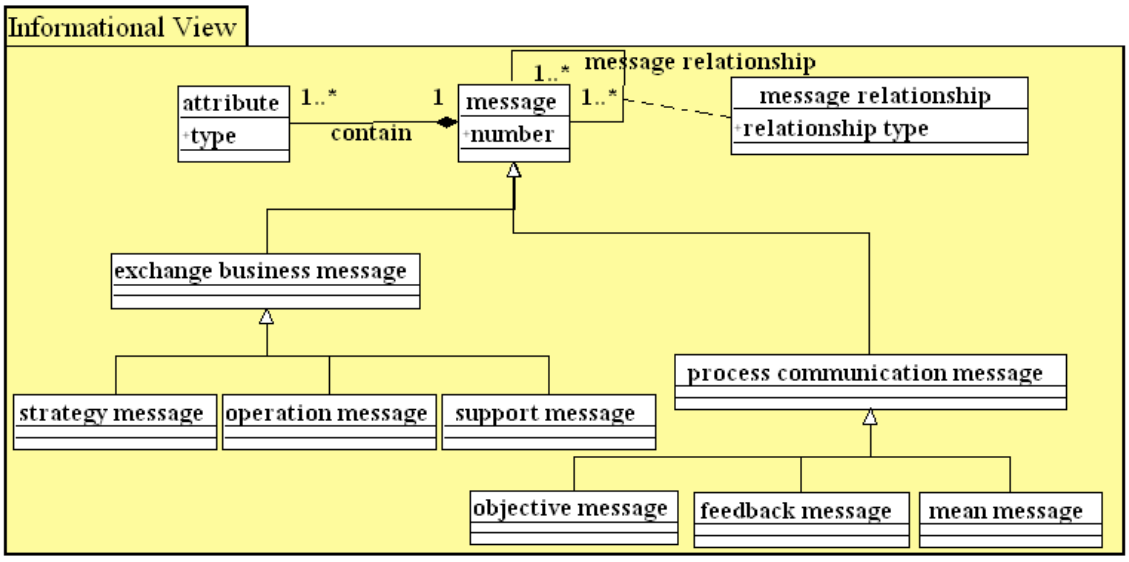

Figure 11. Metamodel – Informational View

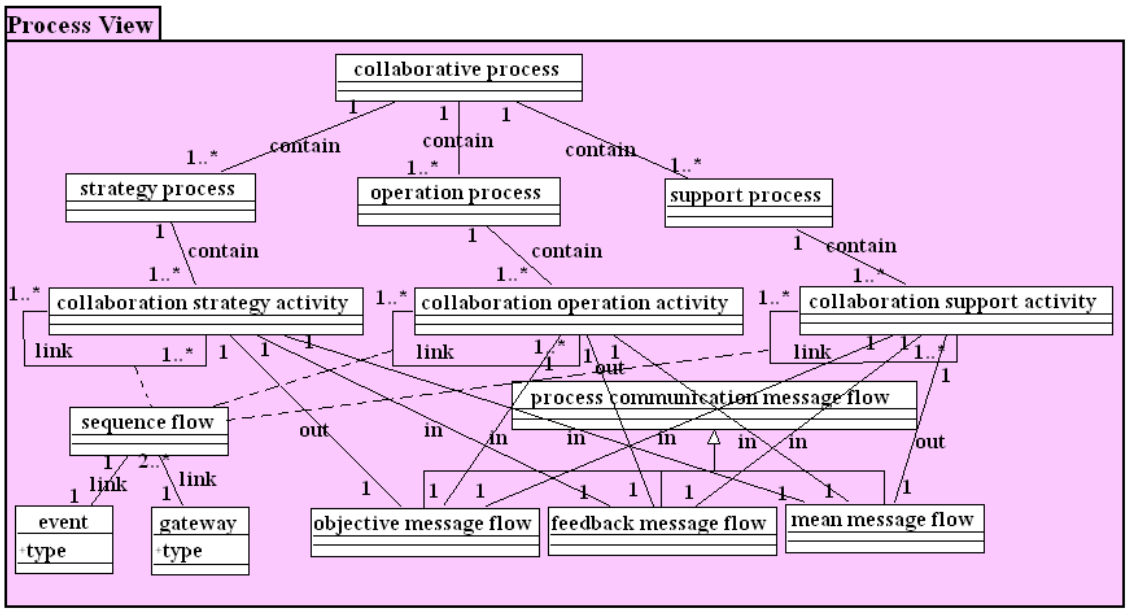

Figure 12. Metamodel – Process View

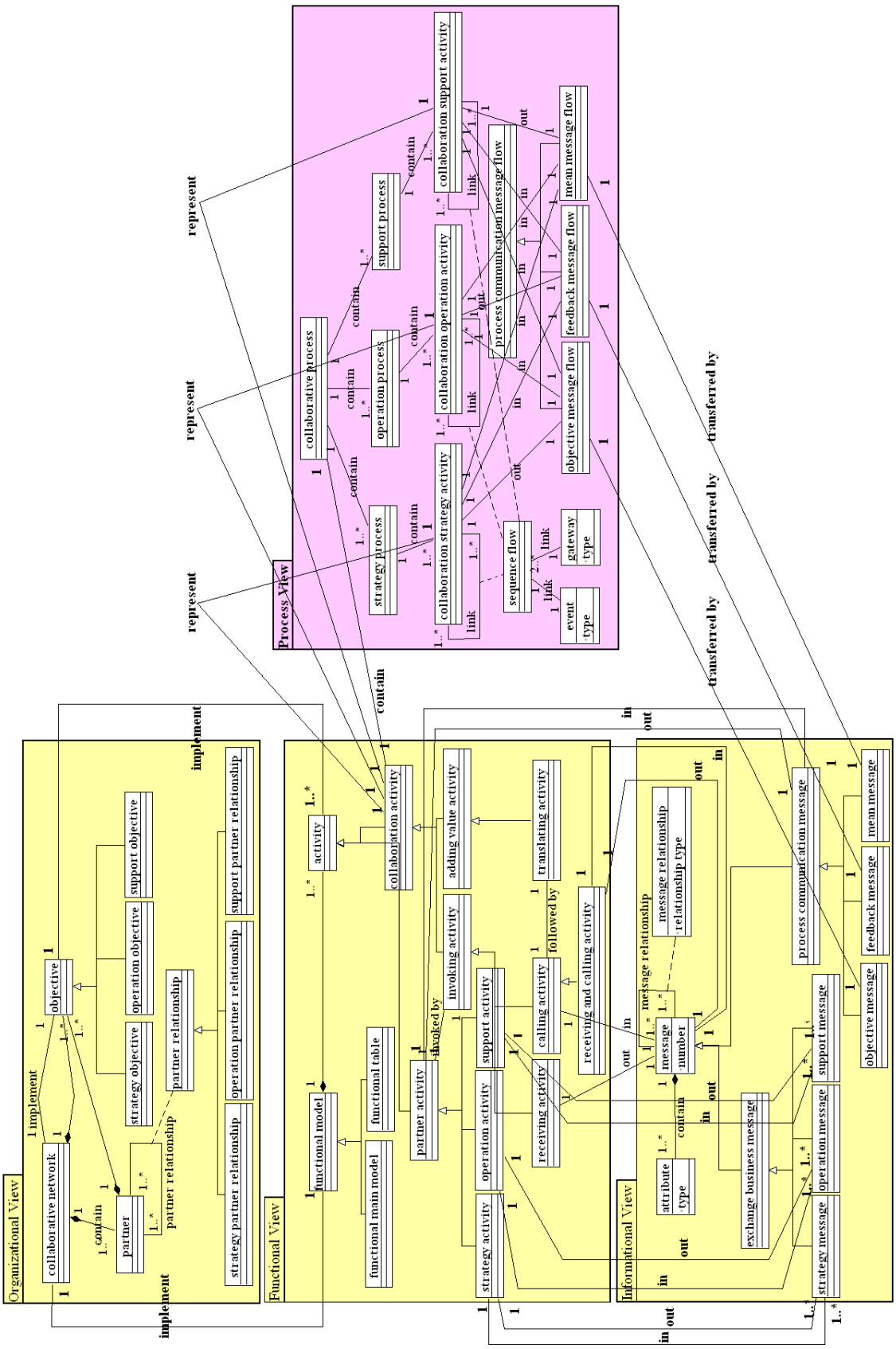

Figure 13. Metamodel – Links among the Organizational, Functional, Informational and Process Views

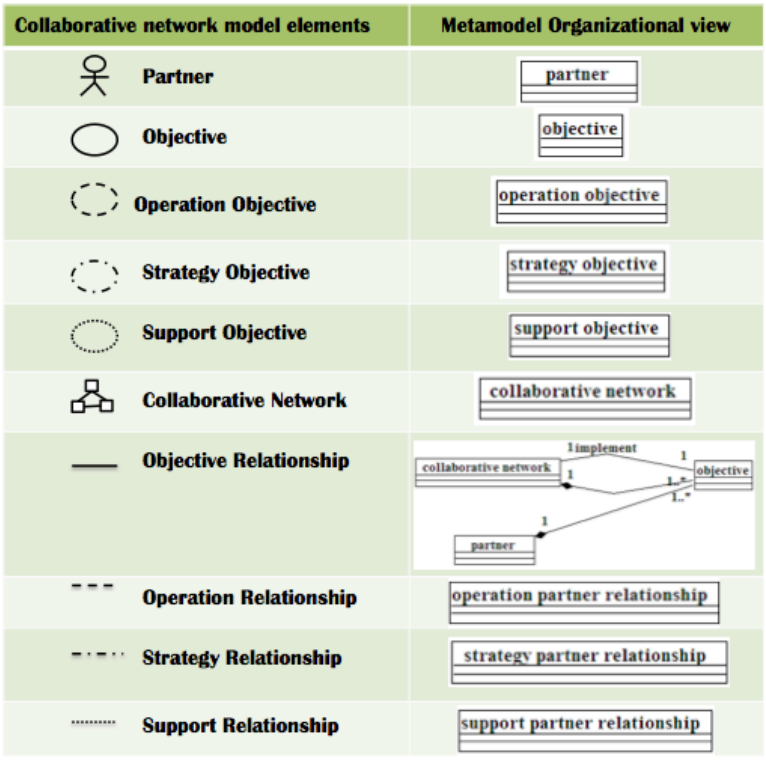

Figure 14. Matching Collaborative Network metamodel concepts to collaborative metamodel Organizational View metamodel concepts

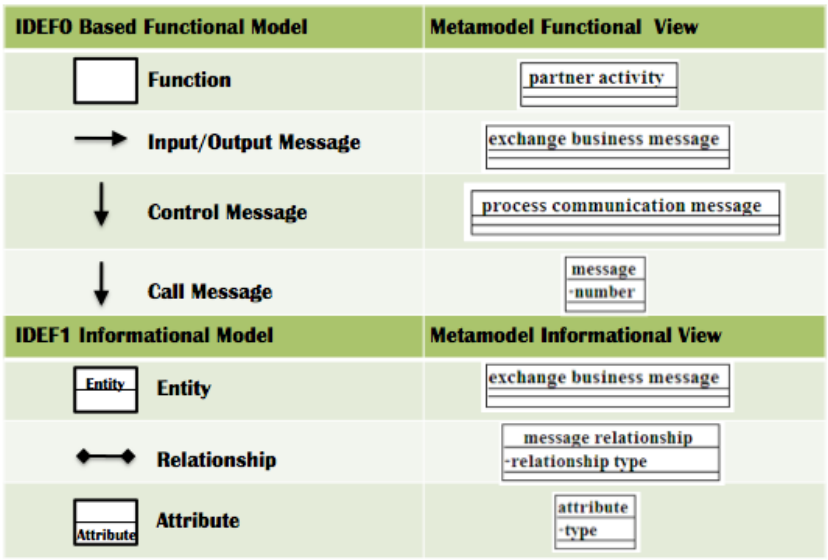

Figure 15. Matching Functional and Informational metamodel concepts to collaborative metamodel Functional View and Informational View metamodel concepts

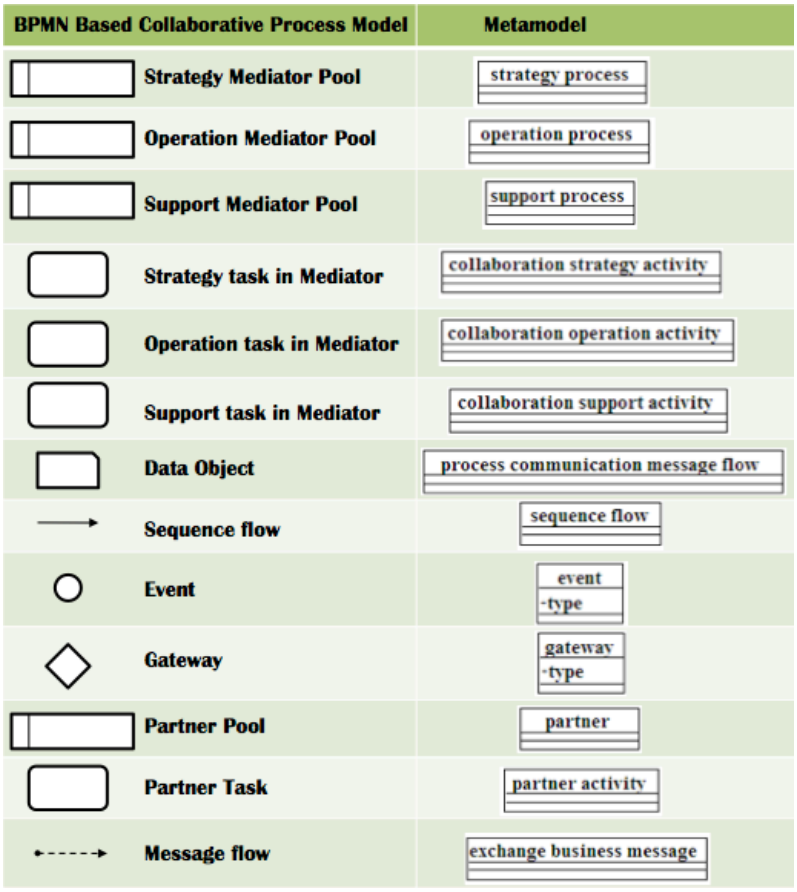

Figure 16. Matching BPMN-based Collaborative Process metamodel concepts to collaborative metamodel Process View metamodel concepts

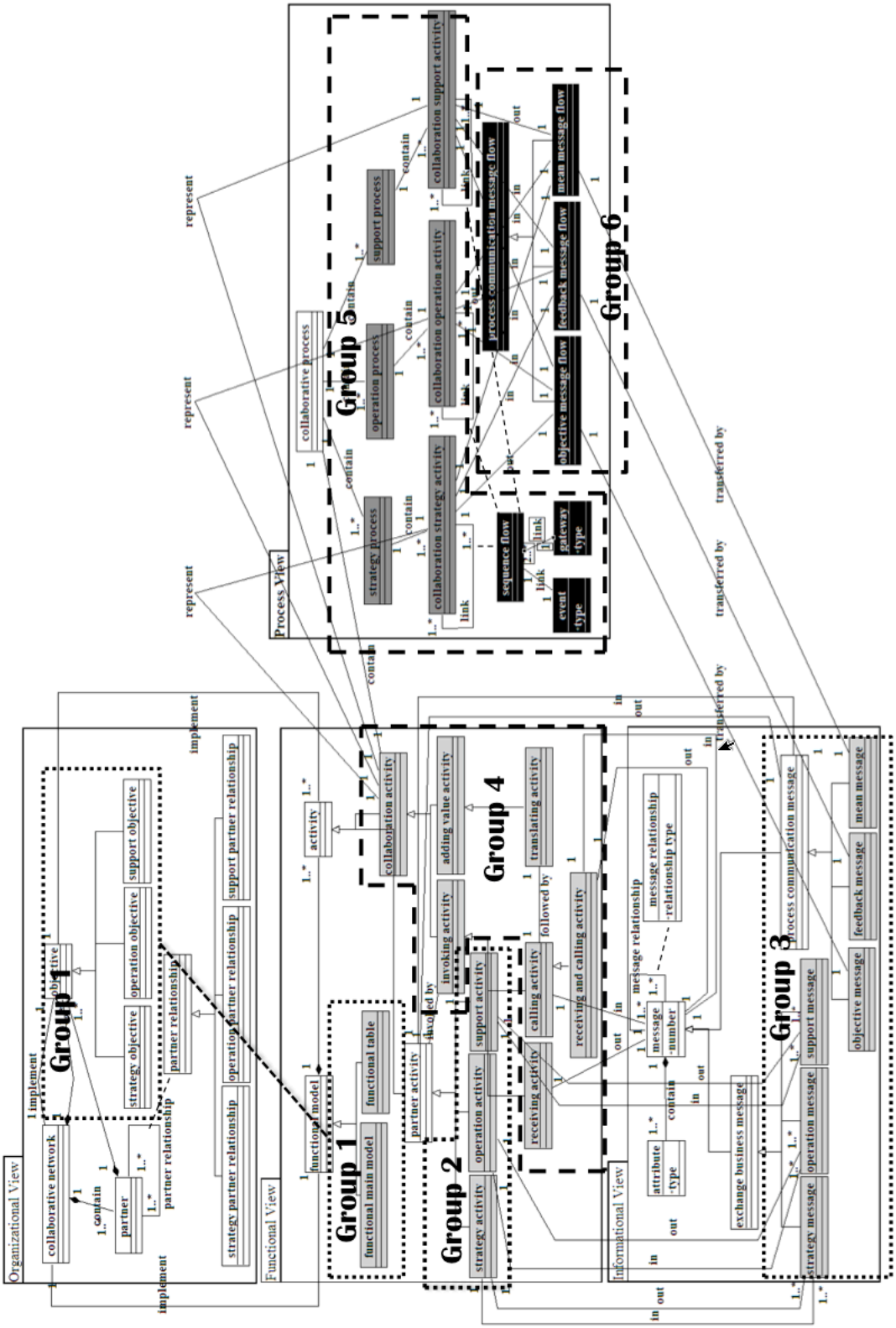

Figure 17. Groups of Matching Rules

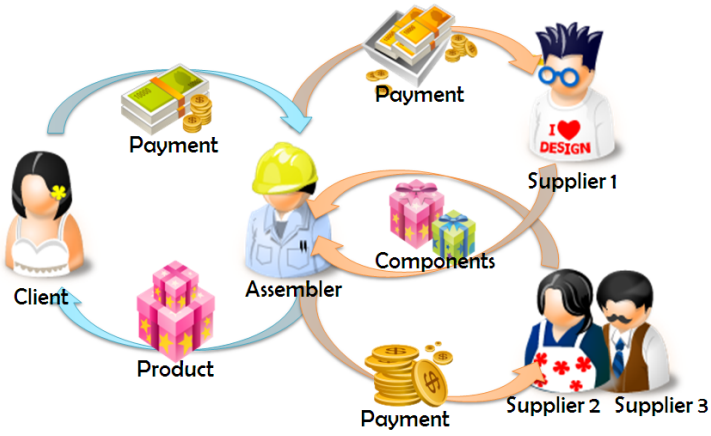

Figure 18. Example Presentation

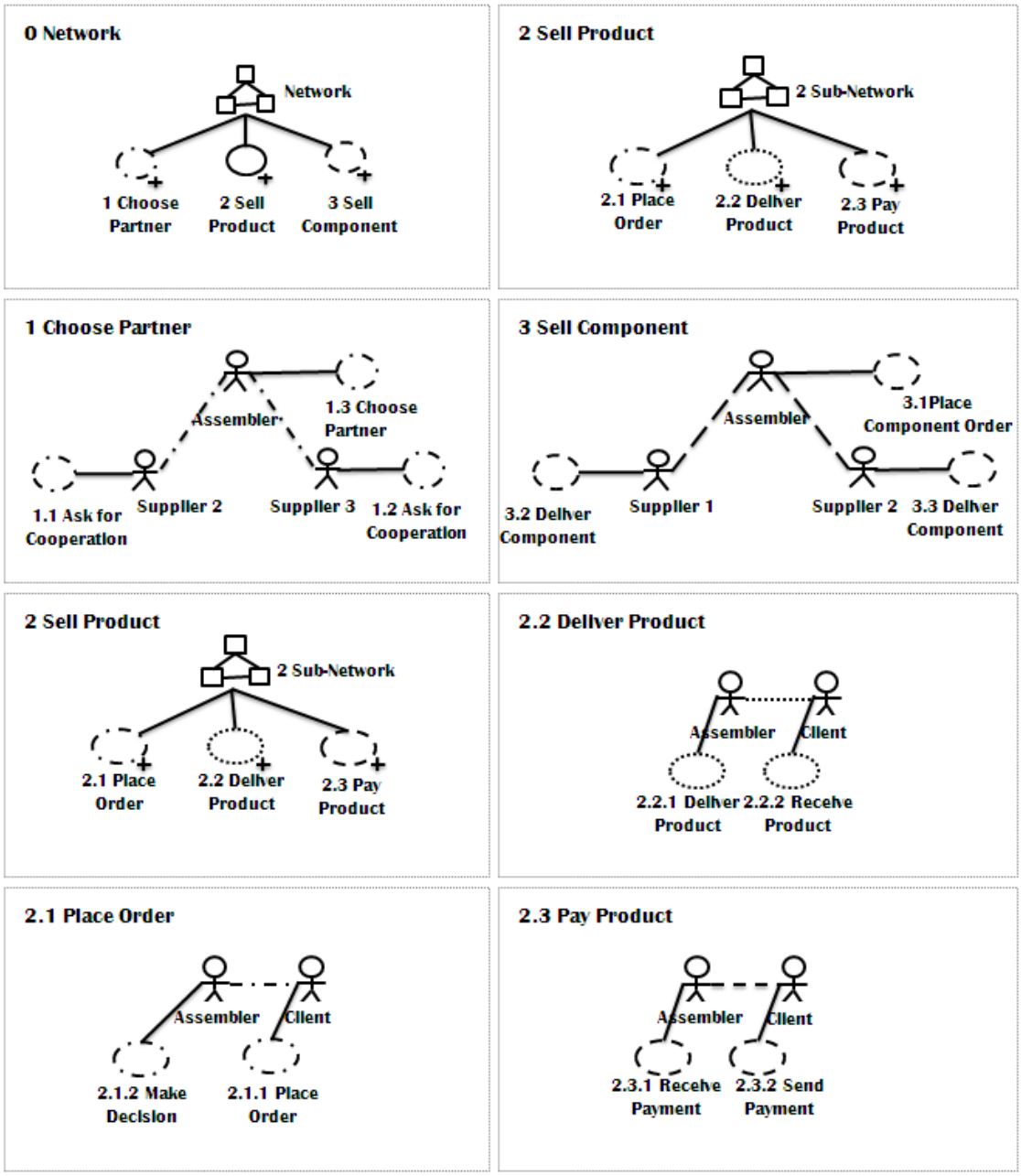

Figure 19. Collaborative network model example

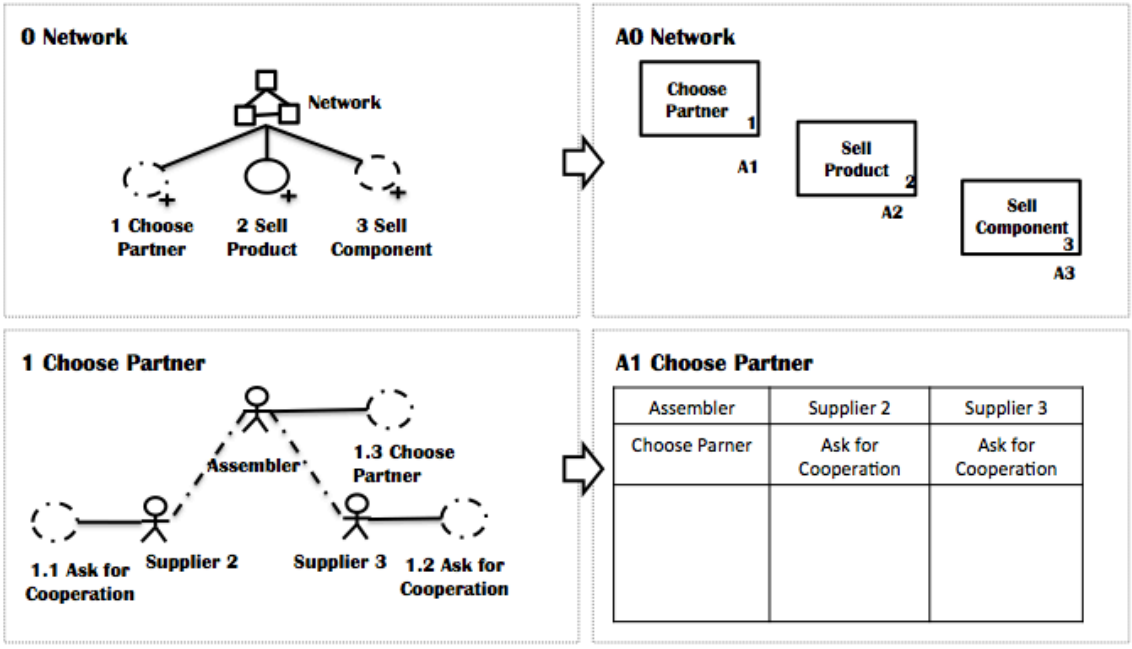

Figure 20. Functional Model Initial Results

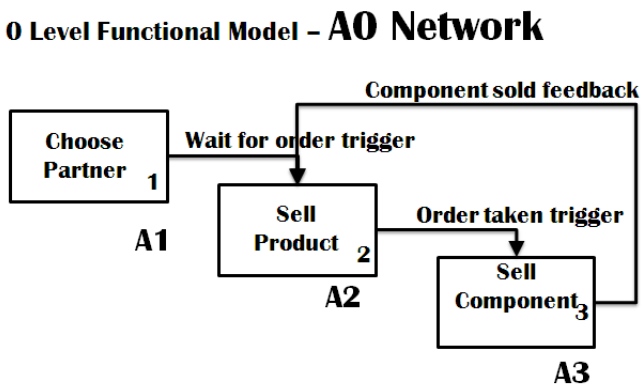

Figure 21. Functional model A0 network

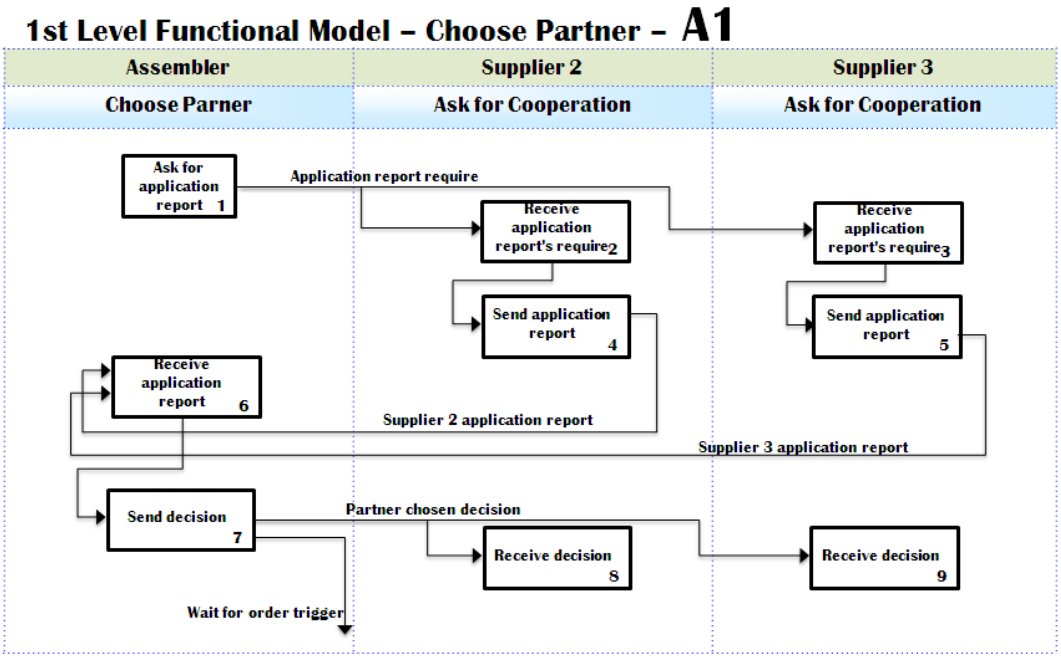

Figure 22. Functional model A1 Choose Partner

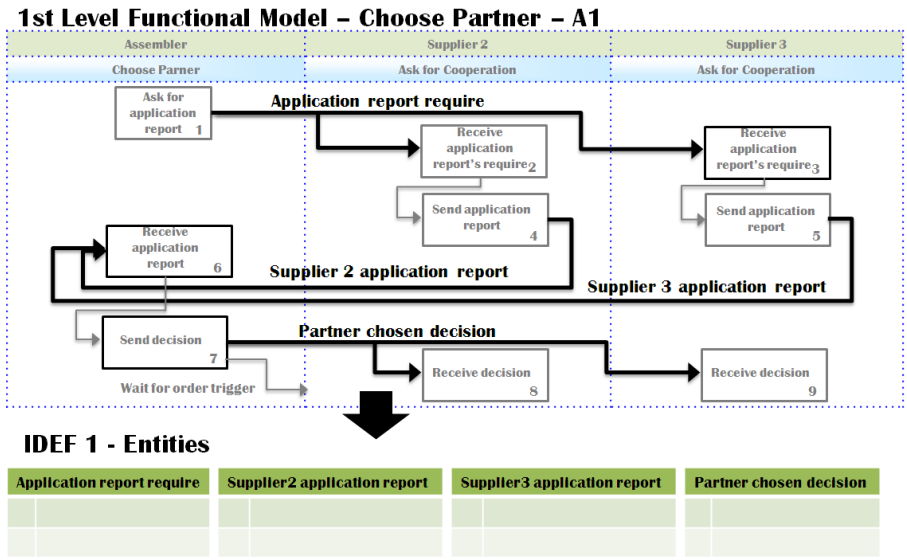

Figure 23. Informational model initial results

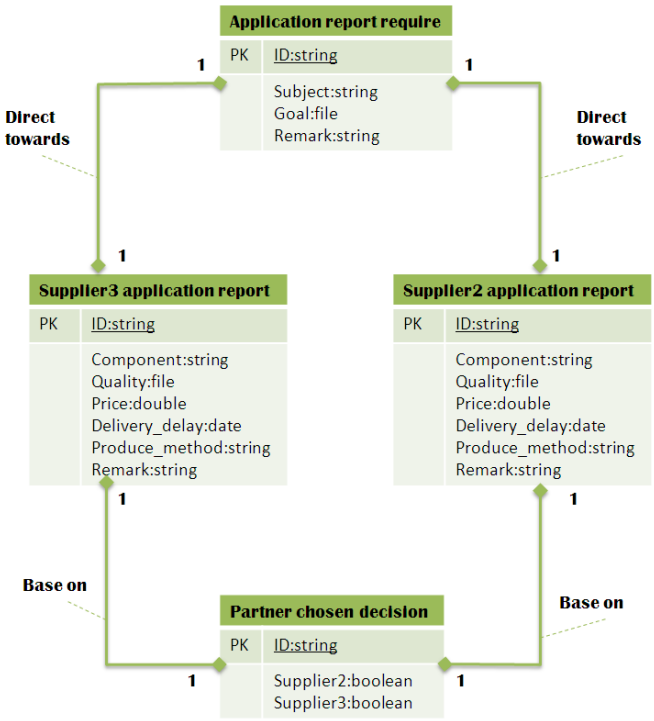

Figure 24. A1 Choose Partner – Informational model

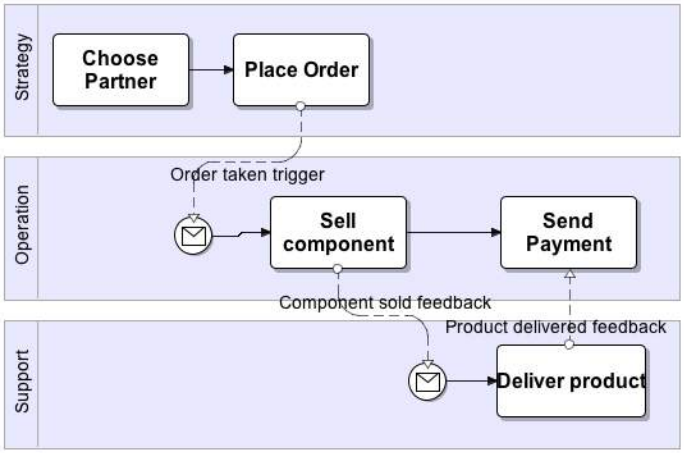

Figure 25. Main collaborative process

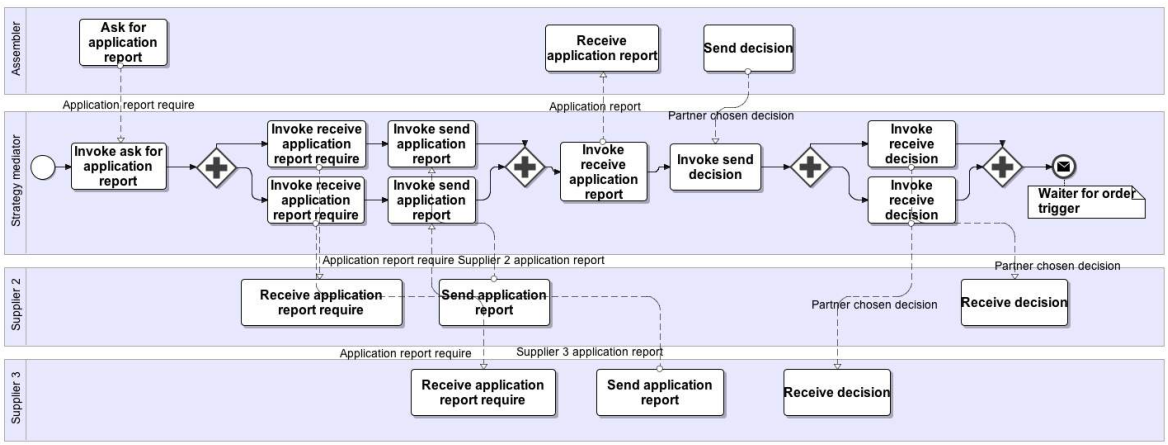

Figure 26. Choose partner – collaborative strategy process

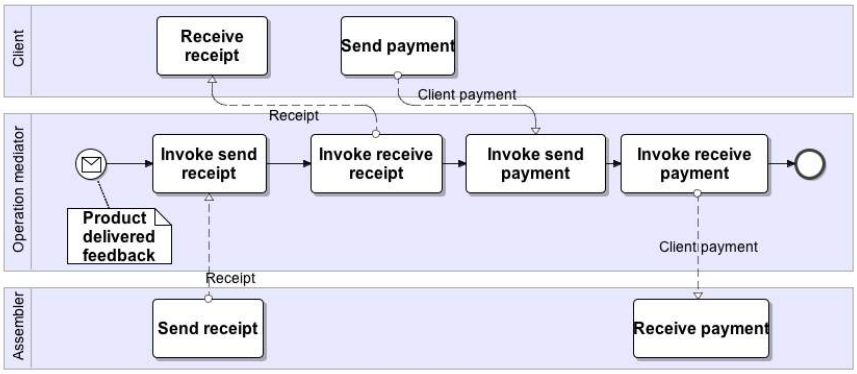

Figure 27. Send payment – collaborative operation process

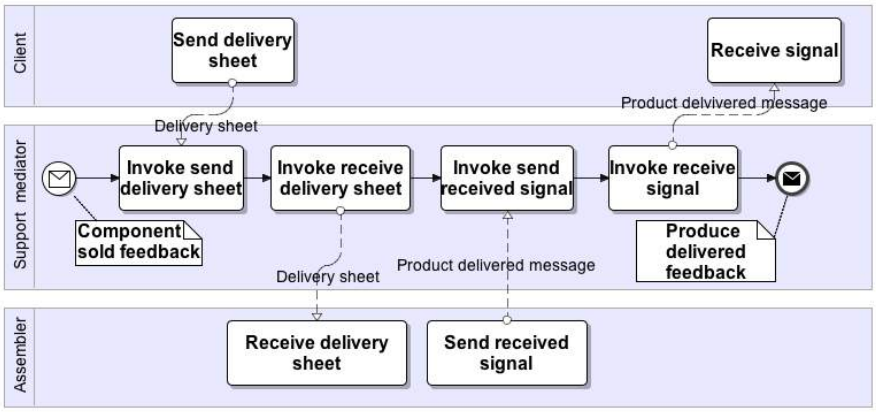

Figure 28. Deliver product – collaborative support process