APPLYING SOFTWARE PROCESS METRICS IN BUSSINESS PROCESS MODEL

Elvira Rolón [1], Francisco Ruiz [2], Félix García [2], Mario Piattini [2]
[1] *Autonomous University of Tamaulipas*
University Center Tampico-Madero, 89336 Tampico, Tamps. México
E-Mail : erolon@proyectos.inf-cr.uclm.es
[2] *University of Castilla La Mancha*
Paseo de la Universidad No. 4,13071 Ciudad Real, Spain
E-Mail : {Francisco.RuizG, Felix.Garcia, Mario.Piattini}@uclm.es

Abstract: In this paper we define a set of metrics for the evaluation of business process models. The proposal is based on the FMESP framework, which aims to integrate the modeling and measurement of software processes. FMESP includes a set of metrics to provide the quantitative basis necessary to know the maintainability of the software process models. This proposal has been used as the starting point to define a set of metrics for the evaluation of the complexity of business process models defined with BPMN. To achieve this goal, the first step has been to adopt the metrics of FMESP, which can be directly used to measure business process models, and then, new metrics have been defined according to the particular aspects of the business processes and BPMN notation

Key Words: Business Process, BPMN, Metrics, Conceptual Models, Software Process..

.

1. INTRODUCTION

Software processes and business processes present certain similarities. The most common is that both try of capturing the main characteristics of a group of partially ordered activities carried out to achieve a specific goal, that they are those of obtaining a product software (Acuña and Ferré 2001) or a satisfactory results (generally a product or service) for the customer and other stakeholders of the process respectively
(Sharp and McDermott 2000).
As regards the modelling of both types of process, these also have certain characteristics in common. When talking about the modelling of the software process, it should be pointed out that this refers to the definition of the processes as models, and Finkelstein et al. (1992) defines it as an abstract description of the activities by which the software is developed, focusing on models that are executable, interpretable or able to be accede to automated reasoning. Adding to this specification, Curtis et al. (1992) define some of the specific goals and benefits of modelling the software process, such as: 1. Ease of understanding and communication, 2. Process management support and control, 3.- Provision for automated orientations for process performance, 4. Provision for automated execution support, and 5. Process improvement support. On the other hand, in business process modeling the main concept is the business process, which describes the activities involved in the business and how they relate to and interact with the necessary resources to achieve a goal for the process (Beck et al. 2005; Erickson and Penker 2000). Business process models describe how a business works, or more specifically, how they accomplish missions, activities, or tasks (Dufresne and Martin 2003). Some specific goals of business process modelling are:
(Beck et al. 2005; Erickson and Penker 2000): 1. To ease the understanding of the key mechanisms of an existing business, 2. To serve as the basis for the creation of appropriate information systems that support the business. 3. To improve the current business structure and operation, 4. To show the structure of an innovated business, 5. To identify outsourcing opportunities and,

6. To facilitate the alignment of business specifications with the technical framework that IT development needs.

Something that particularly characterizes software and business processes is the fact that for more than one decade and, as result of the confrontation of the new technologies, more competitive markets, business environments in constant change and requirements for customer's satisfaction, the developers and software presidents, as well as people of business and the organizations in general have been focused in their processes like a reference point to survive and prosper (Florac et al. 1997). It has increased the necessity for analyzing, evaluating, measuring and improving the processes.

As a result of the situation outlined above, the modelling of business processes in particular is becoming increasingly popular in the last years. A current solution is the business process management through the BPMS (Business Process Management Systems) which that offer benefits tactical and strategic to the enterprises and it has been popular in the business market (Mc. Daniel 2001). However, a process is in general very complex and embraces decisions at very different levels.

In this work, our target is to focus on the conceptual level of the business process modelling, since we believe that it is one of the point key to obtain models of quality that can serve as support for an effective maintainability and management of business processes. Wedemeijer and de Bruin (2004) defines the conceptual process model as an abstracted model of the business process whose purpose is to outline all actions indispensable to produce all of the essential results in a customer-triggered business process, regardless how, when, by whom or by which means these outputs are produced. Conceptual process models show what a system does or must do, they are independent of implementation (i.e., they depict the system independently of any technical implementation) and the language to perform it is usually a graphic language. This is the case of Business Process Modelling Notation (BPMN) (BPMI 2004), which is the new standard for modeling business processes and Web services processes, proposed by the Business Process Management Initiative (BPMI).

The first goal of BPMN is to provide a notation that can be easily understood by all business users, from the business analysts to technical developers and business people (White 2004). To achieve this, BPMN facilitates the modeling of high-level business process through a Business Process Diagram

(BPD), which is based on a flowcharting technique tailored for creating graphical models of business process operations.

BPMI tries to unify the diversity of proposals and terminology related to business process modelling by means of the standard notation BPMN in the same way as the SPEM (OMG 2002) specification tries in software process modelling field. SPEM is a generic metamodel for the definition of software processes and it is based on UML metamodel, which means that it inherits its expressiveness to represent descriptive software process models.

In this paper, we describe a proposal of metrics for business process models represented in BPMN. This proposal is based on the application and adaptation of the measurement framework of FMESP, which includes metrics for the evaluation of software process models defined with SPEM, to business process models.

The rest of the paper proceeds as follows: In section 2, the topic of metrics for conceptual models of software processes is tackled and then, the basic elements of the BPMN metamodel are defined. In section 4 the metrics defined in FMESP and the adapted proposal for business process models are presented and then, the new metrics defined in particular for business process models are described. In Section 6, an example of calculation of the metrics is provided in which a business process model represented in BPMN is measured and finally, some conclusions and further works are outlined.

2. METRICS FOR CONCEPTUAL MODELS OF SOFTWARE PROCESS

In the same way as it has happened to business processes, the software process research has also acquired big dimensions in the last years due to the growing complexity of the software systems. It is due to the processes need to continuously undergo changes and refinements to increase its ability to deal with the requirements and expectations of the markets and the stakeholders of the company. Hence, processes need to be continuously assessed and improved and it has motivated a wide range of projects devoted to the creation of quality models and methods for software process improvement (Fuggetta 2000).

In our work we have based on the FMESP proposal (García et al. 2005), which consists of a framework for the modeling and measurement of software process. FMESP is based on the idea that it is necessary to carry out a good administration of the software processes with the purpose of obtaining software products with quality, and such management considers it in an integrated way by embracing two important aspects: the process modeling and process evaluation. As a result, it provides the conceptual and technological support for the modeling and measurement of software processes in order to promote their improvement.

For the evaluation of the software process, FMESP includes a set of metrics, which measures the structural complexity of SPMs. The aim is to evaluate the influence of the structural complexity of the software process models on their maintainability. The FMESP metrics have been defined at two different scopes: model scope, to evaluate the overall structural complexity of the model and; level scope, to evaluate the concrete complexity of the fundamental elements of the model, namely activities, roles and work products. The model scope metrics are shown in the Table 1.

The FMESP metrics were defined by analysing the SPEM metamodel (OMG 2002) and they are grouped in: base measures which were obtained by counting the number of significant SPEM metamodel constructors and their relationships and; derived measures, which are obtained as a result of applying measurement functions on another base and/or derived measures. An example of a software process model represented with SPEM with the calculation of the model scope metrics is shown in the Figure 1.

With the aim to establish which metrics are useful SPMs maintainability indicators, a family of experiments was carried out (Canfora et al. 2005). The FMESP metrics defined to evaluate the complexity of concrete elements in the software process model (activities, work products and process roles) are not described here due to they are out of the scope of this paper.

3. ELEMENTS OF BPMN

Business processes models (BPMs) have a wide range of uses such as the support to re-engineering processes, simulation or as base in order to develop systems to automate the processes of the model. Besides, BPMs can be also created or presented using many different methodologies. These methodologies are very different among themselves, since each one has a different way to see the processes depending on the purpose for which they were created (Dufresne and Martin 2003).

Among the methodologies mentioned in the literature, the following deserve special attention for the modeling of business processes: IDEF 0 (FIPS 1993), IDEF 3

(Mayer et al. 1995), UML (Erickson and Penker 2000), UML 2.0 (OMG 2003), and BPMN (BPMI 2004). The latter is the notation standard on which our work of evaluation of business processes models is based at conceptual level.

BPMN provides a graphical notation for expressing business processes in a Business Process Diagram (BPD), based on a flowcharting technique tailored for creating graphical models of business process operations that allows the easy development of simple diagrams. At the same time it is able to handle the complexity inherent to business processes (Owen and Raj 2003). Another important characteristic of BPMN is that the XML languages designed for the execution of processes of business such as BPEL and BPML can be visually expressed with a common notation.

The BPD is composed of two basic categories: the first one composed of core elements with which is possible to develop simple process models and; a complete list of elements that allows the creation of complex or high-level business process models. The four basic categories of elements are Flow Objects, Connecting Objects, Swimlanes and Artefacts. The symbols of the core elements are shown in the Table 2.

Inside each category of the core elements shown in Table 2 there is a more extensive list of business process constructors in the BPMN notation.

4. APPLYING THE PROPOSAL FMESP TO MODELS BPMN

The objective with the definition and validation of the metrics in FMESP is to determine a group of useful indicators of the maintainability of software process models by evaluating their structural complexity. The proposal of FMESP is based on the fact that the research on software process measurement had been centered in the study of the results of the execution and not in the repercussion that could have the structural complexity of the processes models in its quality.

A similar situation happens in the area of business processes modelling. As a result of the research on the side of business people, in the literature we can find diverse proposals for the evaluation of processes, mostly from the point of view of the results obtained in their execution. It means that the aspects evaluated in business process measurement research mainly belongs to a process execution level, where two categories of metrics are even contemplated: operational and structural (Tjaden 1999). On the other hand, there are also proposals or frameworks in order to evaluate the quality of business processes modeling techniques (Hommes and van Reijswoud 2000)

Considering our interest in evaluating the business process by starting from the model that represents it in a conceptual level, our work recaptures the FMESP proposal but adapting and extending it to business process models. To achieve it we have defined a set of metrics to evaluate the structural complexity of business process models in a conceptual level. The goal is to have empirical evidence about the influence that the structural complexity of business models can have on their maintainability. It can provide companies with the quantitative basis necessary to develop more maintainable business process models. The first step to achieve this goal is to define a set of suitable metrics for the evaluation of the structural complexity of business models. This definition has been based on the elements that compose the BPMN metamodel. These metrics have been grouped in two main categories: Base and Derived Measures.

The base measures have been defined by counting the different kind of elements that compose a business process model represented with BPMN. In Table 3, the

base measures defined for the constructor "Event" in the BPMN metamodel are shown.
As we can observe in table 3, the base measures defined for all the triggers of events are included
(Start, Intermediate and End). They belong to the BPD "Flow Objects" category. With these, the cause of the beginning or ending of a flow within the model can be identified, as well as those elements that modify the flow at an intermediate point of the same.
In the Table 4, the base measures for the BPMN metamodel element "activity" are shown.
As we can observe in table 3, the base measures defined for all the triggers of events are included
(Start, Intermediate and End). They belong to the BPD "Flow Objects" category. With these, the cause of the beginning or ending of a flow within the model can be identified, as well as those elements that modify the flow at an intermediate point of the same.
In the Table 4, the base measures for the BPMN metamodel element "activity" are shown.
Within Flow Objects, the activity element of the BPD can be made up of atomic activities (tasks) and of compound activities (collapsed sub-processes) and within each category different classes can be observed, as is shown in the previous table where a metric for each one of the four types of tasks and for the five types of sub-process is defined.
In the same category of "Flow Objects", the "Gateways" are the elements used to control the divergence and convergence of Sequence Flow. In the BPD, there are five types of Gateways, and we have defined metrics for each type (Table 5).
With these metrics, it is possible to know the number of Gateways that generate forks or joins of sequence flow at a specific point in the process. Other important elements to considerer within of the BPD core elements are shown in the Table 6 with its respective base measures.
Based on the base measures defined, the proposal of metrics for business process models includes some significant derived measures, obtained by means of measurement function, which establishes the existing proportions among the different elements of the model. The derived measures for business processes models with BPMN are shown in the Table 7.
With the proposed base and derived measures, it is possible to evaluate the structural complexity of business process models expressed in BPMN. When analyzing the model structurally, the quality of the model can also be assessed. In particular, this is done with reference to the three quality criteria for conceptual models given by Lindland: semantic quality, syntactic quality and pragmatic quality (Lindland et al. 1994).

5. EXTENSION OF FMESP

In the previous sections, we have described two proposals of metrics to evaluate software process models and business process models respectively. These metrics have been defined on two different metamodels, namely SPEM for software processes and BPMN for business process models. It is important to highlight that SPEM is a generic metamodel, and the measures proposed can be applied to other process modelling languages, even not specific to software as BPMN. On the other hand, being BPMN specifically focused on business processes it presents some aspects that are not contemplated for software processes and it means that new specific metrics are necessary. According to the issues mentioned, in order to measure BPMN business process models the metrics of the framework FMESP for SPEM have been successfully applied, but new metrics (not defined in FMESP) have been necessary due to the specific notation of

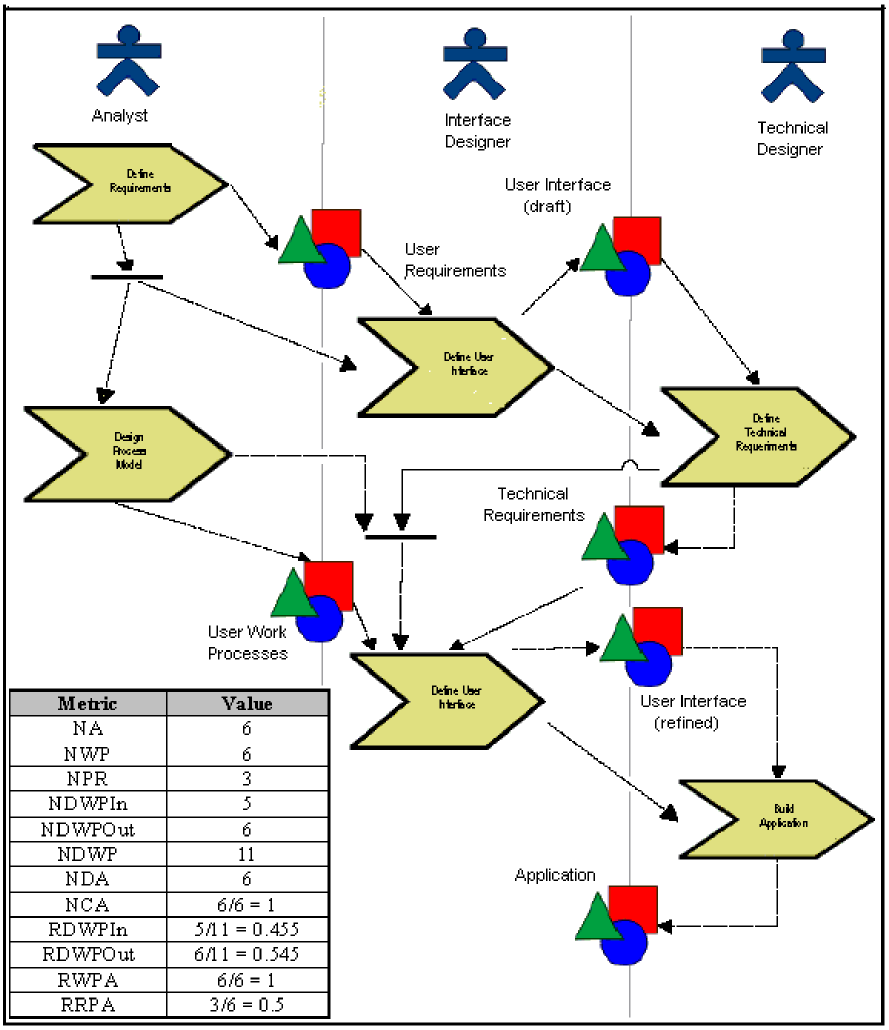
BPMN to model some particular aspects of business processes. The table 8 shows the modelling elements considered in SPEM and BPMN notations.
As we can observe in Table 8, there are some elements useful in BPMN for the modeling of business process that SPEM does not contemplate, such as the Events, Gateways, Message Flow and Pools. The base measures defined for these particular elements are shown in the table 9.
Since we have new base measures coming from the use of the metamodel of BPMN, a new group of derived measures is generated which has not been defined in FMESP. These derived measures that arise and which starts from the base measure shown in the previous table are set out in the Table 10. Note that although the activities are contemplated in both proposals, here they are included as an extension of FMESP because in BPMN, as we have already seen, atomic and compound activities can be observed. These can, in turn, have different characteristics or properties. With all the metrics defined, the base ones as well as the derived ones, we believe that one could have information about the structural complexity of the model of business processes, allowing us to evaluate aspects like their understandability, coherence, completeness, modifiability and consistency in order to assure the quality of the model at conceptual level (Lindland et al. 1994). In the following section, an example of a business process model using MPMN is presented, in which the metrics, as defined in FMESP for software process models, are applied as well as the metrics that we have defined for business process models.

6. APPLICATION EXAMPLE OF MSP AND MBP

To illustrate the calculation of the metrics defined for business process models one example is provided which has been taken from (BPMN 2004). The example (Figure 2) represents a concurrent engineering chip design process and our objective is to apply the metrics defined in this work in order to know its structural characteristics.
The values of the metrics defined in FMESP and the set of metrics defined according to BPMN applied in the above model are shown in the following tables. For reasons of space, in the case of the metrics for business processes, only the derived measures will be shown.
As can be appreciate by looking at the previous tables, practically no difference exists between the defined values of the metrics for the two types of processes (software and business). The difference that one can observe is in those metrics based on elements that are not contemplated by SPEM, but which can at the same time be useful in analyzing the business processes models structurally.
In this way, it is proven that although currently in the pertinent literature there are not proposals of metrics for the evaluation of business process models at conceptual level, it is possible to carry out their evaluation by applying defined metrics for software process models and by defining new specific ones for business process models.

Table 1. Model Scope Metrics

Metric	Definition
NA	Number Activities of the software process model
NWP	Number of Work Products of the software process model
NPR	Number of Roles which participate in the process
NDWPIn	Number of input dependences of the Work Products with the Activities in the process
NDWPOut	Number of output dependences of the Work Products with the Activities in the process
NDWP	Number of dependences between Work Products and Activities *NDWP(PM) = NDWPIn(MP) + NDWPOut(MP9*
NDA	Number of precedence dependences between Activities
NCA	Activity Coupling in the process model $NCA(PM) = \frac{NA(PM)}{NDA(PM)}$
RDWPIn	Ratio between input dependences of Work Products with Activities and total number of dependences of Work Products with Activities. $RDWPIn(PM) = \frac{NDWPIn(PM)}{NDWP(PM)}$
RDWPOut	Ratio between output dependences of Work Products with Activities and total number of dependences of Work Products with Activities. $RDWPOut(PM) = \frac{NDWPOut(PM)}{NDWP(PM)}$
RWPA	Ratio of Work Products and Activities. Average of the work products and the activities of the process model $RWPA(PM) = \frac{NWP(PM)}{NA(PM)}$
RRPA	Ratio of Process Roles and Activities. $RRPA(PM) = \frac{NPR(PM)}{NA(PM)}$



Metric	Value
NA	6
NWP	6
NPR	3
NDWPIn	5
NDWPOut	6
NDWP	11
NDA	6
NCA	6/6 = 1
RDWPIn	5/11 = 0.455
RDWPOut	6/11 = 0.545
RWPA	6/6 = 1
RRPA	3/6 = 0.5

Figure 1. Software Process Model with SPEM and Metric Values

Table 2. Core Elements Set of BPD

BPD Core Element Set			
Flow Objects	**Connecting Objects**	**Swimlanes**	**Artefacts**
Events	Sequence Flow	Pool	Data Objects
Activities	Message Flow	Lanes	Groups
Gateways	Association		Text Annotation

Table 3. Base Measures for the element Event in BPD Object Flow

Core Element	Notation	Metric Name	Base Measure	Definition
Start Event	Start	NSNE	Number of Start None Events	Indicates the total number of start none events in the model
	Timer	NSTE	Number of Start Timer Events	Indicates the total number of start timer events in the model
	Message	NSMsE	Number of Start Message Events	Indicates the total number of start Message events in the model
	Rule	NSRE	Number of Start Rule Events	Indicates the total number of normal start events in the model
	Link	NSLE	Number of Start Link Events	Indicates the total number of start link events in the model
	Multiple	NSMuE	Number of Start Multiple Events	Indicates the total number of start multiple events in the model
Intermediate Event	Intermediate	NINE	Number of Intermediate None Events	Indicates the total number of intermediate none events in the model
	Timer	NITE	Number of Intermediate Timer Events	Indicates the total number of intermediate timer events in the model
	Message	NIMsE	Number of Intermediate Message Events	Indicates the total number of intermediate message events in the model
	Error	NIEE	Number of Intermediate Error Events	Indicates the total number of intermediate error events in the model
	Cancel	NICaE	Number of Intermediate Cancel Events	Indicates the total number of intermediate cancel events in the model
	Compensation	NICoE	Number of Intermediate Compensation Events	Indicates the total number of intermediate compensation events in the model
	Rule	NIRE	Number of Intermediate Rule Events	Indicates the total number of intermediate rule events in the model
	Link	NILE	Number of Intermediate Link Events	Indicates the total number of intermediate link events in the model
	Multiple	NIMuE	Number of Intermediate Multiple Events	Indicates the total number of intermediate multiple events in the model
End Event	End	NENE	Number of End None Events	Indicates the total number of end none events in the model
	Message	NEMsE	Number of End Message Events	Indicates the total number of end message events in the model
	Error	NEEE	Number of End Error Events	Indicates the total number of end error events in the model
	Cancel	NECaE	Number of End Cancel Events	Indicates the total number of end cancel events in the model
	Compensation	NECoE	Number of End Compensation Events	Indicates the total number of end compensation events in the model
	Link	NELE	Number of End Link Events	Indicates the total number of end link events in the model
	Multiple	NEMuE	Number of End Multiple Events	Indicates the total number of end multiple events in the model
	Terminate	NETE	Number of End Terminate Events	Indicates the total number of end terminate events in the model

Table 4. Base Measures for the element Activity of the BPD Flow Objects

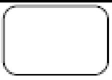
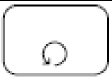
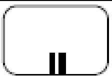
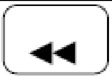
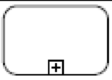
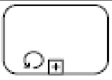
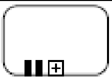
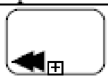
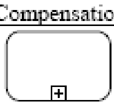
Core Element	Notation	Metric Name	Base Measure	Definition
Task	 Task	NT	Number of Task	Indicates the total number of tasks in the model
	 Looping	NTL	Number of Task Looping	Indicates the total number of task looping in the model
	 Multiple Instantes	NTMI	Number of Task Multiple Instances	Indicates the total number of task multiple instances in the model
	 Compensation	NTC	Number of Task Compensation	Indicates the total number of task compensation in the model
Collapsed Sub-Process	 Collapsed Sub-Process	NCS	Number of Collapsed Sub-Process	Indicates the total number of Collapsed Sub-Processes in the model
	 Loop	NCSL	Number of Collapsed Sub-Process Looping	Indicates the total number of Collapsed Sub-Process Looping in the model
	 Multiple Instance	NCSMI	Number of Collapsed Sub-Process Multiple Instance	Indicates the total number of Collapsed Sub-Process Multiple Instance in the model
	 Compensation	NCSC	Number of Collapsed Sub-Process Compensation	Indicates the total number of Collapsed Sub-Process Compensation in the model
	 Ad-Hoc	NCSA	Number of Collapsed Sub-Process Ad-Hoc	Indicates the total number of Collapsed Sub-Process Ad-Hoc in the model

Table 5. Base Measures for the Gateway Control Types in the BPD Flow Objects.

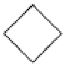
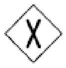
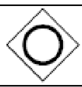
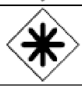
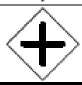
Core Element	Notation	Metric Name	Base Measure	Definition
Exclusive Decisión Data-Based XOR Decision	 ó 	**NEDDB**	Number of Exclusive Decision/Merge Data-Based	Indicates the number of points of exclusive decision and merging based on data of the model
Exclusive Decisión Data-Event XOR Decision		**NEDEB**	Number of Exclusive Decision/Merge Event-Based	Indicates the number of points of exclusive decision and merging based on events of the model
Inclusive (OR)		**NID**	Number of Inclusive Decision/Merge	Indicates the number of points of inclusive decision and merging of the model
Complex		**NCD**	Number of Complex Decision/Merge	Indicates the number of points of complex decision merging of the model
Paralell (AND)		**NPF**	Number of Parallel Fork/Join	Indicates the number of points of parallel forking and joining of the process

Table 6. Base Measure for the Connecting Objects, Swimlanes and Artefacts

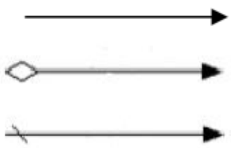
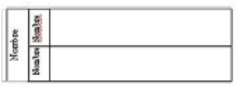
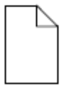
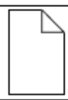
Core Element	Notation	Metric Name	Base Measure	Definition
Sequence Flow		**NSF**	Number of Sequence Flows in the Process	Indicates the number of Sequence Flow between events and activities in the process model
Message Flow		**NMF**	Number of Message Flows between Participants in the Process	Indicates the number of Message Flow between participants in the process model
Pool		**NP**	Number of Pools in the Process	Indicates the number of Participants in the process model
Lanes		**NL**	Number of Lanes in the Process	Indicates the number of internal roles, systems and internal department within the Pools in the Process Model.
Data Objects (Input)		**NDOIn**	Number of Data Object-In of the Process	Indicates the number of Data Objects used as inputs to the activities in the Process Model.
Data Objects (Output)		**NDOOut**	Number of Data Object-Out of the Process	Indicates the number of Data Objects used as outputs of the activities in the Process Model.

Table 7. Derived Measure of elements common to the category of Flow Objects

Name	Formula	Metric	Definition
NTSE	TNSE = NSNE+NSTE+NSMsE+NSRE+ NSLE+NSMuE	Total Number of Start Events of the Model	Indicates the total number of Start Events in the Process Model
NTIE	TNIE = NINE+NITE+NIMsE+NIEE+NICaE+ NICoE+NIRE+NILE+NIMuE	Total Number of Intermediate Events of the Model	Indicates the total number of Intermediate Events in the Process Model
TNEE	TNEE = NENE+NEMsE+NEEE+NECaE+ NECoE+NELE+NEMuE+NETE	Total Number of End Events of the Model	Indicates the total number of End Events in the Process Model
TNT	TNT = NT+NTL+NTMI+NTC	Total Number of Task of the Model	Indicates the total number of Tasks in the Process Model
TNCS	TNCS = NCS+NCSL+NCSMI+NCSC+NCSA	Total Number of Collapsed Sub-Process of the Model	Indicates the total number of Collapsed Sub-Process in the Process Model
TNE	TNE = NTSE + NTIE + TNEE	Total Number of Events of the Model	Indicates the total umber of the events (start, intermediate and End) in the Process Model
TNG	TNG = NEDDB+NEDEB+NID+NCD+NPF	Total Number of Gateways of the Model	Indicates the total number of Gateways in the Process Model.
TNDO	TNDO = NDOIn + NDOOut	Total Number of Data Objects in the Process Model	Indicates the total number of Data Objects (Inputs and Outputs) in the Process Model.

Name	Formula	Metric	Definition
CLA	$CLA = \frac{TNT}{NSF}$	Connectivity level between Activities	Indicates the proportion between the total number of Tasks and the total of precedence dependences (Flows of Sequence) of the Process.
CLP	$CLP = \frac{NMF}{NP}$	Connectivity Level Between Pools	Indicates the proportion of the total of Participants in the process and the Message Flows among them.
PDOPIn	$PDOPIn = \frac{NDOIn}{TNDO}$	Proportion of Data Object like Incoming Product and the total of Data Objects	Indicates the proportion of the Data Objects that represent an input for an activity and the total of Data Objects in the Process Model.
PDOPOut	$PDOPOut = \frac{NDOOut}{TNDO}$	Proportion of Data Object like Outgoing Product and the total of Data Objects	Indicates the proportion of the Data Objects that represent an Output of an activity and the total of Data Objects in the Processes Model.
PDOTOut	$PDOTOut = \frac{NDOOut}{TNT}$	Proportion of Data Object like Outgoing Product of Activities of the Model	Indicates the proportion of the Data Objects that represent an output in relation to the Tasks of the Process Model.
PLT	$PLT = \frac{NL}{TNT}$	Proportion of Pools and/or Lanes of the Process and Activities in the Model	Indicates the proportion of the number of roles or departments in relation to the tasks of the Process Model.

Table 8. Constructor of SPEM and BPMN for definition of metrics

Element	SPEM (FMESP)	BPMN
Events		✔
Activities	✔	✔
Gateways		✔
Work Products (Data Objects)	✔	✔
Roles (Lanes)	✔	✔
Dependencias (Sequence Flow)	✔	✔
Message Flow		✔
Pools		✔

Table 9. New Based Measure based on BPMN

Element	Category	Base Measure
Events	Start	NSNE, NSTE, NSMsE, NSRE, NSLE, NSMuE
	Intermediate	NINE, NITE, NIMsE, NIEE, NICaE, NICoE, NIRE, NILE, NIMuE
	End	NENE, NEMsE, NEEE, NECaE, NECoE, NELE, NEMuE, NETE
Activities	Tasks	NT, NTL, NTMI, NTC
	Collapsed sub-process	NCS, CSL, NCSMI, NCSC, NCSA
Gateways		NEDDB, NEDEB, NID, NCD, NPF
Message Flow		NMF
Pools		NP

Table 10. New Derived Measure based on BPMN

Name	Metric
NTSE	Total Number of Start Events of the Model
NTIE	Total Number of Intermediate Events of the Model
TNEE	Total Number of End Events of the Model
TNT	Total Number of Task of the Model
TNCS	Total Number of Collapsed Sub-Process of the Model
TNE	Total Number of Events of the Model
TNG	Total Number of Gateways of the Model
CLP	Connectivity Level Between Pools
PDOPIn	Proportion of Data Object like Incoming Product and the total of Data Objects
PDOPOut	Proportion of Data Object like Outgoing Product and the total of Data Objects
PDOTOut	Proportion of Data Object like Outgoing Product of Activities of the Model
PLT	Proportion of Pools and/or Lanes of the Process and Activities in the Model

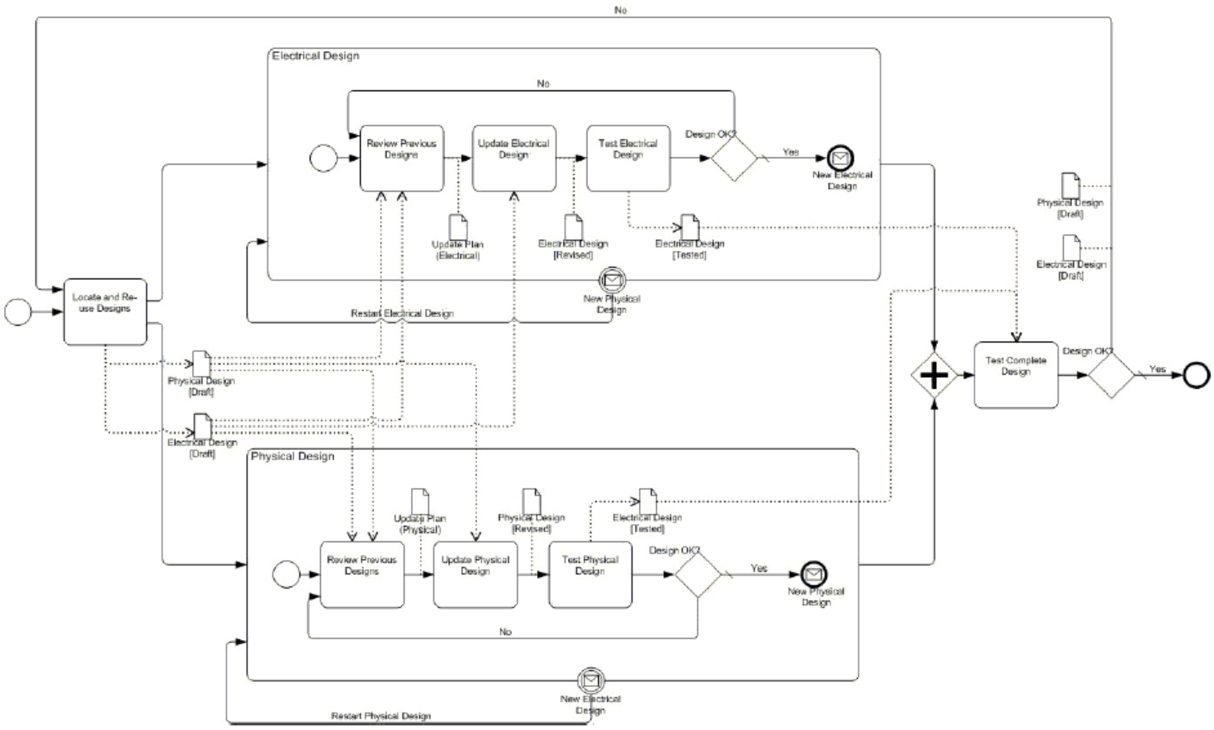


Figure 2. Model concurrent of Engineering with BPMN

Table 11. Value of Metrics defined in FMESP and Derived Measure wit BPMN

Metrics of FMESP	
Metric	**Value**
NA	8
NWP	8
NPR	2
NDWPIn	14
NDWPOut	8
NDWP	22
NDA	11
NCA	8/11 = 0.727
RDWPIn	14/22 = 0.636
RDWPOut	8/22 = 0.363
RWPA	8/8 = 1
RRPA	2/8 = 0.25

Derived Measure with BPMN	
Metric	**Value**
NTSE	3
NTIE	2
TNEE	3
TNT	8
TNCS	0
TNE	8
TNG	4
TNDO	22
CLA	8/11 = 0.727
CLP	0
PDOPIn	14/22 = 0.636
PDOPOut	8/22 = 0.363
PDOTOut	8/8 = 1
PLT	2/8 = 0.25

7. CONCLUSIONS AND FURTHER WORK.

In this paper, we have proposed and illustrated how the proposal of FMESP can be applied in order to evaluate business process models at conceptual level. The FMESP proposal has been shown, in which a group of metrics is defined for the evaluation and measurement of the structural properties of software process models. These metrics were defined following the SPEM terminology and they can be applied as useful maintainability indicators. In FMESP, there are two categories of metrics: model and core element scope and each category contains base and derived measures. These metrics make possible to determine the structural complexity of software process models.

Taking into consideration that in the field of process engineering there are not metrics applicable to business process models at conceptual level, we make use of the philosophy of FMESP in order to evaluate the structural complexity of business process models. We have taken as our starting point a definition of base measures and derived measures following the BPMN terminology, which is the most recent standard notation defined by BPMI for the modeling of business process.

In this work it has been proved that it is possible to apply metrics for software process models to business process models, since they present certain similarities regarding the core elements that both are made up of. However, it has been necessary to extend the metrics defined in FMESP to embrace all the aspects considered within a business process model.

By integrating both proposals we provide a more refined framework for evaluating business process models. This gives support to Business Process Management, which has as one of its stages the definition and modelling of the process being assessed. It will allow a more appropriate management of the business processes and can provide organizations with important profits. Model metrics can be very useful to select the models with the most easiness of maintenance among various alternatives in companies with change their models to improve their business processes. Also, it can help to facilitate the business processes evolution in these companies by assessing the process improvement at conceptual level.

The business process model metrics provide companies with objective information about the maintainability of these models. More maintainable models can benefit the management of the business processes mainly in two ways: i) guaranteeing the understanding and the diffusion of the processes, as they evolve, without affecting their successful execution; ii) reducing the effort necessary to change the models with the consequent reduction of the maintenance.

Currently we are developing a family of experiments with the purpose of to evaluate quality aspects of the conceptual business process models. These experiments will be developed with a population integrated by experts in business analysis and in software engineering in order to be able a comparison between results of both kinds of stakeholders and to determine the influence of these different points of view.

Participants will receive a kit consisting of a set of business processes models represented with BPMN. Models will have different characteristics and dimensions. A questionnaire will also be provided for each one of the models including questions related with its understandability and complexity. In order to assess how influence the BPMN notation in the modifiability of models other additional section of the questionnaire will ask about several modifications -specially studied- to the original model.

8 REFERENCES

Acuña, S. T. and Ferré, X. (2001). Software Process Modelling. Proceedings of the 5th. World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando Florida, USA. Beck, K., Joseph, J. and Goldszmidt, G. (2005). Learn Business Process Modeling Basics for the Analyst.

IBM. www-128ibm.com/developersworks/library/ws-bpm4analyst

BPMI (2004). Business Process Modeling Notation. Specification Version 1.0. Business Process Management Initiative. May 3, 2004. BPMN (2004). www.bpmn.org

Canfora, G., García, F., Piattini, M., Ruiz, F. and Visaggio, C. A. (2005). "A Family of Experiments to Validate Metrics for Software Process Models." Journal of Systems and Software 77(2) pp. 113-129.

Curtis, B., Kellner, M. I. and Over, J. (1992). "Process Modeling." Communications of the ACM Vol. 35(No. 9): pp. 75-90.

Dufresne, T. and Martin, J. (2003). Process Modeling for E-Business. INFS 770 - Methods for Informations Systems Engineering: Knowledge Management and E-Business. George Mason University. Spring 2003.

Erickson, H.-E. and Penker, M. (2000). Business Modeling with UML- Business Patterns at Work. Robert Ipsen. USA ISBN 0-471-29551-5

Finkelstein, A., Kramer, J. and Hales, M. (1992). Process Modelling: a Critical Analysis. Integrated Software Reuse: Management and Techniques. P. Walton and N. Maiden, Chapman and Hall and UNICOM: pp. 137-148.

FIPS (1993). Integration Definition for Function Modeling (IDEF0). Standard. National Institute of Standards and Technology. December 1993.

Florac, W. A., Park, R. E. and Carleton, A. D. (1997). Practical Software Measurement: Measuring for Process Management and Improvement. Guidebook. CMU/SEI-97-HB-003. Carnegie Mellon University. April 1997.

Fuggetta, A. (2000). Software Process: A Roadmap. Proceedings of the 22th. International Conference on Software Engineering (ICSE-2000), Limerick, Ireland.

García, F., Ruiz, F., Piattini, M., Canfora, G. and Visaggio, C. A. (2005). "Framework for the Modeling and Evaluation of Software Processes." Journal of Systems Architecture (accepted to appear).

Hommes, B.-J. and van Reijswoud, V. (2000). Assessing the Quality of Business Process Modelling Techniques. Proceedings of the 33rd Hawaii International Conference on Systems Sciences (HICSS 2000), Maui, Hawaii, USA, IEEE.

Lindland, O. I., Sindre, G. and Solvnerg, A. (1994). "Understanding Quality in Conceptual Modeling." Software IEEE Vol. II(Issue 2): pp. 42-49.

Mayer, R. J., Menzel, C. P., Painter, M. K., de White, P. S., Blinn, T. and Perakath, B. (1995). Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report. Interim Technical Report. September 1995.

Mc. Daniel, T. (2001). "Ten Pillar of Business Process Management." eAIJournal: pp 30-34.

OMG (2002). Software Process Engineering Metamodel Specification. Adopted specification, version 1.0. Object Management Group, Inc. November 2002.

OMG (2003). Unified Modeling Language (UML) Specification: Infrastructure, version 2.0. Object Management Group. December 2003.

Owen, M. and Raj, J. (2003). BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. White Paper. Popkin Software. September, 2003.

Sharp, A. and McDermott, P. (2000). Workflow Modeling: Tools for Process Improvement and Application Development. Artech House (Pub). London ISBN 1-58053-021-4

Tjaden, G. S. (1999). Business Process Structural Analysis. Georgia Tech Center for Enterprise Systems. October 1999.

Wedemeijer, d. L. and de Bruin, d. i. E. (2004). Conceptual Process Models: Using Process Architecture in Practice. Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA´04), Zaragoza, España, IEEE Computer Society.

White, S. A. (2004). Introduction to BPMN. IBM Corporation. bpmn.org (pub). May 2004. www.bpmn.org.