Using COSMIC FSM Method to Analyze the Impact of Functional Changes in Business Process Models

Wiem Khlif[1], Asma Sellami[1], Mariem Haoues[1] and Hanêne Ben-Abdallah[2,1]
[1]Mir@cl Laboratory, University of Sfax, Sfax, Tunisia
[2]King Abdulaziz University, Jeddah, K.S.A.

Keywords: COSMIC Functional Size Measurement, Business Process, Functional Change, Impact Analysis, Requirement Change.

Abstract: Today's dynamic, socioeconomic constraints often force enterprises to change their Business Processes (BP), for instance, by deleting some activities or merging certain work units. These changes induce functional changes on the business process whose performance might be affected. Consequently, there is need for a well-defined change measure and a control process to ensure the success of the BP management projects. This paper proposes to apply COSMIC Functional Size Measurement method to evaluate the BP functional changes to help both business process developers and decision makers to accept/defer or deny a functional change. The proposed evaluation identifies the *real* impact of a functional change on the project and provides indicators for analyzing the functional change status. In addition, when there are multiple change requests, it uses a heuristic algorithm to prioritize the changes while minimizing the required effort to answer them.

1 INTRODUCTION

Today's dynamic, socioeconomic constraints often force enterprises to change their Business Processes (BP), by replacing some of their activities, restructuring their business units (Khlif et al., 2017) merging with partners (La Rosa et al., 2013), moving to new IT platforms like the cloud (Ferme et al., 2016), and so on. These changes induce functional changes on the business process, whose impact on the BP management must be carefully analyzed. Indeed, a functional change may incur additional cost to be implemented, and degrade the BP performance.

Several researchers addressed change impact analysis in different phases of the BP lifecycle:

- in the requirement phase to identify the elements impacted by a change (*e.g.*, (Khlif *et al.*, 2017));
- in the design phase to analyze the effects of a change (*e.g.*, (Uronkarn and Senivongse, 2014)); or
- in the implementation phase by mining BP event logs to overview changes that have happened (*e.g.*, (Nezhad *et al.*, 2011) and (Van der Aalst, 2011)).

Independently of their application phase, the proposed functional change analysis solutions are designed to identify an adopted change. In other words, they do not provide for a means to measure the impact of a change request in order to assist in deciding on its adoption or rejection. This is the main contribution of the herein proposed method.

More specifically, in this paper, we propose to build on our preliminary work (Khlif *et al.*, 2017) on Functional Change (FC) impact analysis in the early requirement phase of a BP project. We reuse the top-down approach to decompose BP modelled in the standard BPMN notation (ISO/IEC 19510, 2013) into fragments that specify requirements at different levels of abstraction. This decomposition helps in assessing the impact of a FC, and henceforth deciding whether to adopt it or refuse it. In addition, we propose here to measure the impact of a FC by applying the COSMIC method (COSMIC, 2017) which can be used for sizing functional changes. Compared to other FSM methods, COSMIC can be used for sizing FC and it can be applied to various functional domains (business application, real time, infrastructure software, etc.) at any phase of the software life-cycle.

Toward this end, we first propose to expand the Task Descriptions (TD) (Lauesen, 2002) proposed to textually document the Functional User

Khlif, W., Sellami, A., Haoues, M. and Ben-Abdallah, H.
Using COSMIC FSM Method to Analyze the Impact of Functional Changes in Business Process Models.
DOI: 10.5220/0006707301240136
In *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018)*, pages 124-136
ISBN: 978-989-758-300-1
Copyright © 2019 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

Requirements (FUR) (COSMIC, 2017). The expanded TD template describes behavioural elements of the BP, including the relationships among tasks and their types. It helps BP developers to establish a concrete way of quantifying a FC in terms of COSMIC Function Point (CFP) units. It also helps decision makers in accepting or refusing a FC based on their analysis of its impact on the BP project.

Furthermore, to provide for this decision-making, we present two contributions: First, we propose a method for the assessment of the FC status. This later can be used as indicator to monitor FC requests, and it can also be used to identify the FC risks on the project scope. Second, we propose an algorithm for assigning priorities for multiple change requests based on two factors: the preference of the change requester and the FC status.

The remainder of this paper is organized as follows: Section 2 overviews the COSMIC Functional Size Measurement (FSM) method, discusses related work and presents an expanded textual description of a business concept for measurement purposes. In section 3, we measure the Functional Size of a FC and identify its impact on the functional size of the affected business concept and its impact on the related business concepts. Then, we proposes an algorithm for prioritizing functional changes. Section 4 first illustrates our change impact analysis approach through the business application "Airline Company"; secondly, it identifies threats to the validity of our study. Section 5 summarizes the presented work and outlines some of its extensions.

2 BACKGROUND

2.1 COSMIC FSM Method

COSMIC FSM method proposes a standardized measurement process to measure the functional size of any software. The software functional size is derived by quantifying the FURs (COSMIC, 2017). FURs represent the “user practices and procedures that the software must perform” (ISO 14143, 2012). A FUR involves a number of functional processes. Each Functional Process (FP) consists of a set of functional sub-processes that move data or manipulate data. A data movement moves a single data group from/to a user (respectively Entry and eXit data movement) or from/to a persistent storage (respectively Read and Write data movement). The unit of measurement is one data movement for one data group, referred to as one CFP. The size of a FP is equal to the number of its data movements. Consequently, the software size is equal to the sum of the sizes of all its functional processes.
Furthermore, COSMIC defines a FC as "any combination of additions of data movements or of modifications or deletions of existing data movements" (COSMIC, 2017). To measure the Functional Size of a FC, referred to as FS(FC), COSMIC recommends to attribute one CFP for each changed data movement regardless of the change type (addition, deletion, or modification). Thus, the functional size of the software after a FC is given as the sum of all added data movements minus the functional size of all removed data movements (COSMIC, 2017).

2.2 Related Work on FC Analysis in BP Models

This section overviews studies focusing on change impact analysis in different phases of the BP lifecycle:

In the requirement specification phase, (khlif et al., 2017), presented a top-down decomposition method of BPM into fragments. It used a Lauesen’s Task & Support Descriptions template (Lauesen, 2002) to derive from the fragment descriptions, the scenarios' descriptions of the functional requirements associated with the whole BPMN model. Besides, the presented method provides for measuring and analysing change impact of BP models at different levels of abstraction.

In the design phase, (Uronkarn and Senivongse, 2014) proposed a structural change pattern-driven traceability in BP. Given an initial BP model and its modified version, they compared both models to instantiate the change patterns, and, therefore determine the affected BP model elements. This approach can be used when the change has already been adopted and the aim is to go-back and analyzed its effects.

In addition, (Wang *et al.*, 2012) proposed a pattern-based approach to facilitate the change impact analysis for service-oriented BP models. They focused on a typical scenario where multiple services are supported by a single BP. Their change impact analysis approach can be used to enable the analysis of the BP change propagation and associated services. They proposed algorithms for analyzing change impact and scopes.

In the implementation phase, (Nezhad et al., 2011) (Van der Aalst, 2011) proposed an analysis of “Event logs” through process mining approaches to provide support for changing processes. The proposed change processes provide an aggregated overview of all changes that happened so far.

In summary, many researchers studied change impact analysis in BP models either at the design or implementation phases. However, it is not reflected in the requirement phase since it lacks a detailed textual description. In addition, change impact analysis (CIA) before starting the changes is important to decide on the change acceptance. The CIA provides a useful information that lead to better solution for continuous improvement, for predictions in subsequent execution phases, and the avoidance of project failure.

2.3 Business Concept Description

In our previous work (Khlif et al., 2017), we presented a hierarchical approach to describe a business process by dividing a BPMN model into fragments. Each one depicts how the organizational process fits into the business concept. It can serve both as a check on sequential tasks in each lane, and as a description of the whole business process.

The fragment-based decomposition provides the information needed to apply COSMIC FSM method at both the functional level (fragments) and the dynamic level (business activities). The dynamic level of a fragment is modelled through the textual description of its business activities (BA). Because there is no standard for BA documentation, we use the Task and Task & Support Descriptions (Lauesen, 2002) for the requirements specification as a means to document the BA concept.

That is, as illustrated in Figure 1, we propose to document each business activity with: the Trigger/precondition for execution, the detailed tasks of the Main scenario, the detailed tasks of the Alternative scenario and Error scenario. This business concept description contains three blocks. The first block is used to identify the activity, it includes information such as name, purpose of the activity. The second block describes the activity, and it provides information such as Trigger/precondition for execution, main scenario expressed by sub tasks and their sequence, alternative and error scenarios are indicated by the variant during the execution of the tasks and problems, etc. Finally, the third block includes special requirements (such as non-functional requirement & project requirements and constraint). The Main Scenario (MS) expresses an unconditional set of steps that describe how the fragment can be achieved and all related actors' interests can be satisfied. Each step is essential to achieve the fragment and each step must succeed. Variations of the Main Scenario (VMS) meets the post conditions of a business fragment. The conditions are expressed, after a split gateway (exclusive, inclusive complex), by the conditional sequence flow. Exception Scenario (EMS) indicates the exception scenario does not realize the post conditions of an activity. It can be generated by different types of intermediate events.

Name of the activity: <unique name assigned to an activity> ***Purpose of the activity:*** < a summary of an activity purpose>
Trigger/Precondition for execution: <A list of conditions that must be true to initialize the activity> *Relationships:* <contains>: [activities in relation with this activity by "contains"] <can_be_realized>: [activities in relation with this activity by "can_be_realized"] ***Activity and their sequence (Main scenario)*** ***Begin*** at <steps of the scenario of the trigger to goal> [<Pre-condition>] <Task description> <type: ActiveREQ, ActiveREP, ActiveRET, ActivePER> [<Int-datagroup>] [<Out-datagroup>] ***End*** ***Variant during the execution of the activity and problems (Alternative scenario)*** ***Begin*** at Task description [<Pre-condition>] <Task description> <type: ActiveREQ, ActiveREP, ActiveRET, ActivePER> [<Int-datagroup>] [<Out-datagroup>] The main scenario back to Task description ***End*** ***Variant during the execution of the task and problems (Error scenario)*** ***Begin*** at <Task description> [<Pre-condition>] <Task description> <type: ActiveREQ, ActiveREP, ActiveRET, ActivePER> [<Int-datagroup>] [<Out-datagroup>] The main scenario back to Task description ***End***
Critical situations of execution of the activity Special requirement: <non Functional requirement> <Project requirement and constraints>

Figure 1: Detailed description of a business concept.

In a business activity, a task is typically detailed by the <Task description>, <Task Type><Int-data group><out-datagroup>. The task type can be "ActiveREQ", "ActiveREP", "ActiveRET", "ActivePER" or "Passive" representing respectively "Entry", "eXit", "Read", "Write" or "data manipulation" in COSMIC concepts. "ActiveREQ" corresponds to a task representing the act of asking for something. "ActiveREP" corresponds to a reply sent after asking for something. "ActiveRET" corresponds to a task allowing data retrieval. "ActivePER" corresponds to a task allowing the data record. "Passive" task does not lead to an exchange of data.

For example, in the Figure 6, we present the detailed description of the fragment F2 of the "Airline company" model (See Figure 7). The "*recover devices list*" type in BA23 represents "ActiveREQ" and "ActiveRET". The "*maintain devices*" expresses an "ActiveREQ" and "ActivePER".

Note that the business concept description of Figure 1 can be used at different levels of abstractions: in the high level, the business concept represents a fragment or a business activity, whereas in the low level, it represents an atomic task.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

3 FUNCTIONAL SIZE OF FC AND ITS IMPACT ANALYSIS

The FUR affected by a change expresses a FC, such as adding a task, modifying a data object, deleting a relation between tasks, etc. Aligning the FUR with COSMIC is crucial for the traceability of a FC and eventually the BP project success. In this section, we use this alignment to assess the Functional Size of a FC (noted by FS(FC)) in terms of CFP units, and to determine its impact on the size of the changed business concept and the whole BPMN model.

Our change impact analysis operates in the four steps illustrated in Figure 2: A FC can be submitted by a requester (Step 1). Each FC is translated into a description that annotates the BPMN model (Step 2) before proceeding with its impact analysis (Step 3). The functional change impact analysis generates the FC status report that will be used to evaluate the risk on the project scope (Step 4). The provided measurement results can be used as an indicator to make the appropriate decision regarding the change purpose. Responses to risk can be classified as accepting, deferring or denying (Fairley, 2009).

In this Change Impact Analysis (CIA), we distinguish between inter FC and intra FC. An inter FC affects a business activity (BAa) in relation with other BAs or a fragment F or the related fragments in the BPMN model. An intra FC affects only the BA (or the fragment) that does not have any impact on other BAs (or other fragments) in the BPMN model. Thus, an intra FC may affect directly the size of the BPMN and that of the BAa (or the fragment) without changing any of its other BAs (or fragments). In contrast, an inter FC may impact the size of the affected BA (BAa), its fragment (BAa in Fa: Figure 2), the sizes of the related fragments (F1,..., Fn), and consequently the size of the whole BPMN model. An inter FC will incur a substantial impact if it affects information such as trigger/precondition for execution, variants during execution of the task and problems, and sub-tasks and their sequence.

In the case of an inter FC, we identify:

- the affected BA (BAa in Fa),
- the various business activities (BA1, ..., BAn) in relation with BAa. According to the type of the relationships and also its direction (see section 3.1), we identify whether these business activities are affected by the FC or not. Once we determine all the BAa and the Number of their Relationships (NRBA) with BAa, we identify the sensitivity of BAa, ("High", "Medium", or "Small"). This latter depends on the FS(BAa) and NRBA.
- the FC Status. The status of an inter FC depends on the FS(FC), and the BAa sensitivity;

On the other hand, we identify all changed fragments related to the affected one (Fa). The same interpretation given to the affected BA (BAa in Fa) is applied to Fa. Once we determine all the affected fragments and the Number of their Relationships (NRF) with Fa, we identify the sensitivity of Fa. This latter depends on the FS(Fa) and the NRF. Then, the FC status is identified in the same manner as BA. In both cases, the status of an intra FC depends only on the FS(FC) itself.

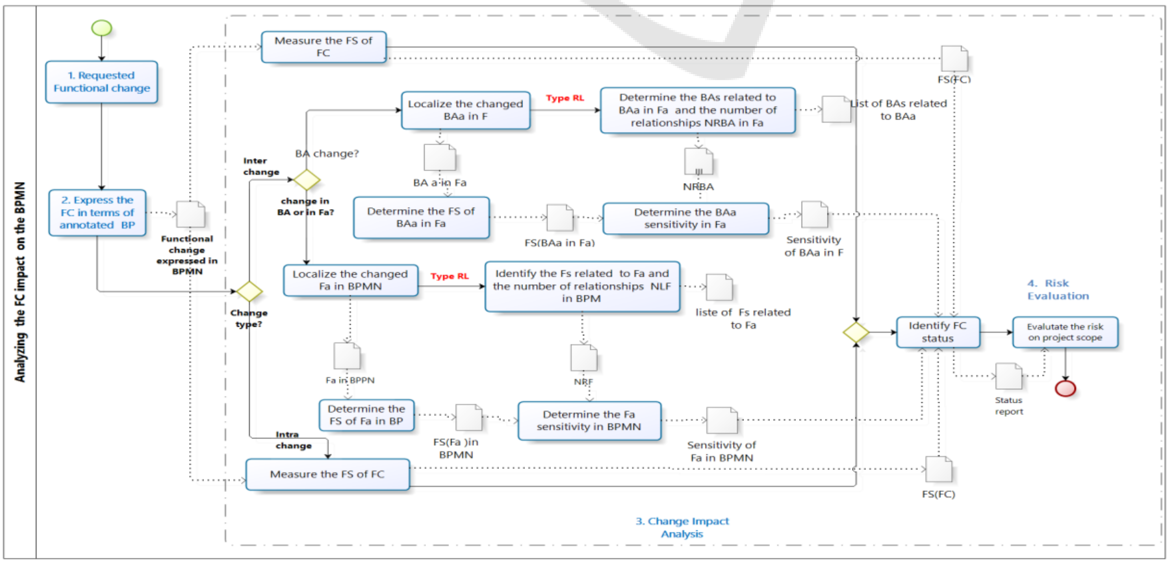


Figure 2: Phases for the FC impact analysis in BPMN models.

3.1 Sensitivity of the Changed Business Activities in a Fragment

3.1.1 Identification of the Affected BA

The identification of the BAa with its relations (BAa, BA1a,..., BAna) affected by an inter FC depends on the relationships types between BAa and (BA1, ..., BAn). In fact, we propose two types of dependency relationship: "contains" and "can_be_realized". The "contains" relationship is expressed by the default branch after a decision gateway. The "can_be_realized" relationship is considered as an alternative or exception flow (See Table 1). If two BAs (*i.e.* BAi and BAa) are related by a relationship (contains or can-be-realized), this does not mean that the FC will certainly lead to a change in BAi. In the example shown in Figure 3, if A3 is affected by a change, then this does not lead to a change in A1 despite of the relationship can-be-realized between A1 and A3. On the other hand, a BA can be affected by a FC even if it is indirectly in relation with BAa. For instance, as illustrated in Figure 3, if A1_contains_A0, and A0 is affected by the FC, then A1 is affected by this change. In addition, A2 will be affected by this change despite the absence of a direct relation between A0 and A2. This is explained by the fact that A1_contains_A0 which is connected to A2 by the "contains" relationship across A1. To handle these indirect relationships, we determine the changed business activities using Algorithm 1.

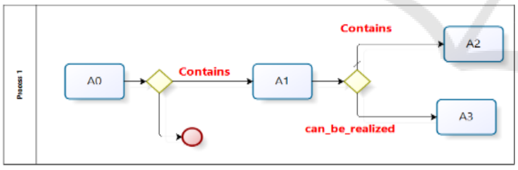


Figure 3: Identification of the affected business activities.

In Algorithm 1, the list of the BAs in the BPM are grouped in listeBA. For each BA in listeBA (BAi), we verify if it has a relationship (contains or can_be_realized) with the BAa. The listaffectedChBA includes all the BAs directly affected by the change. Then, for each BAi in listaffectedChBA, we determine the BAs in listeBA that have a relationship with BAi. Hence, the list of the indirectly affected BAs is identified.

Algorithm 1: Affected business activities identification, BAa in Fa, BPMN.

```
listAffectedChBA [], I :=0
listeBA := getBA list
for BAi in listeBA do
   if BAi_contains_BAa then
     Set listAffectedChBA [I++] := BAi
   Else if BAi_can_be_realized_BAa then
     Set listAffectedChBA [I++] := BAi
   end if; end for
for BAi in listAffectedChBA do
   listeBA := getBA list
   for BA in listeBA do
     if BA_contains_BAi then
       Set listAffectedChBA [I++] := BA
     Else if BA_can_be_realized_BAi then
       Set listAffectedChBA [I++] := BA
     end if; end for; end for
```

Let BAi ∈ (BA1, ...,BAn), according to the following measurement formulas:

$$FS(BA_i)f = FS(BA_a) + \begin{cases} FS_{cond}(condition) + FS(BA_i)_i \ \ if\ BA_i \xrightarrow{can\_be\_realized} BA_a \\ FS(BAi)_i \ \ if\ BA_i \xrightarrow{contains} BA_a \\ 0 \ \ otherwise \end{cases} \quad (1)$$

where:

- FS(BAi)i: the initial functional size of BAi;
- FS(BAi)f: the final functional size of BA;
- FS(BAa): the FS of the affected BAa;
- FScond (condition): the FS of the condition. It is equal to 1 CFP if it exists and if it has variables, otherwise it is equal to 0 CFP.

Equation (1) measure the FS(BAi) after a FC that affects BAa. For instance, if there is a "can_be_realized" relationship between BAi and BAa, then the FS(BAi)f is equal to the FS(BAi)i plus the FS(BAa) and the FScond(condition).

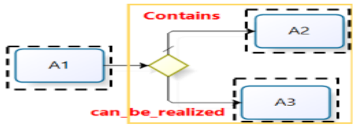
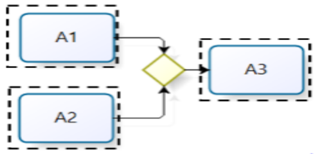
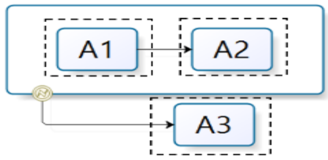
To derive the relationships "contains" and "can_be_realized" between business activities, we propose the patterns listed in Table 1.

3.1.2 Identifying the Sensitivity of BAa in Fa

Business designers/developers assess the sensitivity of a BA to help managers in determining how much attention they should devote to assess the FC request. Informally, the sensitivity of a BA expresses the degree of risk encountered on a project. In our context, we determine the sensitivity of the changed BA based on the FS(BAa) and the number of its relationships (NRBA) with other business activities.

The identification of BAa sensitivity is required in the case of inter FC that affect a BA. For this purpose, we need to determine the mean value of data

Table 1: Patterns for deriving relations between BAs.

	Pattern	Description
(a)	If there are several activities connected by an exclusive or inclusive gateway, the task/sub process preceding the gateway A1 is related to a task/sub process A2 covering the default branch by a "contains" relationship. Also, the task/sub process preceding the gateway is related to the task/sub process A3 in the alternative flow by a "can_be_realized".	
(b)	When two or more control flows are merged into one by an exclusive gateway, the activity A3 identified after the merge has to be related to the activities of each branch by means of a contain relationship. Thus, it reflects that whenever an activity A of a branch is completed the activity A following the merge has to be executed.	
(c)	If there is a sub-process in which the exceptions are handled, and the sub process's flows are covered by several activities A1 and A2, then a new Activity A3 must be created to control the exception flow. This has to be related by can be realized relationships to the activities covering sub process's flows. The exception is thus encapsulated in an activity A that extends to the others.	

movements in Fa (noted by AV_F). NRBA and the comparison of AV_F with FS(BAa) will be used as indicators of the BAa sensitivity.

$$AV_F = \frac{FS(Fa)}{n} = \frac{\sum_{i=1}^{n} FS(BAi)}{n} \qquad (2)$$

where:

- FS(Fa): the functional size of the affected fragment Fa;
- *n*: the total number of BAs in Fa (n > = 1, since there is at least one BA in F).
- FS (BAi): the functional size of the BAi

Similarly, the sensitivity of Fa in a BPMN is determined in the case of an inter FC that affect a F. It depends on the FS(Fa) and the number of relations with other fragments (NRF) in BPM. In fact, we determine the average value of data movements in BPM (AV_{BPM}) as follows:

$$AV_{BPM} = \frac{FS(BPM)}{m} = \frac{\sum_{i=1}^{m} FS(F_i)}{m} \qquad (3)$$

where:

- FS(BPM): functional size of the BPM;
- *m*: the total number of fragments in the BPM.
- FS(Fi): functional size of the Fi

On the other hand, to facilitate the interpretation of BA or Fa sensitivity, we propose the following classification: "High", "Medium", and "Small" (see Algorithm 2 and Algorithm 3). In this classification, we assume that if FS(BAa) is relatively greater than AV_F and NRBA is relatively important, then the change impact is risky, *i.e.*, it represents a "High" sensitivity in comparison to a BA with "Medium" or "Small" sensitivity.

Algorithm 2: BAa sensitivity identification, FS(BAa), NRBA.

AV_F : = FS(Fa) / n
If FS(BAa) = 1 **and** NRBA = 0 **then** Set sensitivity := Small
else if FS(BAa) > AV_F and NRBA > 1 **then**
Set sensitivity : = High
Else Set sensitivity := Medium; **Endif**

Algorithm 3: Fa sensitivity identification, FS(Fa), NRF.

AV_{BPM} := FS(BPM) / m
If FS(Fa) = 1 **and** NRF= 0 **then** Set sensitivity := Small
else if FS(Fa) > AV_{BPM} and NRF > 1 **then** Set sensitivity := High
Else Set sensitivity := Medium; **endif**

3.2 Classification of FC Status

The FC status is classified into an ordinal scale including "in scope" FC and "out of scope" FC (Fairley, 2009). If the FS(FC) = 1 CFP then the FC is classified as an "in scope" change. An "in scope" FC can be accomplished with *few* or no changes in the BPM life cycle progress. It indicates that the FC status is classified as "Moderate" or "Low". A FC that can be handled without any impact on the BP project is considered as "Low". The "Moderate" status indicating a *few* changes of the affected scope can be manifested when the FS(FC) > AV_F/AV_{BPM} and FS(FC) < = FS(BPM). It expresses minor changes in the project scope.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

The "out of scope" shows that the proposed FC affects a *big* number of data movements. It indicates that the FC status is "High". It expresses a potential change, such as the development of a new business project. It can be manifested when the FS(FC) > FS(BPM). The identification of an intra FC status, as given in Figure 4, is based only on the FS(FC). While the identification of an inter FC status, as given in Figure 5, is based on both the sensitivity of the affected business concept and the FS(FC).

Intra FC status		
In scope		Out of scope
Low	Moderate	High
FS(FC) = 1	AV_F/AV_{BPM} < FS(FC) <= FS(BPM)	FS(FC)> FS(BPM)

Figure 4: FC status evaluation in the case of an intra FC.

		Inter FC Status		
		In scope		Out of scope
The affected business concept Sensitivity	High	Moderate	Moderate	High
	Medium	Low	Moderate	High
	small	Low	Moderate	High
		FS(FC)= 1 CFP	AV_F/AV_{BPM}< FS(FC) <= FS(BPM)	FS(FC) > FS(BPM)

Figure 5: FC status evaluation in the case of inter FC.

In Figure 5, each element in the matrix represents a different value of FS(FC) and BAa/Fa sensitivity. It is divided into "Red", "Yellow" and "Green" zones which represent respectively High, Moderate and Low status. The red zones are centred on the right corner of the matrix (FS(FC)> FS(BPM)). While, the green zones are centred on the left corner (when FS(FC) =1 CFP) except when the sensitivity of the affected business concept is high and the FS(FC) =1 CFP. The yellow zones extend specially the middle of the matrix.

In summary, the effort required to fulfill a FC with a status represented by a red zone in the above matrices is more important than the effort required to answer a FC with a status represented by green zones.

3.3 Sizing the Business Concept after a FC Request

To measure the FS of a business concept, the mapping between the business model concepts and those of the COSMIC method is required. Therefore, we can determine not only the size of the added data movements in a business concept but also their types based on the task type (see Table 2).

Table 2: Alignment of COSMIC data movement types with BA elements.

Data Movement Types	BA Elements
Entry	Task: Type = "ActiveREQ", Precondition, Event
eXit	Task: Type = "ActiveREP"
Read	Task: Type = "ActiveRET"
Write	Task: Type = "ActivePER"

3.3.1 FS(BAa) in FA after an Inter/Intra FC

Table 3 presents the FS(BAa) after an intra FC. It is expressed when it affects one of the BAa elements. The impact of an intra FC on the FS(BAa), is given in Table 3, where:

- FS(BAi): is the FS of the affected business activity before the FC.
- FS(BAf): is the FS of the affected business activity after the FC.

Table 3: FS(BAa) after an intra FC-FC in BAa elements.

<table>
<tr><th colspan="3">Functional Change Concerning Elements in BAa</th></tr>
<tr><td>Addition (SP: Pre-ondition) | (Task: re-condition) | (SP: event)
FS(BAf) = FS(BAi) +1CFP</td><td>Deletion (SP: Pre-ondition) | (Task: re-condition) | (SP: event)
FS(BAf) = FS(BAi) – 1CFP</td><td>Modification (SP: Pre-condition) | (Task: Pre-condition) | (SP: event)
FS(BAf) = FS(BAi)</td></tr>
<tr><td>Addition (Task: Type = Passive)
FS(BAf) = FS(BAi)</td><td>Deletion (Action: Type = Passive)
FS(BAf) = FS(BAi)</td><td>Modification (Task: Type = Passive)
NewType! = Passive FS(BAf) = FS(BAi) +1CFP</td></tr>
<tr><td>Addition (Task: Type! = Passive)
FS(BAf) = FS(BAi) +1CFP</td><td>Deletion (Task: Type! = Passive)
FS(BAf) = FS(BAi) – 1CFP</td><td>Modification (Task: Type! = Passive)
if NewType = Passive FS(BAf) = FS(BAi) −1CFP else FS(BAf) = FS(BAi)</td></tr>
<tr><td>Addition (Task: Int-datagroup |Task: Out-datagroup) are in the same object of interest) and (Task: Type !=Passive)
FS(BAf) =FS(BAi)</td><td>Deletion (Task: Int-datagroup |Task: Out-datagroup) are in the same object of interest) and (Task: Type! =Passive)
FS(BAf) = FS(BAi)</td><td>Modification Task: Int-datagroup |Task: Out-datagroup) and (Task: Type! =Passive)
FS(BAf) = FS(BAi)</td></tr>
</table>

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

We kept only the elements that may affect the FS(BAa) such as: SP: Pre-condition, Task: Pre-condition, SP: event, Task: Type, Task: Int-datagroup, Task: Out- datagroup. For example, in the case where BAa is in a fragment, the addition or deletion of a task or SP having a pre-condition will lead to a change in the FS(BAa). However, the modification of a task or SP having a pre-condition does not lead to a change in the FS(BAa).

Moreover, the addition or the deletion of a task with Type = Passive does not lead to a change in the FS(BAa). In this case, the modification of a task with Type = Passive by a Task with NewType != Passive increase the FS(BAa) by 1CFP. Whereas, the addition or the deletion of a Task with Type != Passive leads to a change in the FS(BAa). In this case, the modification of a Task with Type != Passive by a Task with NewType = Passive will decrease the FS(BAa) by 1 CFP.

The deletion or the modification of input or output data group related to the same business object with task Type !=Passive, does not lead to a change in the FS(BAa).

Table 4 presents an inter FC applied on the business activity. It is expressed when the affected BAa is related to other business activities in the same fragment. For example, the addition, deletion or modification of a relationship outgoing from the affected business activities (BAa) has an impact on the related (BAn) in the same fragment.

Table 4: FS(BAa) after an inter FC.

<table>
<tr><th colspan="6">FC Concerning Direct Relationship with BAa</th></tr>
<tr><td colspan="2">Addition (BAa_can be realized_ BAn)
FS(BAf) = FS(BAi) + FS(BAn) + FS(cond)</td><td colspan="2">Deletion (BAa_can be realized_ BAn)
FS(BAf) = FS(BAi) -FS(cond) - FS(BAn)</td><td colspan="2">Modification (BAa _can_be _realized_ BAn)
if NewR= contains (BA, BAn)
FS(BAf) = FS(BAi) + FS(BAn) - FS(cond)</td></tr>
<tr><td colspan="2">Addition (BAa_contains _BAn)
FS(BAf) = FS(BAi) + FS(BAn)</td><td colspan="2">Deletion (BAa_contains _BAn)
FS(BAf) = FS(BAi) − FS(BAn)</td><td colspan="2">Modification (BAa_contains_BAn)
if NewR= can_be_realized(BAa, BAn)
FS(BAf) =FS(BAi) + FS(BAn) + FS(cond)</td></tr>
<tr><th colspan="6">FC Concerning Indirect Relations with BAa</th></tr>
<tr><td colspan="3">(BAa_contains_BAn)
FS(BAf) = FS(BAi) + FS(BAnf)</td><td colspan="3">(BAa_ can_be_realized _BAn)
FS(BAf) = FS(BAi) + FS(BAnf) + FS(cond)</td></tr>
</table>

The impact of an inter FC on the FS(BAa), is given in Table 4, where:

- FS(BAi): FS of BAa before the FC.
- FS(BAf): FS of BAa after the FC.
- FS(BAnf): FS of BAn after the FC.
- FS(BAn): FS of of BAn before the FC.

According to formula 1, the addition or the deletion of the relation "can_be_realized" between BAa and BAn modify respectively the FS(BAf) by adding/deleting the FS(BAn) and the FS of the condition to/from the initial FS(BAi). The modification of the "can_be_realized" between BAa and BAn by "contains" leads to a change in the FS(BAa). Hence, the FS(BAf) is equal to FS(BAi) plus the FS(BAn) minus the FS of the condition (FS(cond)).

In the case of "contains" relationship, this later is expressed by a default flow, which does not specify any condition. For data-based exclusive or inclusive gateways, one type of flow is the default condition flow. This flow will be used only if all other outgoing conditional flow are not true at runtime. The addition or the deletion of this relationship between BAa and BAn modify respectively the size of BAa by adding/deleting the FS(BAn) to/from FS(BAi). The modification of the "contains" relationship between BAa and BAn by a "can be realized" relationship will lead to a change in the FS(BAa). Hence, the FS(BAf) is equal to its FS(BAi) plus the FS(BAn) and FS(cond).

On the other hand, if a change affects a BAn which has a relationship with BAa, then the FS(BAa) is changed as given in Table 4. For instance, if BAa_contains_BAn, and the FS(BAn) is changed, then the FS(BAa) is changed. The FS(BAf) is equal to the FS(BAi) plus the FS(BAnf). The same thing is applied to the "can_be_realized" relationship. Thus, if BAa_can_be_realized_BAn, and the FS(BAn) is changed, the FS(BAf) is equal to the FS(BAi) plus the FS(BAnf) and FS(cond).

3.3.2 FS(Fa) in BPM After an intra/inter FC

A intra FC in a Fragment can be expressed in two cases: on the one hand, it is applied on the addition, deletion or the modification of a BA in the affected fragment Fa, and this BA has no impact on the other business activities in Fa. On the other hand, the intra FC is applied on fragment's elements. In fact, if a change request proposes the addition, or the deletion of a data object, a message flow with data group, or sequence flow with condition, then the FS of the associated fragment F will be changed as given in

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 5. For example, the addition of a data object will increase the FS(F) by 1 CFP. However, the modification of a data object, a message flow with data group, or sequence flow with condition, does not lead to a change in the FS(F).

Table 5 summarizes the impact of a FC where:

- FS(Fi): the FS of the Fa before the change.
- FS(Ff): the FS of the Fa after the change.
- FS(BA): the FS of the BA.

Table 5: FS(F) after an intra FC.

Functional Change Concerning BA in Fa		
Addition (BA in Fa)	**Deletion (BA in Fa)**	**Modification (BA in Fa)**
FS(Ff)= FS(Fi) + FS(BA)	FSf(Ff) = FSf(Fi) –FS(BA)	See Table 3
Functional Change Concerning Elements in Fa		
Addition (data Object) FS(Ff) = FS(Fi) + 1CFP	**Deletion (data Object)** FS(Ff) = FS(Fi) - 1CFP	**Modification (data Object)** FS(Ff) = FS(Fi)
Addition (Message flow with data group) FS(Ff) = FS(Fi) + 1CFP	**Deletion (Message flow with data group)** FS(Ff) = FS(Fi) - 1CFP	**Modification (Message flow with data group)** FS(Ff) = FS(Fi)
Addition (Sequence flow with condition) FS(Ff) = FS(Fi) + 1CFP	**Deletion (Sequence flow with condition)** FS(Ff) = FS(Fi) - 1CFP	**Modification (Sequence flow with condition)** FS(Ff) = FS(Fi)

The inter FC is expressed when a change in a fragment Fa affects another fragment. It is possible to apply Table 4 in this case by replacing: BAa with Fa, BAi with Fi, BAnf with Fnf, BAf with Ff and BAn with Fn.

3.3.3 FS(BPM) after an Inter/Intra FC in BAa/Fa

The FS of a BPM is measured as the sum of the FS of all the fragments it includes. Thus, each inter/intra FC affecting either a fragment Fa or a BAa in Fa, will certainly lead to a change in the FS of the BPM.

3.4 Prioritizing FC Requests

If more than one FC is proposed simultaneously, we identify which FC should be realized before the other one. In order to define how the change can be implemented sequentially or in parallel, we propose a heuristic algorithm for assigning priorities to functional changes based on their status identified separately.

H 1: If there is a "contains" relationship between two BA/F (BA1/F1 contains the behavior of BA2 /F2) and two functional changes are proposed where FC1 affects BA 1/F2 and FC2 affects BA 2/F2, then FC2 must be implemented before FC1.

H2: If there is a "can_be_realized" relationship between two BAs/Fs (BA 1/F1 can be realized after BA 2/F2) and two functional changes are proposed where FC1 affects BA 1/F1 and FC2 affects BA2/F2, then implement FC2 before FC1.

H3: If the inter change proposes the addition or the deletion of a "contains" relationship, an intra FC change is implemented before the inter change.

H4: If the inter change proposes the addition or the deletion of a "can_be_realized" relationship, implement inter change before the intra change.

Algorithm 4: Prioritization of changes G(V, E).

input:– G =(V, E); V set of nodes each representing a FC, E is the edges representing the FC inter-dependency.

```
if PR-Req are given then
allChanges := getChangeOrder (V,  PR-Req)
else allChanges := getChangeOrder (V, FC status)
endif
while allChanges !=null do
forFCi in allChanges do
for FCi+1 in allChanges do
if ∃ dependRel (FCi, FCi+1) then
dependRel := true
 break; endif
end for
if dependRel = true then
SubChange := least cost path (G⁻¹ FCi)
for FCj in SubChange do perform (FCj)
remove (FCj from allChanges)
endfor
else perform (FCi);  endif;
endfor; end while
```

Based on the above heuristics, we build a directed graph G = (V, E), where V represents nodes expressing the changes and E represents directed edges expressing the dependency between the changes. For each node (*i.e.* FC), we can identify its status as explained in section 3.2. In addition, to prioritize a set of FCs, we propose algorithm 4 which takes as input the list of the change requests. For each FC, we identify the preference of the change requester (PR_Req) and the FC status. The algorithm generates the scheduling of the changes based on the above heuristics. As proposed in (Fairley, 2009), user requirements are classified into Essential, Desirable, or Optional. By referring to this classification, we identify three categories of PR_Req (Desirable, Essential, Optional). A desirable FC is a change that adds value to the business process. It is implemented

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Name of the fragment: F2
Purpose of the fragment: Manage rented devices

Trigger/Precondition for execution: <Rented device> <type: ActiveREQ>

Activity and their sequence(Main scenario BA22)

Begin at <cancel rental device contract>
<cancel rental device contract> [<drop activity>] <type: ActiveREQ>
End

Variant during the execution of the activity and problems (Alternative scenarioBA21)

Begin at <Device operating control>
Trigger/Precondition for execution: <High activity><type: ActiveREQ>
*Relationships:*BA23
<**contains**>:*parallel bloc:*use rented device for flight, recover devices list
<**can_be_realized**>:*parallel bloc:*use rented device for flight, maintain devices
Begin
Activity and their sequence (Main scenario BA23)
[<device does not need maintenance>] <use rented device for flight> <type: ActiveREQ>, <recover devices list> <type: ActiveREQ and type: ActiveRET> [<Devices list>]
Variant during the execution of the activity and problems (Alternative scenario B23)
[<device need maintenance>] <use rented device for flight> <type: ActiveREQ>, <maintain devices> <type: ActiveREQ and ActivePER> [<Maintained Devices list>]
End
End

Variant during the execution of the task and problems (Error scenario)

Begin
<none>
End

Critical situations of execution of the activity

Special requirement: <none>

Figure 6: Detailed description of F2.

after satisfying all the essential changes and if there is no or few changes in the initially estimated time and budget. An essential FC is an obligatory FC that must be implemented even if it will lead to an extra-effort and a "*big*" impact on the business process project; otherwise, the final product will not satisfy the user needs. An optional FC is a change that might be implemented if there is sufficient time and resources after satisfying the essential and desirable changes.

In Algorithm 4, all changes are first grouped in the allChanges based on the PR_Req order. In the case where the change requester does not specify any preference, then the changes are ordered according to their FC status (lines 1–4, Algorithm 4). Consequently, the change with the lowest FC status is the first to be implemented. Changes are then implemented with respect to the order provided in the list of allChanges. However, if there are inter-dependencies between changes, then we determine the sub-graph including the minimum cost start from the first FC in the list of allChanges (lines 5–13). We execute all the changes identified in the sub-graph (line 14). Then, we remove these changes from the list of allChanges (line 15). We repeat these steps until the end of the list of allChanges.

4 CASE STUDY AND THREATS TO VALIDITY

In this section, we illustrate the applicability of our proposed approach through the "Airline Company" BPMN as described in Figure 7 and the documented FC2 in Figure 6. Afterward, we discuss threats to validity of our study.

4.1 Measurement Illustration

Based on the BPMN decomposition approach (Khlif *et al.*, 2017), the "Airline Company" BPMN is presented as a series of fragments (F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11 and F12). Each fragment may contain one or more business activities documented using our expanded description. For example, Figure 6 presents the F2 description.

The FS of the "Airline Company" BPMN model is determined by aggregating the FS of all fragments which are described by the proposed extended textual description. The total FS(BPM) is equal to 22 CFP.

For instance, Table 6 presents the functional size of the affected fragments (F2, F8 and F9) before and after the change. In order to illustrate the proposed change impact analysis in the business concept, we propose:

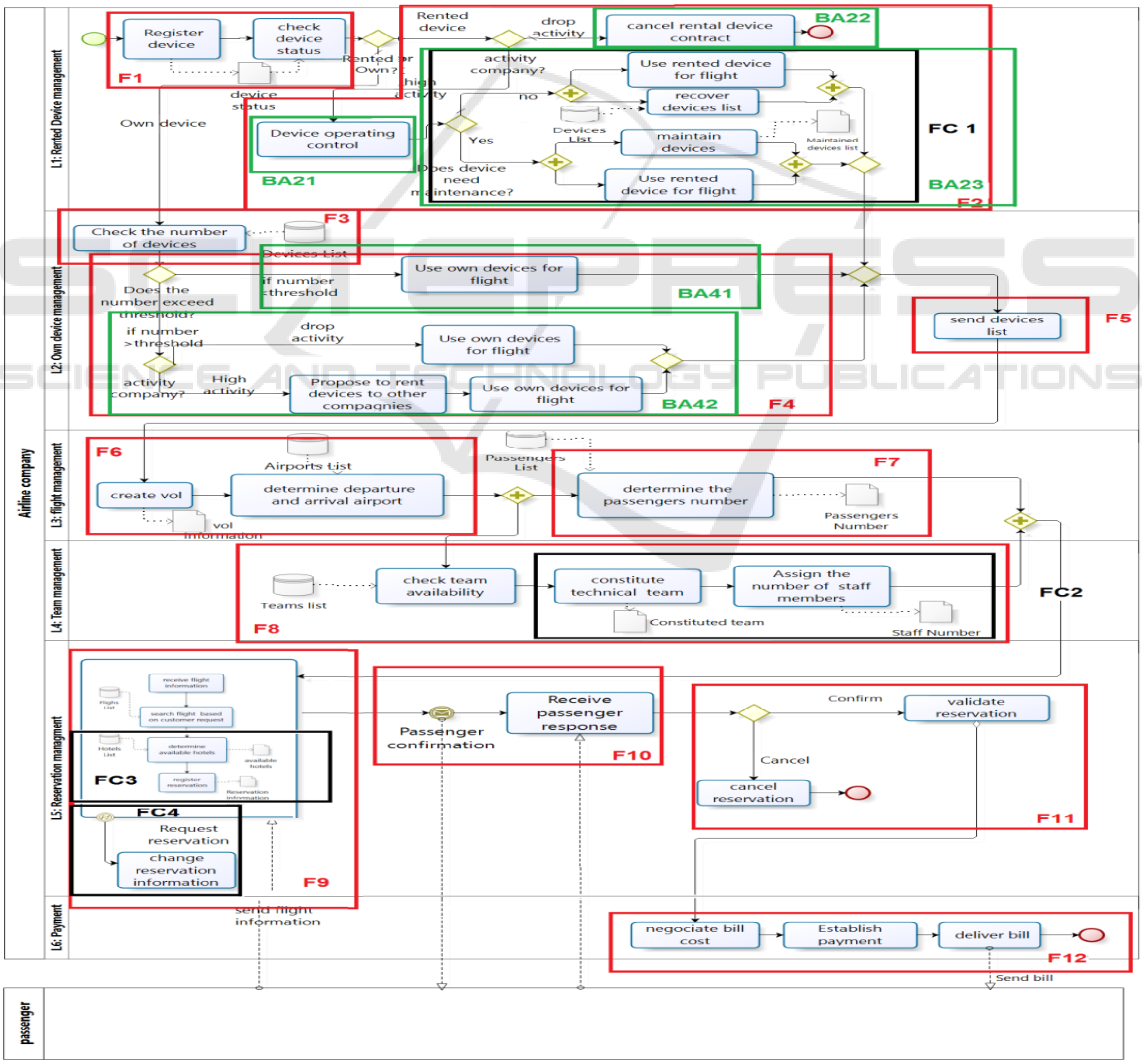
- an inter FC in F2 referred to FC1 "adding BA23" (Figure 7) including the "contains" and "can_be_realized" relationships. FC1 has an impact not only on the FS of the affected fragment F2, but also on the FS of fragment F5 as shown in Table 1.b.
- FC2 expresses an intra FC by adding two sequential tasks with Type = ActivePER. These added tasks have no impact on any other BAs/fragments.

- FC3 is an intra FC that proposes to add two tasks in F9 with Type = ActiveRET or Type = ActivePER.
- FC4 is an inter FC since there is a "can_be_realized" relationship expressing that an exception is related to the activities covering the sub-process "Request reservation" (See Table 1)

As presented in Table 6, all the requested changes (FC1, FC2, FC3 and FC4) are "in-scope". For example, regarding FC1, AV_{F2} = 1 CFP and the FS(FC1) = 4 CFP. Thus, as presented in Figure 5, this FC is considered as "in-scope" since $AV_{F2} < FS(FC1) <= FS(BPM)$. It is considered as a "Moderate" FC.

The measurement results in Table 6 show that all the proposed changes are "in-scope". The acceptance of an "in-scope" FC can be accomplished with no disruption to the planned work activities. Thus, no modifications are needed in the schedule, budget and resources to handle the proposed changes. However, in the case of "out-of-scope" changes, major modifications are required in the schedule, budget and resources. Hence, to take the appropriate decisions regarding an "out-of-scope" change, it is required to estimate the effort and resources to implement the change. Consequently, in the illustrative example, the decision-makers will approve all the FC requests since they are "in-scope" changes.

When prioritizing the change requests, the preference of the change requester is taken into account since all the changes have the same FC status "in-scope". By applying the Algorithm 4, the approved change requests are classified according to



SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Figure 7: "Airline company" model after the change.

Table 6: Measurement results for the "Airline company" model.

Fragment		Functional Sub-process	Before the change	After the change	FS(FC)	FC Type	FC status
F2	BA21	Device operating control	1	1	0	none	AV_{F2} = 3CFP/2 = 1 CFP FC1 status = "In-scope"
	BA22	Cancel rental device contract	2	2	0	none	
	BA23	**Use rental device for flight** **Recover device lists**	-	2	**4 CFP**	Inter FC1: adding BA23 in F2	
		Maintain device **Use rented device for flight**	-	2			
Total			**3 CFP**	**7 CFP**	**4 CFP**		
F8		Check team availability	1	1	**2 CFP**	Intra FC2: adding tasks in F8	AV_{F8} = 1 CFP/ 1 = 1 CFP FC2 status = "In-scope"
		Constitute technical team	-	**1**			
		Assign the number of stuff members	-	**1**			
Total			**1 CFP**	**3 CFP**	**2 CFP**		
F9		Receive flight information	0	0	**FS(FC3) = 3 CFP**	Intra FC3: adding tasks in F9	AV_{F9} = 1 CFP / 1 = 1CFP FC3 status = "In-scope"
		Determine the available hotel	-	**2**			
		Search flight based on customer request	1	1			
		Register reservation	-	**1**			
		Send flight information	1	1			
		Change reservation information	-	**1**	**FS(FC4) = 1 CFP**	Inter FC4: adding an exception	AV_{F9}= 1 CFP / 1 = 1CFP FC4 status = "In-scope"
Total			**2 CFP**	**6 CFP**	**4 CFP**		

the PR_Req. Hence, the designers/developers will take these changes starting by the "Essential" FC, then the "Desirable" FC, and finally the "Optional".

4.2 Threats to Validity

In our study, threats to validity are relevant to internal validity, external validity and construct validity (Wohlin *et al.*, 2000).

- Internal validity threats stem from the determination of change status which can be expressed in two cases. The first one is based on the on the FS(FC) in the case of intra FC. The second one is based on the FS(FC) and the BAa/Fa sensitivity in the case of inter FC. However, the determination of the change status neglects other factors such as change type, the required effort to handle the change request, etc. And the deletion of a business concept should be analyzed if it is developed or not yet developed. In the first case, the deletion saves development time. In the second case, it causes more effort.
  Furthermore, many situations do not require the same level of details of the proposed ordinal scale about FC status. In fact, for an approximate identification of FC status, three categories will be enough. However, a more detailed identification of FC status will requiremore detailed scale.

Besides, the request for functional changes must be determined, otherwise the classification induce an analysis that does not express the real importance of the change.

The second issue related to internal validity is the use of a high-level business activity that does not address the architectural aspects in the code. Thus, some detailed activities may appear in the implementation phase without being identified in the requirement phase.

- External validity: The main threats to the external validity expresses the insufficiency of data that makes difficult the generalization of this study results. As it is not possible to collect data from companies, our research study is exploratory. Our work can be generalized to take into account the development and the requirement phases. Furthermore, the COSMIC method can be used in all the business process life cycle phases. In our study, COSMIC is used in the requirements phase.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Moreover, the required level to apply COSMIC was a detailed level of a business concept. Whereas, in practice, we do not always have a detailed description of a business concept. Further studies will be needed to address FC status in a high level of abstraction.

- The threats of construct validity investigate the potential of applying this study in practice. One predicament to our study stems from the lack of feedback about the FC status it determines. Indeed, the judgment of how important is a FC depends on the specific expertise of the people participating in the change impact analysis, etc. This judgment will improve the change status classification and the ratio AV_{FC}.

5 CONCLUSIONS

The presented work proposed measuring the functional changes in BPMN models using COSMIC method (COSMIC 2017). The proposed FC measure provides a firmer basis for analyzing the impact of change on the functional size of business concepts in terms of CFP units. Based on the functional size of the FC and the sensitivity of the affected business concepts, we determine the status of the FC. FC status can be used to identify the degree of risk on the project scope. It helps both business process developers and manager in making decisions to accept, deny or defer a change request.

While the illustrative example showed the feasibility of the approach, it also confirmed our need to conduct empirical studies to improve the thresholds used to determine the mean value of data movements. To prepare for such empirical studies, we are in the process of implementing CASE tools to get automatically indicators about how to manage the risk of a FC during the BP project development.

Other factors may interfere in identifying the importance of a FC such as the preference of the change requestor, the effort required to answer the change. Moreover, when the functional size is the input for the effort estimation models, it is possible to estimate the effort required to implement the change using one of the estimation tools supporting COSMIC method.

REFERENCES

COSMIC 2017. Common Software Measurement International Consortium. 2017. COSMIC Functional Size Measurement Method, Version 4.0.2.

Fairley, R.E., 2009. *Managing and Leading Software Projects*. Wiley-IEEE Computer Society Press.

Ferme, V., Ivanchikj, A., Pautasso, C., 2016. Estimating the Cost for Executing Business Processes in the Cloud. In *Proc. of Business Process Management Forum - BPM Forum 2016*, Brazil, September pp. 72-88.

ISO/IEC 19510. 2013. Information technology -- Object Management Group Business Process Model and Notation. 2013.

ISO/IEC 14143-6:2012. 2012. Information technology – Software measurement – Functional size measurement – Part 6: Guide for use of ISO/IEC 14143 series and related International Standards.

Khlif., W, Haoues M., Sellami A., Ben-Abdallah H., 2017. Analyzing Functional changes in BPMN models using COSMIC. In *ICSOFT'17,Proc. of International conference on software engineering and applications*, Portugal, July, pp. 265-274.

Lauesen, S., 2002. *Software Requirements: Styles and Techniques.* Addison-Wesley, London.

Nezhad, H.R.M., Saint-Paul, R., Casati, F., Benatallah, B., 2011. Event correlation for process discovery from web service interaction logs. In *the Inter Journal on Very Large Data Bases,20(3*), pp.417–444.

Uronkarn, W., Senivongse, T., 2014. Change Pattern-Driven Traceability of Business Processes. In *IMECS'14 ,InterMultiConference of Engineers and Computer Scientists*. March, Hong Kong, pp 441-455.

Van der Aalst, W.M.P. Process Mining: Discovery, Conformance and Enhancement of Business Processes. 2011. Springer, Germany, pp 1-352.

Wang, Y., Yang, J., Zhao, W., 2012. Change Impact Analysis in Service based Business Processes. In *SOCA'12, International Conference on Service-Oriented Computing and Applications*, pp. 131-149.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2000. Experimentation in Software Engineering: An Introduction, 2000.