# An Ontology Based Collaborative Business Service Selection - Contributing to Automatic Building of Collaborative Business Process

Wenxin Mu, Frederick Benaben, Hervé Pingaud

# An Ontology based Collaborative Business Service Selection - Contributing to Automatic Building of Collaborative Business Process

Wenxin Mu[1], Frederick Benaben[2] and Herve Pingaud[3],

[1] Beijing Jiaotong University, Shangyuancun 3, Handian, Beijing, China
[2] Toulouse University – Mines Albi, Allee de science, Albi, France
[3] Champollion University, Place de verdun, Albi, France
wxmu@bjtu.edu.cn, frederick.benaben@mines-albi.fr, herve.pingaud@univ-jfc.fr

**Abstract.** With the world wide inter-enterprise collaboration and interoperability background, automatic collaborative business process deduction might be seen as a crucial researching subject. We design a methodology of deducing collaborative process by only collecting collaborative objectives and partners' business services. The two key problems are: i) selecting corresponding business services for a set of collaborative objectives; ii) ordering business services with serializations and parallelization. This paper aims to present solution of business service selection. In order to solve the problem, we defined a collaborative ontology, which contains numerous instances of business services and processes of MIT process handbook. The collaborative ontology contains essential concepts in collaborative situation, and owns process deducing rules and algorithms. We provide a brief illustration of implementation within a SaaS toolkit called Mediator Modeling 2ool.

**Keywords:** business process management; model-driven engineering; inter-enterprise collaboration; ontology

## 1    Introduction

In collaborative situation, all the partners come with collaborative objectives and their own objectives to achieve and business services to share. They expect to combine their own business services with suitable ones from other partners to work towards their common objectives. In addition, collaborative business process is a combination of business functions, which is inter-linked and filled with sequences and orders. With these needs, *objective-oriented business service selection* and *collaborative business process creation* are absolute essentials in collaboration world. Considering self-updating and re-building of collaborative business process, we shall design an automatic way to deal with service selection and process creation in design level. Further more in implementation level, first because the software tool deals with a collaborative situation, all the partners may use the software in the same time or individually. Secondly, in order to interact with other software tools (which is developed in our lab), the software should be able to deploy in ESB (Enterprise Service Bus). These lead that the software tool involved in the methodology should be a web service. SaaSs (Software as a Service) seems to be a good solution.

In our lab, Vatcharaphun Rajsiri has created a knowledge-based system for collaborative process specification [1]. This system deduces a BPMN (Business Process Modeling Notation) based collaborative process model automatically with the help of collaborative objective model and MIT process handbook [2]. But this system has weaknesses. First, the system collects only the main goal of whole collaborative network. It leads to lack of partners' objectives and sub-network information. Designing a model, which models all above information, is necessary. Second, the deduced BPMN collaborative process covers only operational level. If in a complex collaborative situation, partners come from different departments and management level. An operational collaborative business process couldn't satisfy partners. According to [3] and [4], business process covers strategy, operation and support levels. We arrive at conclusion that target collaborative process should contain strategy, operation and support levels.

A model-driven and ontology based methodology, which take collaborative objectives, partners' objectives and business services as input and deduce collaborative business process as output as automatically as possible, seems to be a good solution in this situation.

For the input, we define objective model and function model to collect basic collaborative knowledge from partners. For the deduction method, collaborative ontology and a set of algorithms and transformation rules is defined. The algorithms manage to link business objectives with business services. The transformations rules help to deduce sequences among business functions. For the output, BPMN based collaborative process cartography is deduced. The collaborative process cartography has three types: strategy, support and operation.

In this paper, we focus on introducing solution of business service selection (dash-line box in Figure 1). Section II first presents definition of objective model and function model. Then it provides a simple example to explain the directions for use of models. Section III provides a brief introduction of collaborative ontology. Section IV gives definitions of business services selection algorithms. Section V is a sketch for collaborative process model and collaborative process creation method. Section VI draws some concluding remarks, discusses the feasibility of our work and outlines our future investigations.

## 2    Objective Model and Function Model

This section explains the input part of Collaborative Business Process Deduction methodology (CBPD). In Section II-A, we first address the definition of objective modeling elements. Section II-B presents the definition of function model. And finally Section II-C illustrates examples of objective model and function model.

### 2.1    Objective Model

"*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.*" [5] For our individual

needs, the objective model here should collect both collaborative main goals and partners' individual objectives. For each collaborative goal, partners are regrouped as sub-collaborative network. Partners also have their own objectives. We come up with a result: a real collaborative situation is like a multi-level pyramid: each level could be decomposed to sub-network until partner, and each level could be abstracted to higher-level collaborative network until the whole collaborative network. With supplementary illustration, because the goal of CBPD is to deduce collaborative process cartography (including strategy, operation and support sub-collaborative processes), we consider that the objectives collected in objective model may set to three types: strategy/operation/support objective.

To summarize, the objective model presented here should be able to (Figure 1 presents modeling elements and links):

- Model collaborative network, collaborative sub-network and partners' relationships.
- Model collaborative networks' objective, sub network's objectives and partners' objectives.

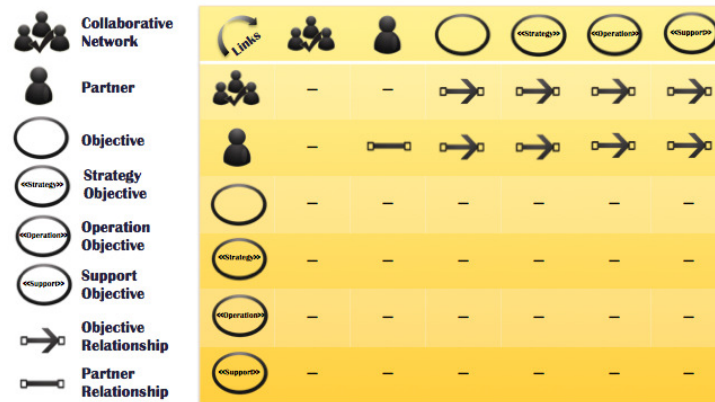Classify objective into strategy, operation and support objective.



**Fig. 1 -** Objective Model Elements Definition

## 2.2  Function Model

The requirements for the function model are to get partners' functions, to simplify user's modeling tasks and to decrease user's workload. The function model just collects functions that partners want to share and which can be published to other partners. We defined an IDEF1 (Integration Definition for Function Modeling) based function model to gather partners' business services.

Standard function model: IDEF0 [6] is reused to present partners' functions. The standard IDEF0 modeling unit is shown in Figure 2 left part. The modeling unit has input and output message. Controlling message and mechanism controls and support function. The function also could send call message to invoke another function. As shown in the middle of Figure 2, function main model reuses function unit and

controlling message. As shown on the right of Figure 2, partner columns separate function model. Partner lists its shared functions in correspondence column. Function model reuses function unit, input and output message, controlling message and call message of standard IDEF0 modeling unit.
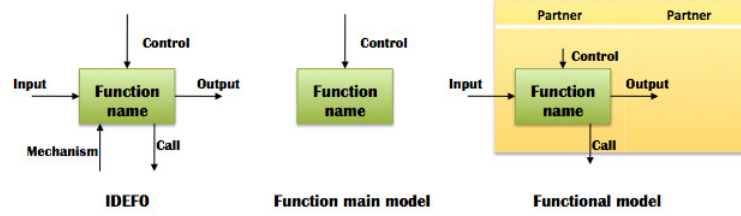


**Fig. 2 -** Function Model Elements Definition [7]

After analysis and evaluation, we found out that the function main model could be partially transferred from objective model. Objectives of collaborative network could be seen as main function. User only fulfills control messages among abstract functions. Transformation equations from objective model to main function model are defined in first-order logic [8].

Due to particularity of transformation rules, first order logic still needs to be expended as followed : i) Element: *X* is *Collaborative Network* → *CollaborativeNetwork(X)*; ii) Relationship: *Y* is *Objective Relationship* which is between *CollaborativeNetwork* $X_1$ and *Objective* $X_2$ → *Objective-Relationship(Y)(CollaborativeNetwork($X_1$), Objective($X_2$)).*

$$\forall \text{CollaborativeNetwork}(X) \ (\forall \text{ObjectiveRelationship}(\text{CollaborativeNetwork}(X),$$
$$\text{Objective}(X_1))) \rightarrow \exists \text{MainFunctionModel}(X) \ \wedge$$
$$\exists \text{MainFunction}(X_1) \in \text{MainFunctionModel}(X) \hspace{2cm} \Box 1 \Box$$

## 3     Collaborative Ontology

The collaborative ontology defined in CBPD aims to support business service selection and collaborative process creation. This ontology defines the concepts and relationships involved in collaborative situation. The collaborative ontology must be fulfilled with instances from different business domains (for example, MIT process handbook from manufacture, Score Model from Supply Chain and instances from crisis management).
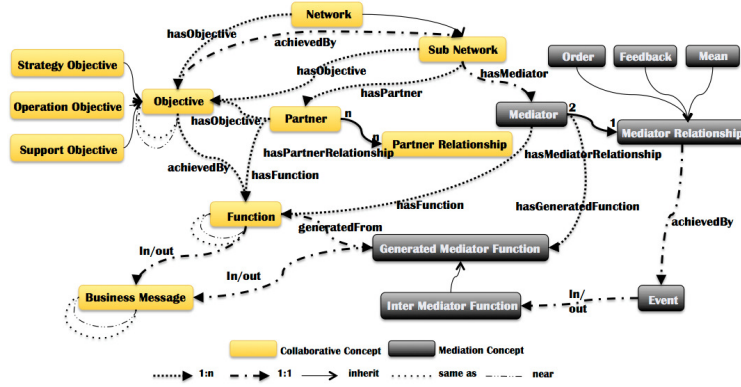
**Fig. 3** - Collaborative Ontology

As explained in Section II, the collaborative concepts part of collaborative ontology is seen as two relationships and three main concepts: *Same As/Near By* relationship, *Network*, *Objective* and *Function*. They are detail explained as followed:

- *Same As/Near By* relationship contains two parts: *Same As* and *Near By*. If concept A is Same As with concept B, then concept A equals to concept B. Concept A and Concept B presents the same concept. If concept A is *Near By* concept B, then concept A and concept B are similar. Concept A may present part of concept B. Concept A and concept B may crosse.

- *Network* means all the involved partners and relationships among them. Another close concept is *Sub Network*.

- *Objective* presents business goals of collaboration and partners. *Objective* has relationship: *Same As/Near By* with another *Objective*.

- *Function* equals to business services or function, which are provided by partners. *Function* may own input and output *Message*. Both *Function* and *Message* have self-related relationships: *Same As/Near By*.

In collaborative ontology, collaborative concepts part serves to match collaborative objectives to partners' business capabilities. Mediation concepts part is based on Mediation [9] concept. In this paper, our vision is on business service selection. So process creation part will not be detailed.

## 4    Business Service Selection Method

In this section, we provide the explanation of business service selection. Section IV-A presents basic theory of business service selection. The example in Section II-C is re-used. Section IV-B provides official algorithms of selecting business service.

### 4.1    Business Service Selection Principal

The principal of business service selection in CBPD is based on collaborative ontology, which has been introduced in last section. In the ontology, there are large numbers of *Objective* and *Function* instances with relationship: *achievedBy*. If we could link business objectives and business function in the model to *Objective* and *Function* instances in collaborative ontology, then we could indirectly link business objective to business function by the help of relationship: *achievedBy*. With above theory, we can complete business service selection task.

As shown in Figure 4, there are two parts: ontology and model. In ontology part, we choose several instances of *Function* and *Objective* in ontology. In model part, we take the objective and function model example in Section II-C. Here for each business objective and function in the model, we want to find same or close instances in ontology. For example, in the model, "Book van & driver" is same or close to "Book transportation" in the ontology, then we make a link: *Same As/Near By* between them.
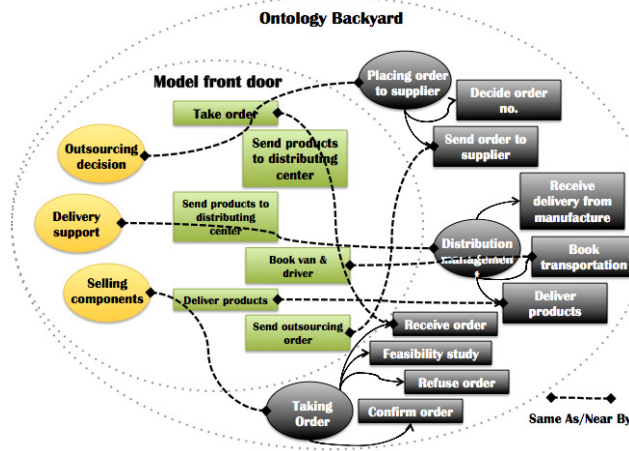


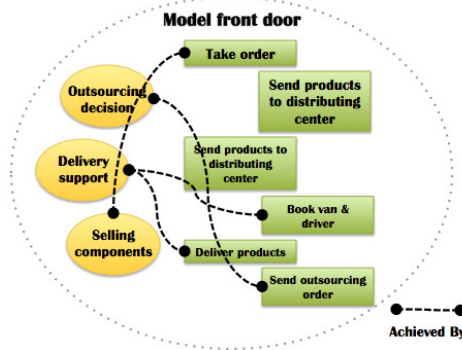**Fig. 4 -** Making Links among Models and Ontology Instances



**Fig. 5 -** Deducing Links among Objectives and Functions

With all the relationship: *Same As/Near By* in Figure 4, suitable business functions are selected for business objectives. Figure 5 shows all the results. Business objectives is linked to business functions by relationship: *achievedBy*.

Even though the basic principal of business service selection is defined, there are still some remarks to consider:

- Making relationship: *Same As/Near By* for each business objective and function is quite hard for user. So we provide an Instance Suggestion Mechanism, which could provide suggested ontology instance for user. Section IV-B algorithm (1) presents the Instance Suggestion Mechanism algorithm.
- Business objectives and functions defined in the model could also be seen as *Business* and *Function* instances in collaborative ontology for future uses. There should be a self-update mechanism to enlarge the collaborative ontology. Section IV-B algorithm (3) explains self-updating mechanism.

### 4.2    Business Service Selection Algorithms

**1) Instance Suggestion Algorithm.** Instance suggestion algorithm deals with selecting same or nearest ontology Objective or Function instances for each business objective and function. Algorithm(1) takes keyword of business objective as input, uses collaborative ontology as data and provides a list of suggested ontology instances. This algorithm has three main parts:

- Line 3: finding an *Objective* instance in collaborative ontology which owns the same keyword: $objective_{key}$ as business objective, the instance is added to suggestion list: $L_{suggestion}$;
- Line 6-Line 17: Taking frontal parts of keyword as a new list of keyword: $L_{word}[1]$ to $L_{word}[i]$ (for example, keyword: "send products to distributing center", new keywords: "send products to distributing" and "send products to"), for each new keyword, if finding an *Objective* instance's keyword in collaborative ontology which starts with or contains the new one or contains, then the *Objective* instance is added to suggestion list;

Line 18-Line 23: Taking related two words, which are contained in keyword as a new list of keyword: $L_{2words}$ (for example, keyword: "send outsourcing order", new keywords: "send outsourcing" and "outsourcing order"), for each related two words, if finding an *Objective* instance's keyword, which contains the two words, then the Objective instance is added to suggestion list.

---

**Algorithm (1) Instance Suggestion**: provide suggested collaborative ontology instances for business objective.

---

**Input:** $objective_{key}$, keyword for the business objective
**Data:** Collaborative Ontology: CO
**Output:** $L_{suggestion}$, list of suggested ontology instances
1  $A_{objective}$, array of all Objective instances of CO;
2  $L_{suggestion} \leftarrow$ Null;
3  **if** $A_{objective}$ contains objective.keyword = $Objective_{key}$ **then**
4         $L_{suggestion}$ adds objective;

---

```
 5 else
 6       Lword, list of words contained in objectivekey;
 7       i, counter for loop ← Lword.length;
 8       word, store a part of keyword ← Null;
 9       for i from Lword.length to 3 //Check first three words
10            word = from Lword[1] to Lword[i] ;
11            if Aobjective contains objective.keyword starts with word then
12                 Lsuggestion adds objective;
13            end;
14            if Aobjective contains objective.keyword contains word then
15                 Lsuggestion adds objective;
16            end;
17       end;
18       L2words, list of 2 words contained in Objectivekey;
19       while L2words has next element: word do
20            if Aobjective contains objective.keyword contains word then
21                 Lsuggestion adds objective;
22            end;
23       end;
24 end;
25 return Lsuggestion;
```

**2) Objective-Function Mapping.** Objective-Function mapping algorithm is the main part of buiness service selection. The principal has been explained in Section IV-A. As shown in Algorithm (2), it takes list of business objectives and list of business functions as input, uses collaborative ontology as data, and outputs list of relationships: achievedBy. The algorithm is explained as followed:

- Line 3-Line 5: starts the mapping from business functions side. If one business function: $E_{function}$ owns relationship: *Same As/Near By* with one ontology *Function* instance: $O_{function}$.
- Line 6-Line 7: if $O_{function}$ owns relationship: *achievedBy* with one ontology *Objective* instance: $O_{objective}$, and if $O_{objective}$ owns relationship: *Same As/Near By* with business objective: $E_{objective}$, then as result: $E_{objective}$ has relationship: *achievedBy* with $E_{function}$.
- Line 8: the relationship is added into the list: $L_{achievedBy}$.
- Line 13 and Line 16: if there is an $E_{function}$, which doesn't find $E_{objective}$, then a relationship: *achievedBy* from Null to $E_{function}$ is created. The relationship is added to $L_{achievedBy}$.
- Line 21: if an $E_{objective}$ is never achieved, then a relationship: *achievedBy* from $E_{objective}$ to Null is created. The relationship is added to $L_{achievedBy}$.

---

**Algorithm (2) Objective-Function mapping**: find correspondence business functions for each business objective and create relationship: *achievedBy*.

---

**Input:** $L_{objective}$, list of business objectives
            $L_{function}$, list of business functions
**Data:** Collaborative Ontology: CO
**Output:** $L_{achievedBy}$, list of relationship
1 $L_{achievedBy}$ ← Null;

**2** $L_{relatedObjectives}$, list of objectives with achievedBy $\leftarrow$ Null;
**3 while** $L_{function}$ has next element: $E_{function}$ **do**
**4**     **if** $E_{function}$.sameas/nearby!=null **then**
**5**         $O_{function} = E_{function}$.sameas/nearby;
**6**         $O_{objective} = O_{function}$.achievedby;
**7**         **if** $L_{objective}$ contains element $E_{objective}$.sameas/nearby = $O_{objective}$ **then**
**8**             $L_{achievedBy}$ adds achievedBy($E_{objective}$, $E_{function}$);
**9**             **if** $L_{relatedObjective}$ doesn't contain $E_{objective}$ **then**
**10**                 $L_{relatedObjective}$ adds $E_{objective}$;
**11**             **end**;
**12**         **else**
**13**             $L_{achievedBy}$ adds achievedBy(null, $E_{function}$);
**14**         **end**;
**15**     **else**
**16**         $L_{achievedBy}$ adds achievedBy(null, $E_{function}$);
**17**     **end**;
**18end**;
**19while** $L_{objective}$ has next element: $E_{objective}$ **do**
**20**     **if** $L_{relatedObjective}$ doesn't contain $E_{objective}$ **then**
**21**         $L_{achievedBy}$ adds achievedBy($E_{objective}$, null);
**22**     **end**;
**23end**;
**24return** $L_{achievedBy}$;

**3) Ontology Updating.** Ontology updating algorithm deals with inserting business objectives and functions in collaborative ontology as Objective and Funtion intances with relationship: Same As/Near By. As shown in algorithm (3):

- Line 2 and Line 3: for each $E_{objecitve}$, if $E_{objective}$ owns relationship: Same As/Near By, then get $O_{objective}$ which is related to $E_{objective}$.
- Line 4 and Line 5: create new ontology instance: $O_{new}$ for $E_{objective}$ and add $O_{new}$ into collaborative ontology.
- Line 6: creats new Relationship: Same As/Near By between $O_{new}$ and $O_{objective}$, and adds the relationship to collaborative ontology also.

---

**Algorithm (3) Ontology Updating**: insert business objectives into ontology as instances and created relationship: *Same As/Near By*.

**Input:** $L_{objective}$, list of business objectives
**Data:** Collaborative Ontology: CO
**1 while** $L_{objective}$ has next element: $E_{objective}$ **do**
**2**     **if** $E_{objective}$.sameas/nearby!=null **then**
**3**         $O_{objective} = E_{objective}$.sameas/nearby;
**4**         $O_{new}$ = change $E_{objective}$ to ontology instance;
**5**         CO adds $O_{new}$;
**6**         CO adds SameAs/NearBy($O_{objective}$, $O_{new}$);
**7**     **end**;
**8 end**;

## 5    Implementation

Nowadays, SaaS [10] is very widely used. It allows users to use application in a Web Client as rich application. No complex client site installation is required. According to [11], compared with the traditional way which software is purchased for and installed on personal computers, SaaS has advantages: e.g., investment reduction, performance improvement, time saving, easier collaboration, global accessibility, etc. Considering our own needs, the software tool should be able to deploy on ESB [12], which means the software must be a web service. Because the software deals with a collaborative situation, all the partners may use the software in the same time or individually. This leads to a conclusion. SaaS is a quite good solution for our tool: Mediator Modeling 2ool.

Mediator Modeling 2ool bases on GWT [13] and GeasyTools [1]. Mediator Modeling 2ool supports objective and function modeling. It implements business service selection and collaborative process creation.

Figure 6 is the screen shot of Mediator Modeling 2ool. On the left, there are modeling palette and file explorer. On the right, there are element properties and collaborative ontology instances' tree. In the middle, it is modeling place. We could create objective and function model here. Web browser can directly launch Mediator Modeling 2ool.
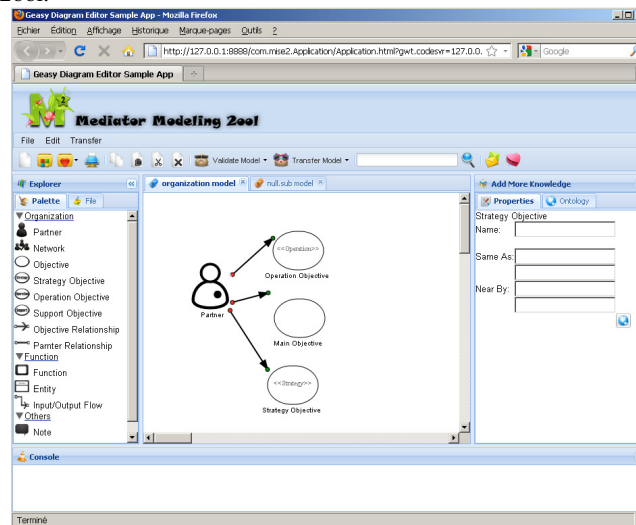


**Fig. 6 -** Mediator Modeling 2ool Main Frame

In the property window in Figure 6, if we click on the button beside text field, a window with collaborative ontology instances' tree comes out. User chooses the same as or near by ontology instance for modeling element by dragging the instance form ontology tree to same as or near by lists' windows. After modeling and choosing same

---

[1] GeasyTools is an open source GWT based API. PetalsLink develops it. The tool helps create graphic elements. It is available on website:
http://research.petalslink.org/display/geasytools/GEasyTools+Overview

as or near by ontology instance, a XML file which contains all the modeling information is saved on the server side. User could save the XML file on local machine too.

For business function, the property also contains semantic annotation. The semantic annotation classifies business function to Service Task, Send Task, Receive Task, User Task and Business Rule Task. Business functions combined with semantic annotations are reused to semantically select web services, which could implement business functions. This part of work is presented in [14].

## 6    Conclusion

Collaborative business process deduction methodology (CBPD) aims to provide a collaborative process cartography, which manages process orchestration and business service selection in collaborative situation. The strong points of CBPD are summarized as followed: i) objective model defines objectives and sub-networks. We could easily verify the small group of partners to complete the task; ii) objective model could model different levels of partners; iii) functional table is separated by columns of partner. This allows each partner to fill its own column independently; iv) collaborative process model is deduced automatically. It saves repeated work and eases heavy workload of user. It also crosses different enterprise modeling standard: from IDEF to BPMN.

However, any system has its weak points. They are summarized as followed: i) business service selection is half manual and half automatic method. User has to create Same As/Near By relationship from model to ontology instances manually. And then Mediator Modeling 2ool automatically links partners' business functions to collaborative and partners' objectives. ii) This system bases on collaborative ontology, it means that a collaborative ontology, which holds numerous instances and covers different domains is required.

With the accomplishment of business service selection, the next task is business process creation. It remains the formalization of transformation rules and definition of gateways in BPMN based collaborative process model. Further more, function model of CBPD defines only business function. We did not consider non-functional characterization (security, privacy and speed). Miss is working on non-functional characterization. [15]

As introduced in [16], CBPD is only the first step of MISE 2.0 project (Mediation Information System Engineering). Beside CBPD, another two main tasks are semantic web services and BPEL transformation. First, MIS deployment bridges the semantic gap between business functions and technical services. Second, it transfers collaborative process model to BPEL (Business Process Execution Language) based collaborative workflow. Last but not least, the collaborative workflow and technical services are deployed on ESB and executed. This part of work is detailed explained in [14].

# References

1. 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.
2. T. W. Malone, K. Crowston, and G. A. Herman, Organizing business knowledge: the MIT process handbook. the MIT Press, 2003.
3. "NF EN ISO 9000 Systèmes de management de la qualité - Principes essentiels et vocabulaire," Sep-2005. [Online]. Available: http://catdoc.mines-albi.fr:8080/Record.htm?idlist=6&record=19134158124919523309. [Accessed: 19-Jan-2012].
4. "NF EN ISO 9000 X50-130 Systèmes de management de la qualité - Principes essentiels et vocabulaire," Oct-2005. [Online]. Available: http://catdoc.mines-albi.fr:8080/Record.htm?idlist=6&record=19143202124919614849. [Accessed: 19-Jan-2012].
5. D. Neiger, L. Churilov, and A. Flitman, "Business Objectives Modelling," in Value-Focused Business Process Engineering : a Systems Approach, vol. 19, Boston, MA: Springer US, 2009, pp. 1–26.
6. C. Menzel and R. J. Mayer, "The IDEF Family of Languages," in Handbook on Architectures of Information Systems, 2006, pp. 215–249.
7. "Announcing the Standard for INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0)." Draft Federal Information Processing Standards Publication 183, 1993.
8. R. M. Smullyan, First-order logic. Dover Publications, 1995.
9. F. Benaben, J. Touzi, V. Rajsiri, and J. P. Lorré, "Mediation Information System Design in a collaborative SOA context through a MDD Approach," Proceedings of MDISIS'08, pp. 1–17, 2008.
10. W. Sun, K. Zhang, S. K. Chen, X. Zhang, and H. Liang, "Software as a service: An integration perspective," Service-Oriented Computing–ICSOC 2007, pp. 558–569, 2010.
11. B. Wu, S. Deng, Y. Li, J. Wu, and J. Yin, "Reference Models for Saas Oriented Business Workflow Management Systems," in 2011 IEEE International Conference on Services Computing (SCC), 2011, pp. 242–249.
12. A. T. Endo and A. Simao, "Model-Based Testing of Service-Oriented Applications via State Models," in 2011 IEEE International Conference on Services Computing (SCC), 2011, pp. 432–439.
13. V. Gupta, Accelerated GWT: Building Enterprise Google Web Toolkit Applications. Apress, 2008.
14. F. Bénaben, N. Boissel-Dallier, J. P. Lorré, and H. Pingaud, "Semantic Reconciliation in Interoperability Management through Model-Driven Approach," Collaborative Networks for a Sustainable World, pp. 705–712, 2010.
15. S. Zribi, F. Bénaben, A. Ben Hamida, and J. P. Lorré, "Towards a Service and Choreography Governance Framework for Future Internet," presented at the I-ESA 2012, Valence, Spain, 2012.
16. W. Mu, F. Bénaben, H. Pingaud, N. Boissel-Dallier, and J.-P. Lorré, "A Model-Driven BPM Approach for SOA Mediation Information System Design in a Collaborative Context," in 2011 IEEE International Conference on Services Computing (SCC), 2011, pp. 747–748.