

Coupling metrics for business process modeling

WIEM KHLIF, NAHLA ZAABOUB, HANENE BEN-ABDALLAH

Mir@el Laboratory,
Faculty of Economics and Management Sciences
Sfax University
B.P. 1088, Sfax 3000
TUNISIA
{Wiem.Khlif, Nahla.Haddar, Hanene.Benabdallah}@fsegs.rnu.tn

Abstract: - Modeling business processes is vital when improving or automating existing business processes, documenting processes properly or comparing business processes. In addition, it is necessary to evaluate the quality of a business process model through a set of quality metrics. One specific categorie of such metrics is coupling which measures the functional and informational dependencies between the tasks/processes in a business process model. Our contribution in this paper consists in adapting object oriented software coupling metrics for business process models. This adaptation is based on correspondences we establish between concepts of the Business Process Modeling Notation and object oriented concepts. The new adapted coupling metrics offer more information about the dependencies among processes and their tasks in terms of data and control. They can be used, for instances to evaluate the transferability effects of errors occurring in a particular task/process. Finally, we validate theoretically the proposed metrics.

Key-Words: - Coupling quality metrics, coupling measurement, quality metrics, business process models

Introduction

The market forces of today's business process (BP) development have begun to place an important emphasis on business process quality. Evidently, the quality of a business process model (BPM) highly influences the deployed business process. This motivated several researchers to propose *metrics* to evaluate the quality of BPM.

In fact, the concept of quality metrics was initially introduced to examine software quality. According to [1], a quality metric is a quantitative scale and a method that can be used to determine the value taken by a characteristic of a software product. Exploiting the maturity of software quality metrics, several researchers adapted several metrics from the field of software engineering for business process models, *cf.* [1] [2]. The adaptation is justified by the fact that a business process model (described by EPC (Event-Driven Process Chain), Petri nets, activity diagrams, or BPMN [3]) manifests several similarities with software models [2] [4]. In fact, business processes and software products have a similar compositional structure: A program is composed of modules or classes, each module consists of statements and each statement contains variables and constants; in the same way, a business process has activities each of which is

composed of elementary operations and each operation uses one or more information to produce new information.

On the basis of these similarities, different researchers confirm that metrics used in software engineering are applicable in the business domain as well, *cf.*, [5] [6] [7]. However, these works are incomplete: several software engineering, especially *coupling* metrics have not been taken into account in the business domain. This shortage motivated us to adapt them in order to have more accurate measurements of complexity of business processes.

Cardoso *et al.* [1] define coupling as a measure that determines the strength of interconnections from one business process to another. Therefore the stronger the coupling between processes is, *i.e.*, the more inter-related they are, the more difficult these processes are to understand, change, and correct and, hence, the more complex the resulting business process model is.

In this paper, we rely on a set of correspondences between OO concepts and business process concepts to adapt software coupling metrics for BPM [8]. In addition, for a better use of the new coupling metrics, it is necessary to validate them. Thus, we propose a mathematical framework which is generic (*i.e.*, independent of any particular BPM) and rigorous, because it is based on precise

mathematical concepts. However, this framework is not intended to be complete or fully objective. Nevertheless, we believe that the formalisms and properties we introduce are convenient and intuitive. This framework contributes constructively to a firmer theoretical ground of BPM measurement.

The remainder of this article is organized as follows: In section 2, we present an overview of coupling metrics defined in the business process domain. Then, we present in section 3 metrics in software engineering that have not been adapted for business process models. Section 4 identifies similarities between business process modeling notation (BPMN) and object oriented software; we rely on these correspondences to show, in section 5, how we adapt the presented software metric to BPM. Then, we propose in section 6 a generalized coupling metric. In section 7, we highlight several properties of the adapted coupling metrics. Finally, we conclude the paper with a summary of the presented work and an outline of its perspectives.

2. Overview on coupling metrics in business processes

Several researchers already identified the potential of business process metrics. These proposed metrics were adapted from the software engineering domain. In this section, we focus on the adapted coupling metrics.

2.1 Adapted coupling metrics

In OO software, coupling of a class can be used to measure the strength of a “relation” between one class and another. Classes are coupled either when a message is passed between objects, or when methods declared in one class use methods or attributes of another class.

Coupling in business process models (BPM) focuses on how strongly the activities in a business process are related, or *connected*, to each other. An activity is connected to another activity if and only if they share one or more information elements [4].

In [1], for a given activity, the coupling metric determines the number of activities related to it. For a given BPM, its coupling metric equals to the number of interconnections between all its activities; in other words, it counts all pairs of activities in the BPM that are connected to each other. In addition, the degree of coupling depends on how complicated the connections are and also on the type of connections between activities (AND, OR, XOR). (For the mathematical definition of this metric, the reader is referred to [1].)

The coupling metric of an activity reflects how critical/important an activity is within a BPM. In fact, an activity with a high coupling metric value functionally determines a large number of activities in the business process. Thus, its malfunctioning may cause several activities to malfunction; this in turn may jeopardize the overall business process functionalities. Such activities should be either avoided within a BPM, or treated with a special care, *e.g.*, by having a monitoring activity for it.

On the other hand, a BPM with a high coupling metric indicates a high level of informational dependency between its activities. Again, such a model produces a vulnerable process and one which maintenance is difficult, etc.

One limit of the coupling metrics proposed in [1] and [4] is that they do not give an indication about the reusability of a BPM. This quality information is important for design through reuse. A second limit is that the first coupling metric is defined in term of data, while the second coupling metric is defined in term of control flow. However, in a business process model a coupling metric should take into account both data and as well as control flow.

3. Other coupling metrics in the software engineering domain

Several metrics have been proposed in the software engineering domain to define coupling. Among these metrics, we cite the following four:

- the coupling between object classes (CBO) defined by Chidamber and Kemerer [7], which counts for each class C the number of classes having methods invoked by those of C;
- the response set for a class (RFC) which determines the set of all methods that can be invoked in response to a message, to an object of the class or by some methods in the class [7];
- the message passing coupling (MPC) metric which measures the complexity of message passing among classes [5][6]. MPC is the number of messages sent out from a method in a class to a method in another class; and
- the information-flow-based coupling (ICP) metric which expresses the number of method invocations in a class C, of methods in class D, weighted by the number of parameters of the invoked methods [9].

On the other hand, Chen and Lu in [10] present also a set of metrics for object oriented design. Among these metrics, they define:

- the operation coupling (OpCpl) to measure the coupling between operations in a class and operations in other classes. It is defined as the sum of the number of operations which access other classes, the number of operations which are accessed by other classes and the number of operations which are “co-operated” with other classes. A “co-operated” operation is one that accesses another class’ operations and vice versa;
- the class coupling metric (ClCpl) to measure the coupling between a class and other classes. It is the sum of the number of accesses to other classes, the number of accesses by other classes and the number of co-operated classes. A co-operated class is one that accesses some other classes and vice versa.

In addition to the above coupling metrics, Brito *et al.* [11] define two types of coupling: 1) Imported Coupling (IC) metric which counts, for a class C, all interactions in which C uses another class; and 2) Exported Coupling (EC) metric which counts, for a class C, all interactions in which C is used.

Also to measure the coupling dimension, Hitz and Montazeri [12] proposed the metric Locality of Data (LD), defined as the ratio of the amount of data local to a class to the total amount of data used by that class:

$$LD = \frac{\sum_{i=1}^n |L_i|}{\sum_{i=1}^n |T_i|} \quad (1)$$

where:

- $L_i (1 \leq i \leq n)$ is the set of “local” variables accessed by method M_i (directly or via read/write methods). By local variables the authors mean non-public instance variables

of a class C, inherited protected instance variables of its super classes and static variables defined locally in M_i . These are non-public instance variables of C, inherited protected instance variables of its super classes and static variables defined locally in M_i .

- $T_i (1 \leq i \leq n)$ is the set of *all* variables used in M_i , except for non static local variables defined in M_i . For the sake of robustness of the measure, all auxiliary variables defined in M_i are excluded because they do not play a major role in a design.

Note that this metric is related to the quality of abstraction embodied by a class, where classes with high data locality are more self-sufficient than those with low data locality.

Overall, the coupling metrics for software engineering presented above define coupling through four angles: message, data, or data and messages. More precisely, the MPC, CFC, ICP, OpCpl and ClCpl metrics are expressed in terms of message; the LD metric is expressed in term of data, while CBO determines coupling in term of both message and data.

4 Correspondences between BPMN and object oriented software

Business processes and software products have many similarities [12] [1]. To exploit these similarities, we focussed on determining correspondences between Business Process Modeling Notation (BPMN) concepts [12] [13] and the object oriented software ones. The choice of BPMN is justified by the fact that this formalism defines an OMG standard notation for modeling business processes. The choice of BPMN is justified by the fact that this formalism defines a standard notation for modeling business processes.

Table 1 enumerates our correspondences between object oriented software concepts and BPMN concepts.

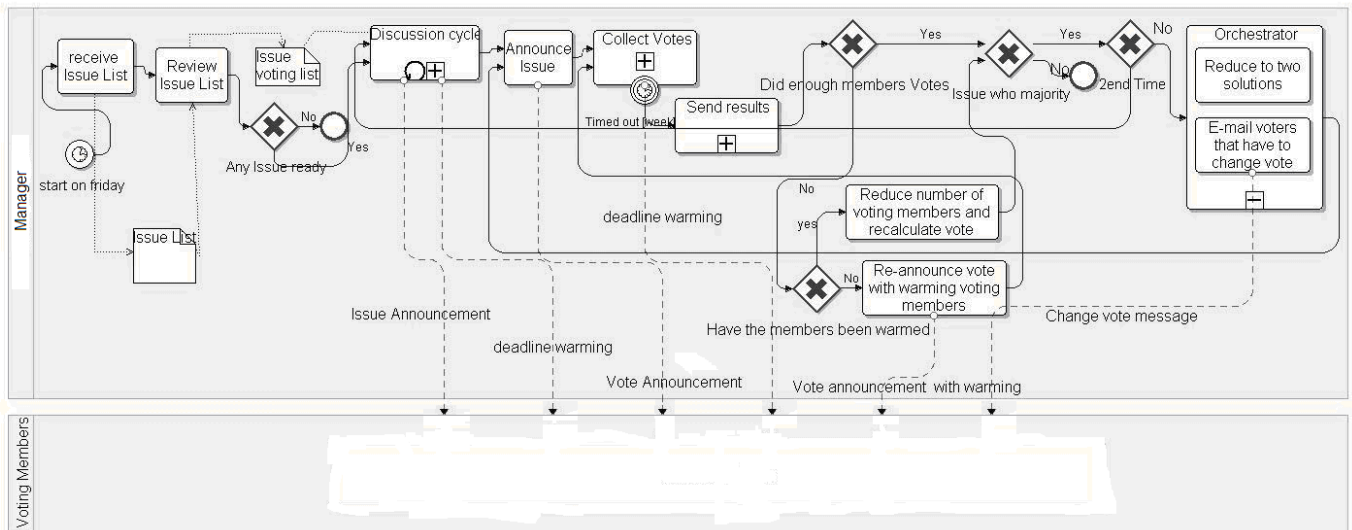


Fig. 1. E-Mail Voting Process

Table 1. Similarities between BPMN and OO software core concepts.

<i>Object oriented software</i>	<i>BPMN Notation</i>
Class/Package	Sub Process, process
Method	Task
Method invocation	Reception of a sequence flow or a message flow by a task.
Variable/ Constant	Data object
Comment line	Annotation
Interface of a class	Interface of a process/sub process: the set of tasks in a process which send or receive a message flow.
Local data in a class	Data object generated by a process and not used by an activity from outside this process.
Data used by a class	Data object used or generated by a process.

Relying on the basic concepts of object-oriented software, we map a class or a package to a process or to a sub process in the business domain.

A class has attributes and methods. The attributes represent the local data of a class. In addition, a class can use data. Thus, we associate local data to data objects produced by to tasks of a process. While data used by a process are mapped to data objects produced or used by it.

A class method is mapped to a task in a sub process or in a process. All public methods determine the class interface. By applying this concept to BPMN, the process/sub process

interface will be defined as being the set of tasks in a process which send or receive a message flow.

Finally, comments in a software product correspond to annotations in BPMN.

5 New adapted coupling metrics for BPM

In this section, we show how the previous mappings of OO software engineering to BPMN concepts can be used to adapt the software coupling metrics for business process models. We also discuss the advantages gained by the new metrics. For a better understanding of these metrics, we first introduce an example that we will use to illustrate the newly defined metrics.

5.1 E-mail voting process

To show the application of adapted coupling metrics we use a modified version of the e-mail voting process given in the OMG specification of BPMN [14]. This example is presented in Fig. 1 with BPMN.

The e-mail voting Process is modelled from the perspective of the manager of the Issues List and the discussion around this list. From that point of view, the voting members of the working group are considered as external Participants within whom communication is ensured by messages (shown as Message Flow). The Issue List Manager reviews the list and determines if there are any issues that are ready for going through the discussion and voting cycle, in which case a Decision must be

made. If there are no issues ready, then the Process is over for that week--to be taken up again the following week. If there are issues ready, then the Process continues with the discussion cycle. Decision and this Sub-Process has two incoming Sequence Flows, one of which originates from a downstream Decision and is part of a loop. It is one of a set of five complex loops in the Process.

The contents of the “Discussion Cycle” Sub-Process and the activities that follow are described in Fig. 2.

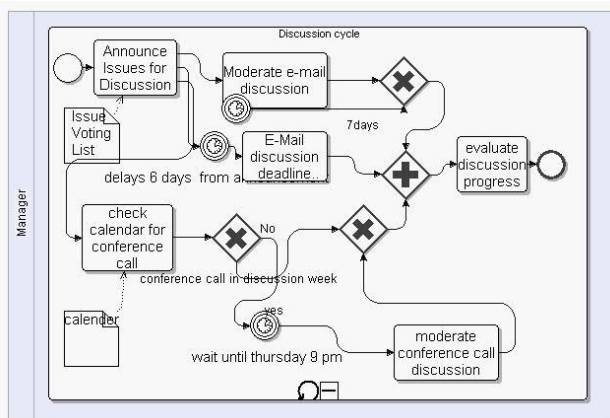


Fig. 2: Discussion Cycle sub-process details

On the other hand, the sub-process “Collect vote” (shown in Fig. 3) starts out with a Task for the issue list manager to send out an e-mail to announce to the working group, and the voting members in particular, which lets them know that the issues are now ready for voting. Since this Task sends a message to an outside Participant (the working group members), an outgoing Message Flow is seen from the “Announce Issues” Task to the “Voting Members” Pool in Fig. 1. This Task is also a target for one of the complex loops in the Process.

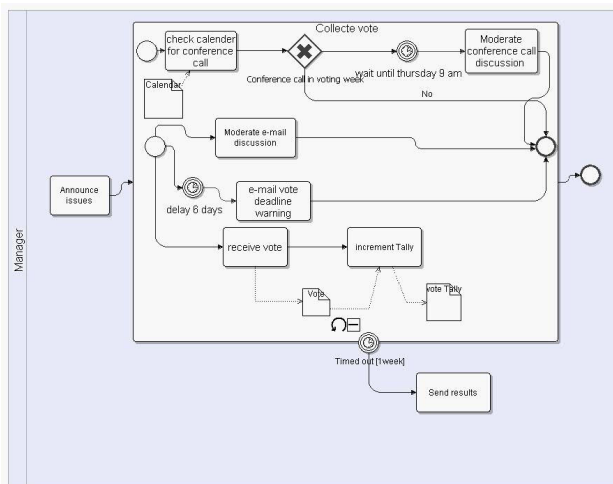


Fig. 3: Collect votes sub-process details

The last segment of the E-Mail Voting Process continues from where the sub-process “Collect Votes” left off. It contains four Decisions that interact with each other and create loops to upstream activities.

The first Decision, “Did Enough Members Vote?”, is necessary since two-thirds of the voting members are required to approve any solution to an issue. If less than two-thirds of the voting members cast votes, which sometimes happens, the issues cannot be resolved. This is modelled by a Decision Flow to another Decision for both of its Alternatives. The “No” Alternative is followed by the “Have the Members been Warned?” If a voting member misses a vote, they are warned. If they miss a second vote, they lose their status as a voting member and the voting percentages are recalculate through a Task (“Reduce number of Voting Members and Recalculate Vote”). If they have not yet been warned, then a warning is sent and the voting week is repeated. If all issues are resolved, then the Process is done. If not, then another Decision is required. The voter is given two chances before it goes back to another cycle of discussion. The first time will see a reduction of the number of solutions to the two most popular based on the vote (more if there are ties). Some voting members will have to change their votes just because their solution is no longer valid. These two activities are placed in a Sub-Process to show how a Sub-Process without Start and End Events can be used to create a simple set of parallel activities. (Informally, this is called a “parallel box”. It is not a special object, but another use of Sub-Processes. For simple situations, it can be used to show a set of parallel activities without the extra clutter of a lot of Sequence Flow.) In reality, these two Tasks cannot actually be done in parallel, but they are modeled this way to highlight the optional use of Start and End Events.

After the parallel box, the flow loops back to the “Collect Votes” Sub-Process. If there already has been two cycles of voting, then the process Flow back to the “Decision Cycle” Sub-Process.

5.2 Adapting the MPC metric

In the business process domain, MPC becomes message or sequence flow passing coupling (MSPC). It is the number of message or sequence

flows sent from a process task to a task of another process.

Note this measure takes into account only the coupling in term of interaction between tasks. Thus, it helps to estimate the degree of dependency between the process tasks and the tasks of others processes.

In our example, the MSPC metric is not applicable.

5.3 Adapting the RFC metric

By examining coupling metrics in the software engineering domain, we notice that the response for a class (RFC) metric [14] focuses on the coupling in terms of control flow. We call the adapted version of this metric response for a process (RFP) in the business domain. We compute it as follows:

$$RFP = |RS|$$

where $RS = \{T_j\} \cup \{R_i\}$ is the set of all responses of a process, where $\{R_i\}$ is the set of tasks invoked by a task i in the process and $\{T_j\}$ is the set of all tasks j in the process.

Let us consider the "Orchestrator" process which contains in parallel the two tasks: "Reduce to two solutions" and "E-mail voters that have to change vote". Its set of responses (RS) contains the following tasks:

$RS = \{ \text{"Reduce to two solutions"}, \text{"E-mail voters that have to change vote"} \}$
 $\cup \{ \text{"Voting Members"}, \text{"Announce Issues"} \}$

Thus, the RFP of "Orchestrator" is equal to 4.

Note that, the larger the RFP is, the greater the complexity of the process is: In deed, if a large number of tasks can be invoked in response to a message, then the process becomes complex and requires a greater level of understanding.

5.3 Adapting ICP metric

This metric becomes the number of tasks in a process P that receive sequence or message flows (IFCP), weighted by the number of data used by these tasks.

Note that the more data is used, the stronger the coupling between tasks.

In our example, the IFCP metric is not applicable, since there is no task in a process that receives sequence or message flows.

5.5 Adapting of Locality of Data (LD)

Another coupling metric we see adaptable is locality of data (LD) [12]. This metric links data from the activity (process or sub process) to the total data used by this activity. The adapted metric, we call *locality of data activity* (LDA), for an activity (sub process or task) with n tasks can be expressed mathematically as follow:

$$LDA = \frac{\sum_{i=1}^n |L_i|}{\sum_{i=1}^n |DT_i|} \quad (2)$$

where DT_i ($1 \leq i \leq n$) is the set of data associated to task T_i within the activity (data objects generated or used by the tasks of a process), and L_i ($1 \leq i \leq n$) is the set of data produced by the activity/task T_i .

A process with a high data locality is more self-sufficient than those with a low data locality. In addition, the Locality of Data activity metric (LD) influences the process's reuse potential. This metric also influences another external attribute: the testability of the process. The higher the LDA value the easier it is to test the process.

Let us consider the "Review Issue List" task in Figure 1; this task uses the "Issue List" data which is produced by the "Receive Issue List" task. Therefore, the LDA of the "Review Issue List" task is computed as follows:

$$LDA = \frac{\sum_{i=1}^1 |\{ \text{"IssueVotingList"} \}|}{\sum_{i=1}^1 |\{ \text{"IssueList"}, \text{"IssueVotingList"} \}|} = \frac{1}{2}$$

We should note that the main point of this metric is the notion of self-sufficiency. A process with a high self-sufficiency, means that the process uses very few data's that are not "local" to it. In fact the "non-local" data that can be used by the tasks of a process are the data produced par other processes and used by them.

5.6 Adapting of coupling type

We adapt the coupling type metric in the business domain as follows:

- ICP (Imported Coupling of a Process): counts, for each (sub-) process, the number of message/sequence flows sent by either the tasks of the (sub-) process or the (sub-) process itself.
- ECP (Exported Coupling of a Process): counts, for each (sub-) process, the number of message/sequence flows received by

either the tasks of the (sub-) process or the (sub-) process itself.

Let us consider the simple sub-process "Discussion Cycle" of our example (Fig. 1): It sends a sequence flow to the "Announce Issues" task and two message flows to the process "Voting members". Thus, its ICP is equal to 3. In addition, this sub-process receives two sequence flows: one from the gateway "Any issues ready" and another from the gateway "2end time". Thus, the ECP of "Discussion Cycle" is equal to 2.

Note that a process with high ICP value highly depends on several external services offered by other processes. This might increase delays, costs and error probability. In addition, a process with a high ECP has a considerable influence on the whole model since a multitude of processes depends on its services. In other words, problems encountered in the business process may be caused by a fault in this influential process.

5.7 Adapting OpCpl

We adapt Operation coupling (OpCpl) metric as Task Coupling (TCpl). It measures the coupling between tasks of a process P and others processes' tasks. It is defined as the sum of the number of tasks in P sending sequence or message flows to other processes, the number of tasks in P receiving sequence or message flows by others processes and the number of tasks in P co-operating with other processes. A co-operating task is one that sends some sequence or message flows to tasks in other processes and vice versa.

The TCpl metric is not applicable in our running example since there is no sequence flow exchanged between the process tasks.

5.8 Adapting CICpl

Our objective was to propose a metric that could be used in the same way as the CICpl metric but to evaluate the processes' coupling. The resulting metric is called Process coupling metric (PCpl).

PCpl measures the coupling between a process and other processes. It is the sum of the message or sequence flows sent from a process P to other processes, the number of message or sequence flows sent by other processes to a process P and the number of processes co-operating with P. A co-operating process is one that sends or receives sequence or message flows.

Let us consider the simple sub-process "Discussion Cycle". It sends three flows. However, no process sends sequence or message to it. We do not take into account the flows received by the gateways.

It should be noted that the ICP / ECP measures take into account respectively the number of message or sequence flows sent/received by process. Contrairely, the PCpl measure takes into account only the number of flows sent directly from a process to other processes, and vice versa. The latter neglects the sequence flows received by such events, gateway.

5.9 Adapting CBO metric

The metric proposed by Chidamber and Kemerer [6] can be adapted to evaluate the coupling of processes in the following way. First, to calculate the interconnection between processes, we need to identify the number of other processes to which a process is coupled, *i.e.* the tasks defined in a process use tasks or data defined in other processes. We call this metric Coupling Between Processes (CBP).

The CBP metric takes into account not only the control flow but also data. Since the coupling in a business process model can be interpreted in terms of control flow and data, we choose the appropriate metric CBP (Coupling between processes) as a generalized coupling metric.

It should be noted that the CBP is the only measure that relates to both aspects of coupling: imported and exported. CBP makes no distinction between import and export coupling: two processes are coupled if one uses the other or vice versa.

6. Proposal of an overall coupling metric

We propose an overall process coupling metric that is inspired from CBO metric. This metric accounts for coupling according to the following cases:

- [1] 1. between two processes whose internal structure is not specified, *i.e.*, both processes are considered as "Black Boxes";
- [2] 2. between two processes where the first one is considered as a Black Box and the second is considered a White Box;
- [3] 3. between two white and black boxes; and
- [4] 4. between two processes considered as "White Boxes".

We extend the definition of the coupling metric CBP as follows: "Two processes are related if the tasks defined in a process use tasks or data defined in the other process", according to the four cases already presented and in term of tasks and data.

6.1 Task-based coupling

In term of tasks, the coupling is defined between tasks or between a task and an event. We start by analyzing the coupling in between tasks.

6.1.1 Coupling between tasks

In this case, the coupling is expressed between two black or white boxes or between a black and white box (or vice versa). We distinguish the following four cases:

- *Between two black boxes:* The coupling is defined as the number of message flows sent and received from the boundary of a process. The black box process sends two messages flow and a sequence flow. In addition, it receives two sequences flow. The number of message flows sent and received from the boundary of the Process of our running example is 5.
- *Between a black box and white box:* In this case, we define the coupling by all tasks of a process (white box) receiving messages flow by the process considered as a Black Box. In our example, the metric is not applicable.
- *Between a white box and black box :* we determine the coupling by the process considered as a black box receiving a message flow by the tasks of the processes considered as a white box.

Let us consider the process "Voting Members", it receives a message flow from the task "E-mail voters that have to change vote" in the process "Orchestrator". In this case, the number of message flow is equal to 1.

- *Between two white boxes:* A process is coupled to other processes if the tasks of a process P use the tasks of others processes. A task uses another by sending a message or sequence flow to another task.

For our running example case, the coupling is not applicable.

6.1.2 Between task and event

In this case, the coupling can not be determined between two black boxes. However, it is expressed between a black and white box and vice versa and between two white boxes.

- *Between a black box and white box:* the process P1 sends messages flow to an intermediate event of a process P2. In our example, the coupling is not applicable.
- *Between a white box and black box:* in this case two processes are coupled by an intermediate event in a process which sends a message flow to a process.

Let us consider the intermediate event 'time out week' in the process 'Collect Votes' sub-process. It sends a sequence flow to the black box 'Send results'.

- *Between two white boxes:* in this case, two processes are coupled by an intermediate event in a process which sends a message flow to a task of another process.

In our example, the coupling isn't applicable.

6.2 Data-based coupling

A task uses data of another process if the link between these two elements is represented by an oriented association or the data is attached to the sequence flow sent by a task to another one.

The coupling is expressed between tow black or white boxes or between a black and white box and vice versa.

- *Between two black boxes:* Consider two processes P1 and P2 connected to data by an oriented association. This expresses that the first process send data to the second process. In our example, the coupling is not applicable.
- *Between a black box and white box:* Consider two processes P1 and P2 connected to data by an oriented association. We explain this by the fact that the process P1 sends data to the task of process P2.

In our example, the coupling is not applicable.

- *Between a white box and black box:* Let two processes P1 and P2 be connected to a data entity by an oriented association. In this case, a task in a process P1 sends data to a process P2.

- *Between two white boxes*: Two processes P1 and P2 are connected to the data by:
 - An oriented association: a task in P1 sends a data to a given task in P2.
 - A sequence flow to which this information is attached, or
 - based on a branching

This coupling measure is useful to determine how complex the testing of various parts of a design is likely to be. The higher the inter-processes coupling, the more rigorous the testing needs to be.

We note also that an excessive coupling is detrimental to modular business process and prevents reuse. The more independent a business process is, the easier it is reuse in another application. In order to improve modularity inter-processes couples should be kept to a minimum. The larger the number of couples, the higher the sensitivity to changes in other parts of the business process modelling and therefore maintenance is more difficult.

In summary, these coupling metrics reflect the different interactions types between processes. Indeed, these types of interactions help to classify the proposed measures. If the interaction is expressed in terms of tasks, the type of interaction can be determined as follows:

- *Process-Process (PP)*: There is an interaction having a type Process-Process if the coupling is defined in terms of task between two black boxes. As an example, we have the PCpl and CBT metric.
- *Process-Task (PT)*: There is an interaction having a type Process-Task if the coupling is defined in terms of tasks between a black box and white box or vice versa. For example, we have the ICP, ECP and CBP metrics.
- *Task-task (TT)*: There is an interaction having a type Task-Task if the coupling is defined in terms of task between two white boxes. As an example, we have the MSPC, RFP, TCpl and CBP metrics.
- *Event-Process (EP)*: there is an interaction having a type Event-Process if the coupling is defined in terms of task respectively between a white box and black box. As an example, we have the CBP metric.

- *Event-Task (ET)*: There is an interaction having a type Event-Task if the coupling is defined in terms of task between two white boxes. As an example, we have the CBP metric.

On the other hand, if the interaction is expressed in terms of data, the type of interaction can be determined as follows:

- *Process-data (PD)*: there is an interaction having a type Process-data if the coupling is defined in terms of data between a black box and white box and vice versa. For example LDA and CBP metrics.
- *Task-data (TD)*: There is an interaction having a type Task-data if the coupling is defined in terms of data between a black box and white box and vice versa. As an example, we have the IFCP, LDA and CBP metrics.

According to our classification, we note that CBP considers the different types of interaction between elements' business process model. It expresses a general definition of the coupling that is more detailed than others metrics.

For a better use of these metrics, it is necessary to validate them. Thus, we propose a mathematical framework which is generic, because it is not specific to any particular BPM, and rigorous, because it is based on precise mathematical concepts. This framework defines the measurement coupling. It does not intend to be complete or fully objective. However, we believe that the formalisms and properties we introduce are convenient and intuitive. This framework contributes constructively to a firmer theoretical ground of BPM measurement.

7 Properties of Coupling

Intuitively, the concept of coupling captures the amount of relationship between the elements belonging to different processes of a business process model. Furthermore, given a process p , two kinds of coupling can be defined: inbound coupling and outbound coupling. The former captures the amount of relationships from elements outside p to elements inside p ; the latter captures the amount of relationships from elements inside p to elements outside p .

We expect coupling to be non-negative (property Coupling.1), and null when there are no relationships among processes (property

Coupling.2). In addition, when additional relationships are created across processes, we expect coupling not to decrease since these processes become more interdependent (property Coupling.3).

On the other hand, the coupling of a process obtained by merging two unrelated processes is equal to the sum of the couplings of the two original processes. In addition, merging processes can only decrease coupling since there may exist relationships among them and, therefore, certain inter-process relationships may disappear in the overall process (property Coupling.4, property Coupling.5).

Given a process p , the above coupling properties obviously hold for the Import Coupling and Export Coupling metrics.

In addition, properties Coupling.1 and Coupling.2 are obviously satisfied for CBP. Property Coupling.3 is satisfied, since CBP cannot decrease by adding one more relationship between features belonging to different processes (*i.e.*, one process uses one more task or data belonging to another process). Furthermore, this metric also satisfies property Coupling.4: CBP can only remain constant or decrease when two processes are grouped into one.

On the other hand, coupling metric Response For a process (RFP) also satisfies property Coupling.3: In fact, adding outside tasks sent message flow to a process can only increase RFP. In addition property Coupling.4 also holds for RFP because merging processes does not change RFP's value since RFP does not distinguish between inside and outside tasks invocations. Similarly, when there are no flows between the tasks the processes, property Coupling.5 also holds for RFP. This result is to expected since RFP is the result of the addition of two terms: the number of tasks in the process (a size measure) and the number of tasks (called a coupling measure).

8 Conclusion

Currently, business process metrics used to evaluate business process model are only a recently explored area of research. In this paper, we adapt several coupling metrics in software engineering to the business process domain.

Our adaptations rely on similarities between BPMN and OO concepts. In addition, they accounted for the different types of interactions that can exist between the activities in a business process model: data and tasks. In terms of tasks,

the coupling is defined between tasks or between a task and an event.

Furthermore, the proposed coupling metrics satisfy several properties which make them suitable for the identification of problems in a business process and design process models. Then, we propose a mathematical framework which is generic, and we propose properties inspired from software engineering in order to validate the proposed metrics.

We are currently extending the proposed coupling metrics to take into account additional important measurement concepts such as complexity and cohesion. In addition, we are defining a formal framework for process models and the BPM metrics. Such a framework will provide for a mathematical foundation for rigorous analyses of business process models.

References:

- [1] J. Cardoso, J. Mendling, J. Neuman, H.A. Reijers. (2006), A discourse on complexity of process models, In: Eder, J.; Dustdar, S. et al, editors, BPM 2006 workshops. *Lecture Notes in Computer Science* 4103, Springer-Verlag, Berlin, pp. 115-126.
- [2] Gruhn.V, and Laue.R, Complexity metrics for business process models, In: Witold Abramowicz and Heinrich C. Mayer, editors, *9th international conference on business information systems* (2006), vol. 85 of Lecture Notes in Informatics, pp. 1-12.
- [3] Business Process Modeling Notation, V1.1, *OMG Available Specification* OMG Document Number: formal/2008-01-17 Standard document.
- [4] I. Vanderfeesten, J. Cardoso, H.A. Reijers, A weighted coupling metric for business process models, Technische Universiteit Eindhoven, Department of Technology Management, PO Box 513, 5600 MB Eindhoven, The Netherlands, University of Madeira, Department of Mathematics and Engineering.
- [5] W. Li and S. Henry, Object oriented metrics that predict maintainability, *Journal of Systems and Software*, Vol. 23, pp. 111-122, February 1993.
- [6] W. Li, S. Henry, D. Kafura, R. Schulman, « Measuring Object-Oriented Design », *Journal of Object-Oriented Programming* 8(4), pp.48-55, August 1995.
- [7] S.R. Chidamber, C.F. Kemerer, Authors' Reply to: comments on «A metrics suite for

object oriented design », *IEEE Transactions on software Engineering*, 21(3), p.265, March 1995.

- [8] W.khlif, L.makni, N.haddar, H.B.Abdallah, «Quality metrics for business process modelling », *Wseas International conferences*, Genova, Italy, October 2009.
- [9] Y.-S. Lee, B.-S. Liang, S.-F. Wu, F.-J. Wang, "Measuring the Coupling and Cohesion of an Object-Oriented Program Based on Information Flow", in *Proc. International Conference on Software Quality*, Maribor, Slovenia, 1995.
- [10] Chen and J.F Lu. A new metric for object oriented design. *Information and software Technology*, 35(4): 232-240, April 1993.
- [11] F. Brito and Abreu and W.Melo. Evaluating the impact of object oriented design on software quality, In *Proc. METRICS'97*, Berlin, Germany, March 1997. *IEEE*.
- [12] Reijers, Hajo A, Design and control of workflow processes: *business process management for the service industry* 2002.
- [13] Owen, M., Raj, A., Popkin Software, BPMN and Business Process Management Introduction to the New Business Process Modeling Standard, (2004).
- [14] Business Process Modeling Notation, V1.1, *OMG Available Specification* *OMG Document Number: formal/2008-01-17* Standard document.