Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2013 Completed Research | ECIS 2013 Proceedings

7-1-2013

Business Process Points - A Proposal To Measure BPM Projects

Maruscia Baklizky
University of Sao Paulo, Sao Paulo, SP, Brazil, maruscia.baklizky@usp.br

Marcelo Fantinato
University of Sao Paulo, Sao Paulo, SP, Brazil, m.fantinato@usp.br

Lucinéia H. Thom
Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil, lucineia@inf.ufrgs.br

Violeta Sun
University of Sao Paulo, Sao Paulo, SP, Brazil, Violeta@usp.br

Edmir P V Prado
University of Sao Paulo, Sao Paulo, SP, Brazil, eprado@usp.br

See next page for additional authors

Authors
Maruscia Baklizky, Marcelo Fantinato, Lucinéia H. Thom, Violeta Sun, Edmir P V Prado, and Patrick Hung

BUSINESS PROCESS POINTS - A PROPOSAL TO MEASURE BPM PROJECTS

Baklizky, Maruscia, School of Arts, Sciences and Humanities, University of São Paulo, Rua Arlindo Béttio 1000, 03828-000 São Paulo - SP, Brazil, maruscia.baklizky@usp.br

Fantinato, Marcelo, School of Arts, Sciences and Humanities, University of São Paulo, Rua Arlindo Béttio 1000, 03828-000 São Paulo - SP, Brazil, m.fantinato@usp.br

Thom, Lucinéia Heloisa, Institute of Informatics, Federal University of Rio Grande do Sul, Caixa Postal 15064, 91501-970 Porto Alegre - RS, Brazil, lucineia@inf.ufrgs.br

Sun, Violeta, School of Arts, Sciences and Humanities, University of São Paulo, Rua Arlindo Béttio 1000, 03828-000 São Paulo - SP, Brazil, violeta@usp.br

Prado, Edmir Parada Vasques, School of Arts, Sciences and Humanities, University of São Paulo, Rua Arlindo Béttio 1000, 03828-000 São Paulo - SP, Brazil, eprado@usp.br

Hung, Patrick, University of Ontario Institute of Technology, 2000 Simcoe Street North, L1H 7K4 Oshawa - ON, Canada, patrick.hung@uoit.ca

Abstract

Organizations need more standardization and efficiency in business process execution, which is leading them to show an increasing interest in the business process management (BPM) approach. Stakeholders and workers from different departments are collaborating to obtain better products/services as a result of the BPM projects, but do not always achieve satisfactory outcomes. BPM projects must be well managed and measured for successful implementation. Software Engineering (SE) projects often use a Functional Size Measurement (FSM) technique to estimate the size of the project. This paper proposes an FSM technique for BPM, based on the Function Points metric from SE. We chose the Function Point Analysis, (an FSM technique), as the basis for the proposed Business Process Point Analysis (BPPA) technique. BPPA is a Process Size Measurement technique, developed for business processes modeled through the Business Process Management Model and Notation. BPPA will enable project managers to measure Business Process Points of BPM projects by allowing them to estimate important variables for better managing projects, such as required resources, human effort, cost and time. This paper provides an overview of the proposed BPPA technique and the results of an empirical analysis based on observations from a BPM specialist and an FPA specialist.

Keywords: *Business Process Management, Business Process Model and Notation, Functional Size Measurement, Function Point Analysis.*

1 Introduction

Business Process Management (BPM) involves concepts, methods and techniques to support the representation and enactment of Business Processes (BP) (Weske, 2007). According to Weske (2007), BPM is an approach that is adopted in organizational environments and leads clients and consumers, analysts and engineers, and other stakeholders to understand BPs better and use them in cooperation.

The application of the BPM approach inside organizations can be understood as a BPM project responsible for the analysis, development, evaluation and enhancement of BPs. According to the Process Management Initiative (PMI) (2008), regardless of the project type being developed, a project manager is expected to guide and control the artifacts being produced in each phase, to estimate relevant aspects in the execution of a project, e.g. schedule performance, and to carry out management activities. In this context, it is important to be aware of and control four variables to ensure a project is well managed: required resources, human effort, cost and time (PMI, 2008).

Although the use of BPM has been increasing lately, there is a lack of tools and techniques that can be used in measuring BPM projects regarding to functional complexity (Sánchez et al., 2010). For example, there is no technique that provides important information for project managers to enable them to understand and control the required resources, effort, cost and time for BPM projects. In contrast, in the Software Engineering (SE) context, there is a wide range of software metrics for estimating different types of project variables (PMI, 2008).

One of the most commonly used software metrics is Function Points (FP). FP is used as a size unit for measuring a software project (IFPUG, 2010). The FP metric, which is associated with a Function Size Measurement (FSM) technique, can be applied during a software project to obtain the software functional size (also called functional complexity). This information, combined with some historical data, provides the required resources, effort, cost and time needed to develop the measured software project (Sommerville, 2004).

In the SE context, the most widely used FSM-like technique is the Function Point Analysis (FPA). It was first proposed in 1979 and has been continuously enhanced by the International Function Point Users Group (IFPUG) (IFPUG, 2010). FPA is employed in software project documentation to compute its functional size. As well as in SE, the BPM project managers also need a metric to produce valuable information about the project sizes – in a similar way to which functional sizes are calculated for software projects – in order to derive other important variables for managing BPM projects.

As in SE, where the FP metric and FPA technique are applied to help manage software projects, an extension of the FP metric and FPA technique from SE projects would be helpful in the BPM domain. Thus, a mapping between the SE and BPM domains provided the means of guiding this study and determining if a parallel could be established. The mapping was carried out by making a comparison between the SE and BPM approaches, in terms of the activities and result artifacts of each phase. Both domains were found to be similar with regard to phases, artifacts and procedures. Their differences are mainly related to the content of their artifacts. The life-cycles of SE (Sommerville, 2004) and BPM (Weske, 2007) comprise the phases shown in Table 1, which also includes a directed mapping between them. In SE, software projects are conducted that mainly concern developing new software products (Phases 1.1 to 2.2) but also address factors such as their operation and maintenance (Phase 3). With regard to BPM projects, the life-cycle is shorter, and usually faster and more dynamic during the development phase (Phases 1 and 2) but, at the same time, there is a greater importance attached to post-development (Phases 3.1 and 3.2); typically, according to Fantinato et al. (2012), "a given business Process Model (PM) evolves far more often than a given software in the same period".

Although FPs and FPA are very helpful in SE, they cannot be directly applied to BPM projects as they are driven by information only available in the software project documentation. This paper motivates, proposes and evaluates the adoption of a "Business Process Points" (BPP) analysis to calculate the process size of BPM projects. It is an adaptation to the functional size from the SE domain so that it can fit in the BPM domain, and be used in the BPP metric. This maintains the main goal of the functional size and allows key estimates to be made of information for the management of the project.

BPM life-cycle phases	SE life-cycle phases
1 – Design and analysis	1.1 – Requirements specification
	1.2 – System and software design
2 – Configuration	2.1 – Implementation and unit test
	2.2 – Integration and systems test
3.1 – Enactment	3 – Operation and maintenance
3.2 – Evaluation	

Table 1. Mapping between SE's and BPM's life-cycles.

The proposed technique, BPP Analysis (BPPA), was built upon the FPA framework. In the same way as documentation of the software project is input for FPA, documentation of BPM project is input for BPPA. The documentation of the BPM project is basically composed of PMs. BPPA only focuses on the PMs specified in the Business Process Model and Notation (BPMN), version 2.0 – a standard modeling notation for PMs (BPMI, 2010); which are referred to in this paper as PM-BPMNs. Finally, the goal of BPPA is to provide a technique that can help managers of BPM projects in the main activities related to the control and monitoring of these projects.

The remainder of this paper is organized as follows: Section 2 discusses related work; Section 3 summarizes the BPPA technique being proposed; Section 4 shows the preliminary results of two empirical evaluations of the proposed BPPA technique; Section 5 summarizes the conclusions and makes recommendations for future works.

2 Related Work

Most initiatives concerning business measurement have been adapted from SE field (Sánchez et al., 2010). Specifically on this issue, Gruhn and Laue (2006) point out some software complexity metrics that can be extended to analyze the complexity of a PM; including Cognitive Complexity and Fan-in/Fan-out metrics. Among the proposed extensions of this study, the unique size-related metric examined is Lines of Code (LoC), a metric related to the structural model size – called Number of Activities (NoA) in the BPM context. None of the proposed extensions is related to the FSM that is proposed by this study.

With regard to the SE context, some FPA extensions were found in the literature, two of which should be highlighted here: FPs for count enterprise data warehouses (IFPUG, 2007) and FPs for count middleware software applications (IFPUG, 2009). Both were published by IFPUG as White Papers to broaden the application of FPA in systems domains and can be regarded as formal extensions of the FPA technique, since they are based on the original FP metric, which cannot be applied in this study because SE and BPM have some different properties.

Cosmic (COSMIC, 2009) and Use Case Points (UCP) (Clemmons, 2006) are two FPA-based metrics. The relationship between these two extended metrics and a BPM context has already been addressed in two other studies and is explored further here.

Kaya and Demirörs (2011) describe an approach to apply the Cosmic FSM method to software projects which use BPMN notation to model software functions. Although their approach uses BPMN notation, in fact, it is restricted to a modeling artifact within the SE context and not in the BPM domain proposed by this study. The Cosmic FSM method was first designed to measure real time applications from SE, which is not the objective of this paper.

Dhammaraska and Intakosum (2009) provide a guide on how to measure PMs using UCPs. PMs are transformed into use case descriptions, the results of which are measured on the basis of the original

UCP technique. In contrast with our approach, they attempt to adapt PMs to the SE context by rewriting them as use case descriptions, but in this process some properties of the PM can get lost. Although the application of UCPs might be easier, the use case descriptions may vary a lot. This can lead to variable productivity, which affects the derivation of the four most important factors mentioned before for project management (Aguiar, 2009). Moreover, UCPs are only applied in organizations with requirements described by use cases, which affects the comparisons made between organizations.

3 Business Process Point Analysis (BPPA)

This section aims at giving a brief description of the proposed BPPA technique, which is based on IFPUG's FPA, as well as highlighting the most important decisions required to create this FSM technique for BPM projects. A technical report is used to make a methodical description of BPPA (Baklizky and Fantinato, 2012). Figure 1 shows the BPPA diagram and the levels of changes from the original FPA diagram. The stages represented in the diagram (as well as their "steps"), are explained in the next subsections, and compared with the FPA stages and steps.

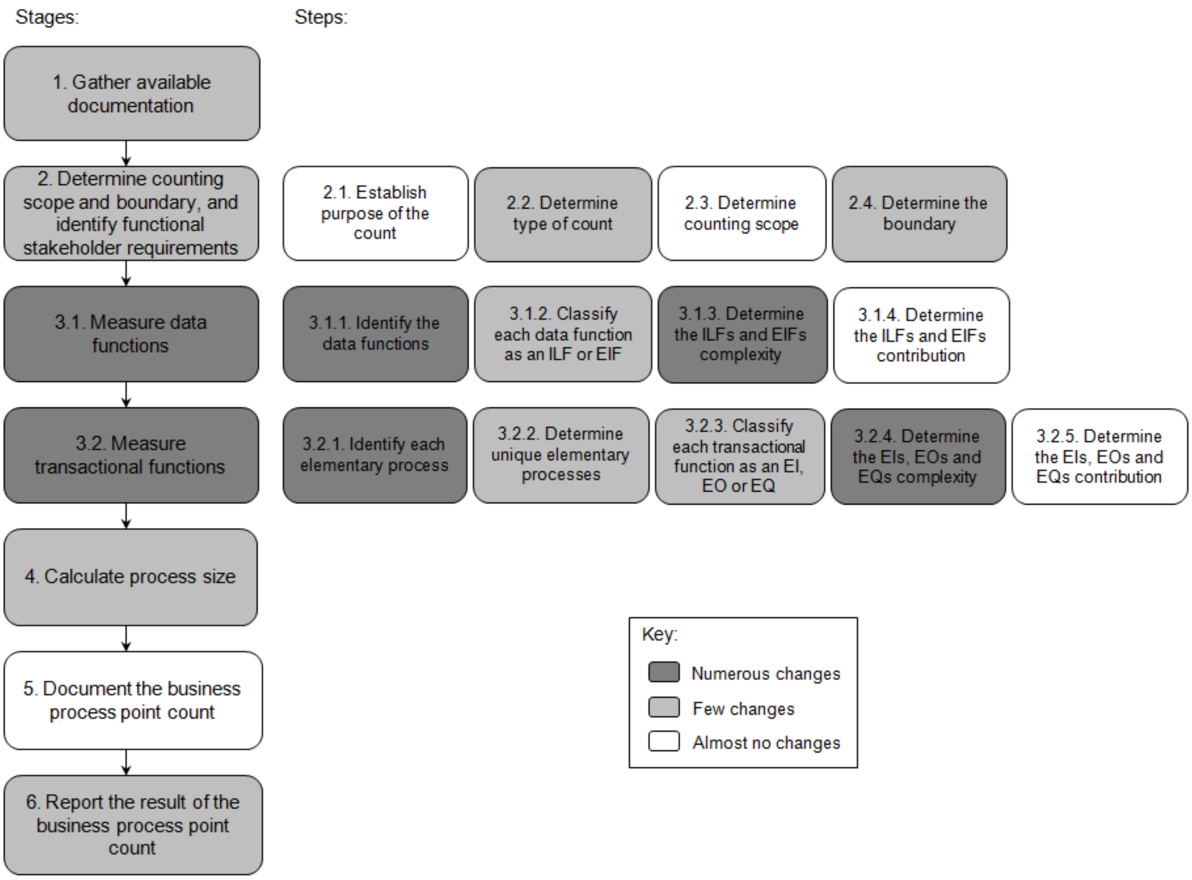

Figure 1. Stages and steps of BPPA technique.

Stage 1 – Gather available documentation

As a reference for this first stage of BPPA, in FPA all the documentation available for the FSM must be gathered. This documentation has to describe system functional requirements, and take account of what is significant for the user. One example is the user requirements specification. Extra artifacts can be used as support documents if relevant, such as entity diagrams and use cases (IFPUG, 2010). In BPPA, the documentation for the FSM must be PM-BPMNs available, since it is the main type of artifact used to represent Functional Stakeholder Requirements (FSR) in a BPM project. Additional artifacts can be also used, such as activity textual descriptions used to complement PM-BPMNs.

Stage 2 – Determine count scope and boundary, and identify FSRs

For this second stage of BPPA, four steps must be followed as shown in the following.

Step 2.1 – Establish the purpose of the count

An FP count is conducted in response to business requirements, and the purpose is determined by the business problem (IFPUG, 2010). FPA and BPPA take the business scenario as a rule. Whereas in FPA the purpose of the count influences the boundary line between two software products, the equivalent influence on BPPA is between two PMs. One example of purpose for a BPP count could be: to give the process size of a BPM project as an input of the effort estimation to configure a BPMS (BPM System) that will give automated support for the enactment of BPs.

Step 2.2 – Determine type of count

The type of the count is based on the purpose defined in the previous step. FPA specifically includes three types of count: development, application, and enhancement projects – which vary according to the phase in which the software is measured in the software life-cycle (IFPUG, 2010). This first version of BPPA technique is designed to address one type of count: development project – where the "Configuration" phase (refer to Table 1) is the only target for the purpose of the count. Future versions of the BPPA technique should address application and enhancement project types – since the "Enactment" and "Evaluation" (Table 1) phases may also be targeted for the purpose of the count.

Step 2.3 – Determine count scope

In FPA, the count scope defines the groups of FURs in the FP count (IFPUG, 2010). In BPPA, the count scope is the group of PM-BPMNs in the BPP count, since they describe the interactions between two or more business entities and represent the activities for a BP and, hence, they also represent FURs in BPM projects. In a similar way to the FP count scope, the BPP count scope should: *(i)* define the BPM project that must be measured; *(ii)* be determined by the purpose of the BPP count; and (*iii)* identify which processes should be included in the process size as well as providing relevant answers to the purpose of the count. In this version of the BPPA technique, only the collaboration diagram type is addressed, since it is the most widely used in organizations that represent PMs.

Step 2.4 – Determine the boundary

For FPA, the boundary is the conceptual interface between the system under study and its users (IFPUG, 2010). In BPPA, the boundary must be the conceptual interface between the main participants of the BP modeled in BPMN and the other participants. This BPP count boundary must: *(i)* indicate the limit between the main and other participants in the same PM-BPMN; *(ii)* define the actions and data elements that are the responsibility of the main participant; and (*iii)* act as a "membrane" through which data are processed by transactions. The boundary can overlap the area of a pool or specific swimlanes (BPMI, 2011) in the PM-BPMN being measured. In addition, the PM-BPMNs are modeled on the basis of a business view (BPMI, 2011). It is used to represent the main participant of a BP and its collaborations with other participants. The modeling view is one of the determining factors in the process of identifying the boundary, as it influences the interpretation of the modeled BP and highlights the needs, responsibilities and data that the main participant requires. The procedure to determine the boundary involves: *(1)* identifying all the participants present in the PM-BPMN; *(2)* checking the modeling view used for the PM-BPMN being measured; *(3)* defining the main participant of the PM-BPMN, on the basis of the previous task; and (*4)* defining the boundary, in accordance with the three guidelines defined at the beginning of this paragraph.

Stage 3 – Measure functions

In this stage of BPPA, two types of functions are measured: data functions and transactional functions.

Stage 3.1 – Measure data functions

Data Function (DF) is the first of the two central items for both FP and BPP. For BPPA, DFs must satisfy one or more data action requirements including data reading, writing, input, output, reception or transmission within the scope of a PM-BPMN. In BPPA, DFs can be Internal Logical Files (ILF) or External Interface Files (EIF), as in FPA. The steps for measuring DFs are described as follows.

Step 3.1.1 – Identify the data functions

In FPA, a DF is a set of logically related data, i.e. data stored persistently for further retrieval, update and reference. The amount of logically linked data is estimated so they can be grouped in DFs (IFPUG, 2011). For BPPA, this concept needs to undergo some changes. Basically, DFs are complex data groupings. Generally, data elements are represented in PM-BPMNs in a simpler granularity level than those directly classifiable as DFs. As a result, all the data elements found in the PM-BPMNs are initially only classified as Data Structures (DS). Each DS represents a data element processed by one or more activities or a data element received or transmitted by message flows. The existing DSs have to be identified and classified so that they can be correctly used in the next steps of this technique. DSs are classified on the basis of both representation and complexity. The classification of representation is based on the element type. DSs can represent three types of BPMN elements: data object (including collection, input and output subtypes), data store and event message. Regarding the classification of complexity, the decision depends on whether the DS is already a data group or not. For the classification of representation, data objects and event messages represent volatile data, i.e. data which are only used in a BP instance. Data stores represent persistent data, as well as data which are permanently stored regardless of the enactment of a specific BP instance. Moreover, data objects and data stores are "directly represented data elements" since they are originally created for this purpose, whereas the event messages are "indirect represented data elements" since they also represent data although their main purpose is to carry out a message exchange.

The procedure to identify the DFs is: *(1)* identify all DSs in the PM-BPMN scope; *(2)* classify each DS according to its representation and complexity types; *(3)* name each DS; *(4)* arrange functionally-related and non-complex DSs in different groups, by taking their representation types as a constraint, i.e. DSs that are directly represented cannot be merged with DSs that are indirectly represented, which means that the original representation types are kept in the resulting groups; *(5)* count the number of DSs grouped in each resulting group of DSs; *(6)* count the number of relationships (associations or flows) linking each DS, from the group of DSs, to the activities of the PM-BPMN; and *(7)* classify each resulting group of DSs as a DF.

Step 3.1.2 – Classify each data function as an ILF or EIF

In FPA, a DF is classified as an ILF or an EIF depending on its primary intent and its relative position to the FP count boundary (IFPUF, 2011). For BPPA, a DF is also classified as ILF or EIF depending on its primary intent and the boundary. However, in BPPA, account is also taken of the types of the DSs which compose the DF, in accordance with the following rules: *(i)* only data objects within the BPP count boundary must be included, since data objects outside this scope do not represent any development effort in a BPM project that takes into account the perspective of the main participant; *(ii)* event messages outside the BPP count boundary must be included, since they are used to model a communication type between the main participant of the PM-BPMN and the other participants; thus, they represent a development effort that also includes the perspective of the main participant; and *(iii)* data stores outside the BPP count boundary must be included in case they are associated with activities within the boundary, since they represent the persistent data that are maintained by the main participant. On the basis of these conventions, primary intent can be analyzed as follows: *(i)* the primary intent of an ILF is to represent a DF which is referenced (through an association with an activity) within the BPP count boundary; and (*ii)* the primary intent of an EIF is to represent a DF which is referenced (through an association with an activity) outside the BPP count boundary.

Step 3.1.3 – Determine the ILFs and EIFs complexity

In FPA, DF complexity is defined in terms of specific information from the functional context of software, which is not available in the BPM domain. For BPPA, DF complexity is based on the number of interRelated DSs (RDS) and the number of Relationships between DF and Activities (RDA). RDSs and RDAs are calculated in Step 3.1.1. RDS refers to the number of grouped DSs that form each DF. RDA refers to the number of relationships linking the DSs to activities, in each DF. Finally, both values are related and the complexity of each DF can be found with the aid of Table 2.a.

Table A		**RDAs**		
		1-19	20-50	>50
RDSs	1	Low	Low	Average
	2-5	Low	Average	High
	>5	Average	High	High

Table B		**Type**	
		ILF	EIF
Process Complexity	Low	7	5
	Average	10	7
	High	15	10

Table 2. a) Process complexity of DFs; b) Size in BPP of DFs - Adapted from IFPUG (2010)

Step 3.1.4 – Determine the ILFs and EIFs contribution

By means of the same FPA values, the size of each DF for BPPA can be determined using Table 2.b.

Stage 3.2 – Measure transactional functions

Transactional Function (TF) is the second of the two central items for both the FP and BPP counts. For BPPA, TFs must satisfy one or more transactional action requirements such as data processing, validation and management within a PM-BPMN scope. In BPPA, TFs can be External Input (EI), External Output (EO) or External Query (EQ). The steps for measuring TFs are described as follows:

Step 3.2.1 – Identify each elementary process

In FPA, Elementary Processes (EPs) are used to form TFs (smaller actions – atomic or non-atomic – are grouped to form a transaction). As in FPA, BPPA only requires taking account of activities within the BPP count boundary to identify EPs. Activities outside this scope do not represent any development effort during a BPM project that is based on the perspective of the main participant. Activities represent actions carried out during the enactment of a PM-BPMN instance and are a generic term for the work that organizations perform in a process. An EP is the smallest set of interrelated activities within a BPP count boundary that is meaningful to its stakeholders. Before identifying the TFs, the EPs must first be identified. The following BPMN elements are included in the EP identification: Activities (Task, Transaction, Event sub-process, Call activity), Activity Markers (Sub-process, Loop, Parallel MI, Sequential MI, Ad hoc, Compensation), Task Types (Send, Receive, User, Manual, Business Rule, Service, Script), Gateways, Events and Sequence Flows. The activities can be classified by their degree of complexity. Atomic activities (e.g. tasks, including the different task types) are classified as "Simple Activities"; non-atomic activities, as "Complex Activities" (i.e. all other activity types, which are not simple tasks). The "Complex Activities" can be directly classified as EPs, whereas "Simple Activities" must be grouped to form an EP.

The procedure to identify the EPs is: *(1)* identify all activities within the BPP count boundary; *(2)* categorize the activities with regard to their level of complexity: simple and complex activities; *(3)* identify the task type for the simple activities and the activity marker for the complex activities; *(4)*

identify the links between: activities and gateways, and activities and events; *(5)* highlight the sequence in which the identified activities, gateways and events are linked through sequence flows; *(6)* identify the relations between activities and DFs; *(7)* group one or more simple activities if they are semantically dependent, to form a complex activity; (8) classify each resulting complex activity as an EP; and (9) count the number of grouped activities that form the resulting EPs.

Step 3.2.2 – Determine unique elementary processes

As in the case of the FPA, before identifying the resulting TFs for BPPA, a uniqueness test has to be carried out for the identified EPs, since TFs must be unique EPs. To verify if an EP is unique within the set of the identified EPs, it cannot be both formed by a semantically equivalent set of activities and be related to the same DFs when compared to one or more EPs; otherwise they represent a same EP. After checking all the possible combinations of EPs, the resulting unique EPs are classified as TFs.

Step 3.2.3 – Classify each transactional function as an EI, EO or EQ

Each TF that is identified in the previous step must be classified as an EI, EO or EQ. For each TF, it is necessary to check: the primary intent, the associated processing logic and the relationship to the BPP count boundary. The following rules govern the classification process: *(i)* the primary intent of an EI is to represent a TF that processes data received from outside the BPP count boundary. This can be implicitly or explicitly represented by some data object, data store or event message; *(ii)* the primary intent of an EO is to represent a TF that sends data outside the BPP count boundary and involves a prior processing logic to retrieve and process internally information from data objects, data stores or event messages, which does not happen to EQs; and (*iii)* the primary intent of an EQ is to represent a TF that sends data outside of the BPP count boundary by simple information retrieval from a data object, data store or event message. No additional processing logic to retrieve or process internal information is allowed, otherwise it will be considered to be an EO.

Step 3.2.4 – Determine the EIs, EOs and EQs complexity

In FPA, TF complexity is defined on the basis of some specific information from the functional context of the software, that is not available on the BPM context. For BPPA, DF complexity depends on: the number of interRelated Activities (RA) and the value of the TF's Weight (TFW). RAs must be calculated during Step 3.2.1 and refer to the number of activities grouped to form each resulting TF. TFW must be calculated by analyzing the task types and activity markers of the activities that compose each resulting TF, which were identified in Step 3.2.1. In calculating the total weight of a TF, each activity that is part of the group forming the TF provides a different individual weight depending on its type. The following individual weights should be taken into account: *(i)* task types – send (1), receive (1), user (3), manual (0), business rule (2), script (1), service (3 for services which are to be implemented and invoked; or 1 for services which are to be only invoked), abstract or not defined (2); *(ii)* activity markers – loop (2), parallel MI (3), sequential MI (3), compensation (3); and *(iii)* sub-process activities, including transaction, event sub-process, call activity, regular sub-process, ad hoc – the types of activities forming these sub-processes must be taken into account to calculate their individual weights. Finally, both values are related and the complexity of each DF can be found using Table 3.a (for EIs) or Table 3.b (for EOs and EQs).

Table A		**Ras**		
		1-4	5-15	>15
TFW (EI)	0-1	Low	Low	Average
	2	Low	Average	High
	>2	Average	High	High

Table B		**Ras**		
		1-5	6-19	>19
TFW (EO/EQ)	0-1	Low	Low	Average
	2-3	Low	Average	High
	>3	Average	High	High

Table 3. Process complexity of TFs: a) Complexity of EIs and b) Complexity of EOs and EQs - Adapted from IFPUG (2010)

Step 3.2.5 – Determine the EIs, EOs and EQs contribution

By means of the same FPA values, the size of each TF for BPPA is determined by using Table 4.

		Type		
		EI	EO	EQ
Process Complexity	Low	3	4	3
	Average	4	5	4
	High	6	7	6

Table 4. Size in BPP of TFs - Adapted from IFPUG (2010)

Stage 4 – Calculate process size

According to IFPUG (2010, p.14), "the purpose and count scope shall be considered when selecting and using the appropriate formula to calculate the functional size", as this is a result from the counting process. Since this version of BPPA takes into account only the development project type, the following formula is used to calculate its process size: **PSP = CDF + CTF**, where: **PSP** is the **P**rocess **S**ize of a given BPM **P**roject, **CDF** is the total value as a result of the final **C**ontribution of all the **DF**s identified for the given count scope, and **CTF** is the total value as a result of the final **C**ontribution of all the **TF**s identified for the given count scope.

Stage 5 – Document the BPP count

Finally, the BPP count can be documented; it contains the following information: the count purpose and type; the count scope and boundary; the count data; a list of all DFs and TFs identified, including their types, complexities and number of BPPs assigned to each one; the final result of the count (Stage 6); and, any assumptions made about the PM-BPMN, as well as any issues that have been resolved.

Stage 6 – Report the result of the BPP count

According to IFPUG (2010, part 1, p.23-24), "(...) consistently reporting the results of functional point counts enables the readers to identify the standard to which they comply". As the last stage of the BPPA technique, the report of the results of the BPP count is defined as follows: **S BPP (BPPA-v)**, where: **S** is the result of the BPP count (Stage 4); **BPP** is the size unit of the BPPA technique; and **v** represents the version of the BPPA technique that was used to obtain the process size being reported.

4 Evaluation

This session provides a brief account of two empirical evaluations undertaken to determine the benefits and drawbacks of the BPPA technique. The first is based on the authors’ observations when applying the proposed technique to a use scenario. The second is based on a specialist's observations when reading the published technique as a technical report.

4.1 Example of an Application

The scenario chosen to perform the evaluation of the BPPA technique being proposed is "The Travel Booking" (BPMI, 2011), a BPM development project composed of only one PM-BPMN, the Travel Booking model, which is shown in Figure 2.

Referring to the illustrative PM-BPMN in Figure 2, there are 10 activities (two sub-processes and 8 simple tasks), 12 events (4 of which are message events), excluding those inside the sub-processes, 4 gateways excluding those inside the sub-processes, and sequence flows between the activities, events and gateways. The general result of applying the BPPA technique to the Travel Booking model indicated that the result of the count was 53 BPP. This value includes the addition of 3 DFs (2 Low ILFs, and 1 Low EIF) and 9 TFs (2 Low EIs, 1 High EI, 3 Low EQs, 1 Average EQ, 1 Low EO, and 1 Average EO). The number of BPPs, like the number of FPs, does not represent any information in itself. On the other hand, it is of value in comparative situations (e.g. comparing the number of BPPs from one project with others and checking whether it requires more effort or not). In addition, BPPs can be used to plan and manage a BPM project in association to comparative historical data.

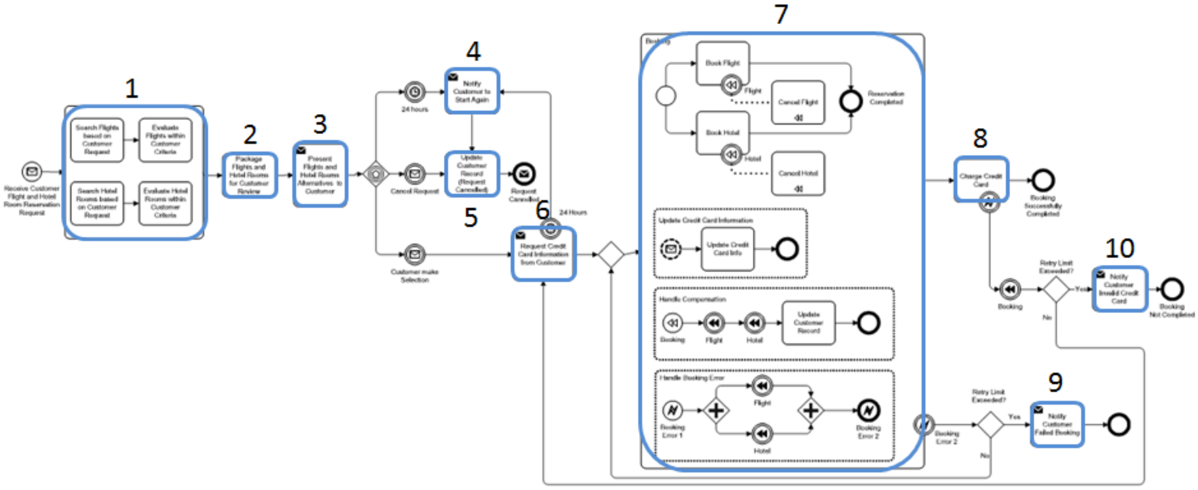

Figure 2. The Travel Booking PM-BPMN - Adapted from BPMI (2001)

The main outcome of this evaluation is that, in general terms, BPPA can be applied as designed and proposed in this paper. A typical scenario is used to test the application of BPPA and the BPP metric can be calculated, to show the feasibility of its use. Some additional results are discussed as follows:

- The chosen scenario only includes a small number of data elements, which can be observed by comparing the number of DFs to the number of TFs. In fact, only data elements of the "event message" type are identified, with no data object or store being identified. Given the knowledge domain of the researchers that proposed BPPA, this is a common picture for BPs modeled through BPMN in which no data is explicitly modeled. Unlike the software domain, both data and transactions have almost the same importance in terms of initial representation. In the BPM domain, they only address the transactional issue in terms of different types of activities;
- With regard to this specific scenario, most of the DFs and TFs have been identified and calculated as "low" complexity (75%). We believe that this is the result of relying on the same factors used in FPA to calculate the DFs and TFs complexities (see Table 2 and 3). Although this is not necessarily an issue, typical scenarios in BPM projects should be scattered at different levels of complexity;
- In this application, we noted that, when measuring TFs, only activities are included that are points to compose the final BPP metric, whereas other elements in the sequence flows, such as gateways and messages, are used to understand the context of the modeled problem and identify the EPs and TFs. However, it is known that there are some implementation efforts based on these two types of BPMN elements during the BP development phase. As in the case of the previous item, there is clearly a possible need to include these elements in the TFW count in Table 3.

4.2 Analysis by a specialist in the field

An invited software metrics specialist has evaluated the BPPA technique to verify that it strictly adheres to the FPA technique. The specialist has been an IFPUG Certified FP Specialist since 2001 and works at a very well known Software Measuring company. His analysis focused on Stages 1, 2 and 3 of the BPPA technique as described in the technical report, which contains a complete description of BPPA. The main results of this empirical evaluation (based on the specialist's observations), as well as by some analysis conducted from the BPM perspective, are given as follows.

Regarding Stage 1 of BPPA, the specialist believes that "the BPMN notation could not be considered part of the user language to represent FURs; the users might be trained to understand it, but they are usually not fluent in it". This concern might be more valid in the FPA context, where the user could be anyone interacting with or using the software. We understand that this is not valid in BPM, since users are more specialized, e.g. business analysts or process analysts normally fluent in BPMN. In BPM projects, the users are usually representatives of the team who are interested in the BP development.

Regarding Stage 2 of BPPA, the specialist points out that "BPPA has only one type of count – the development project type –, not addressing the application and enhancement types". Although this is an actual limitation of the proposed technique, we understand that this first version of the BPPA technique is addressing the most important type, since it covers the first two phases of the BPM life-cycle (Weske, 2007): "Design and Analysis", when the PM-BPMN is created, and "Configuration", when the implementation effort being estimated is applied. The application and enhancement project types must be addressed, so that the next two phases of the BPM life-cycle can be respectively covered: "Enactment" and "Evaluation". Furthermore, the specialist points out that "according to IFPUG (2010), the border should be regardless of technical or implementation considerations; thus, placing a BPMN element such as a pool or swimlane as a boundary delimiter is tying the concept of how the modeling negotiation was performed". Although this is still an open challenge for this study, identifying the BPP count boundary is a tough and very important task. Once the BPMN notation has been chosen as the language to represent FURs, it is not possible to identify a border regardless of technical considerations or, in this case, modeling considerations.

Regarding Stage 3.1 of BPPA, the specialist points out that "due to the flexibility of the BPMN notation, business data may be present in different structures, as mentioned in the document describing the BPPA technique; then, the analyst will need to identify and determine exactly the different types of data being represented in the model". Since the main source of information for BPP count is PM-BPMN, it is important for the analysts to know how to differentiate the data being represented in a BPMN model so that the BPMN technique can be applied as proposed; but, as already discussed, we do not believe that the involved analysts will have problems in learning about them.

Regarding Stage 3.2 of BPPA, the specialist points out that "non-atomic activities contain several tasks and can be related to other exception tasks, e.g. a task that treats an exception condition or reports that a particular transaction cannot be performed; in FPA, exceptional conditions or messages, sent to inform the user that the transaction could not perform some action, are part of the transaction's EP and, hence, non-atomic activities may or may not be EPs. So, it is not possible to determine a 1-1 mapping between these two concepts". For BPMN, it is difficult to identify whether a task should compose a larger EP or not. We have defined some specific rules on how to identify EPs by grouping smaller activities into bigger ones.

We conclude that most of the limitations pointed out by the specialist are due to some special differences between both the target artifacts BP and the software; this means treatments in the BPPA technique should be different from those employed in the FPA technique. Moreover, the specialist did not point out large divergences in the most relevant stages of the BPPA technique – 3.1 and 3.2.

5 Conclusion

This paper proposed a technique, called BPPA (Business Process Point Analysis), for measuring the size of BPM projects based on PM-BPMNs and which can help project managers in their attempts to introduce work estimation for process development projects. Although BPPA is built on the Function Point Analysis (FPA) framework and hence the procedures of both are very similar, there are some FPA features, which were originally proposed for the software. The similarity between BPPA and FPA has been confirmed by an FPA specialist who evaluated BPPA. In his technical judgments, he concluded that BPPA concepts are only partially discordant with FPA. BPPA requires a good deal of effort so that it can be used by organizations in BPM projects.

Future works should include the following: adjusting the factors that influence the definition of the process complexities of DFs and TFs; enlarging the scope of BPP counts to consider the application and enhancement project types; and reducing the dependence on personal considerations during the BPP counts. Moreover, other studies should be undertaken to perform additional analyses through controlled experiments or case studies. These new studies should strengthen the approach evaluation in terms of: *(i)* including impartial specialists from the BPM domain as well as more specialists from the software metrics field; *(ii)* running an experiment in a real-world setting; *(iii)* increasing validity assurance by defining a set of objective metrics to be applied through a clear protocol; and *(iv)* leading to a comparative approach between BPPA and other techniques related to BPM measurement.

Acknowledgments

This work was supported by The State of São Paulo Research Foundation (Fapesp), Brazil. The authors thank Mario Luiz Varella, the FPA specialist who has evaluated BPPA.

References

Aguiar, M. (2009). Function Points or Use Case Points. IFPUG (International Function Point Users Group) MetricViews Summer 2009. Available in: < http://metricas.com.br/downloads/Function_Points_or_Use_Case_Points.pdf>. Accessed in: March, 31st 2013.

Baklizky, M, Fantinato, M (2012). FPA4BPM – Function Point Analysis for Business Process Management (v.1.0) (In Portuguese). Technical Report. Available in: <http://ppgsi.each.usp.br/ arquivos/RelTec/PPgSI-003_2012.pdf >. Accessed in: March, 31st 2013.

BPMI, Business Process Management Initiative (2011). Business Process Model and Notation (BPMN), release 2.0. Object Management Group (OMG).

Clemmons, R. K. (2006). Project Estimation with Use Case Points. Crosstalk the Journal of Defense Software Engineering, p. 18-22.

Dhammaraska, K., Intakosum, S. (2009). Measuring Size of Business Process From Use Case Description. Computer Science and Information Technology (ICCSIT) 2nd. p. 600-604.

C.O.S.M.I.C., Common Software Measurement International Consortium (2009). The COSMIC functional size measurement method, version 3.0.1, measurement manual (the COSMIC implementation guide for ISO/IEC 19761: 2003).

Fantinato, M., Toledo, M. B. F., Thom, L. H., Gimenes, I. M. S., Rocha, R. S., Garcia, D. Z. G. A Survey on Reuse in the Business Process Management Domain. Int. J. of Business Process Integration and Management, v. 6, n. 1, p. 52-76, 2012.

Gruhn, V., Laue, R. (2006). Complexity Metrics for Business Process Models. In: Witold Abramowicz and Heinrich C. Mayer, editors, 9th Int. Conf. on Business Information Systems, p. 1-12.

I.F.P.U.G. (2010). Function Point Counting Practices Manual, release 4.3.1. Westerville, Ohio.

I.F.P.U.G. (2007). Function Points & Counting Enterprise Data Warehouses, release 1.0. NEC White Paper.

I.F.P.U.G. (2009). Function Points & Counting Middleware Software Applications, release 1.0. NEC White Paper.

Kaya, M., Demirörs, O. (2011). E-Cosmic: A Business Process Model Based Functional Size Estimation Approach. 37th EUROMICRO Conf. on Software Eng. and Advanced App. p. 404-410.

P.M.I., Project Management Institute (2008). A Guide to the Project Management Body of Knowledge (PMBoK Guide). Project Management Institute, Inc., Pennsylvania, 4th edition.

Sánchez González, L., García, F., Ruiz, F., Piattini, M. (2010). Measurement in Business Processes: A Systematic Review. Business Process Management Journal 16(1), p. 114–134.

Sommerville, I. (2004). Software Engineering. Addison Wesley, 6th edition.

Weske, M. (2007).Business Process Management: Concepts, Languages, Architectures. Springer,1st ed