International Journal of Software Engineering
and Knowledge Engineering
Vol. 21, No. 3 (2011) 311–338

DOI: 10.1142/S0218194011005359

World Scientific
www.worldscientific.com

MEASURING SOFTWARE FUNCTIONAL SIZE FROM BUSINESS PROCESS MODELS

CARLOS MONSALVE[*,†,‡], ALAIN ABRAN[†,§] and ALAIN APRIL[†,¶]

[*]*CIDIS-FIEC, Escuela Superior Politécnica del Litoral*
Km. 30.5 via Perimetral, Guayaquil, Guayas, Ecuador

[†]*Software Engineering Research Laboratory, École de Technologie Supérieure*
1100 rue Notre-Dame Ouest, Montréal, Québec H3C 1K3, Canada
[‡]*monsalve@espol.edu.ec*
[§]*alain.abran@etsmtl.ca*
[¶]*alain.april @etsmtl.ca*

ISO 14143-1 specifies that a functional size measurement (FSM) method must provide measurement procedures to quantify the functional user requirements (FURs) of software. Such quantitative information, functional size, is typically used, for instance, in software estimation. One of the international standards for FSM is the COSMIC FSM method — ISO 19761 — which was designed to be applied both to the business application (BA) software domain and to the real-time software domain. A recurrent problem in FSM is the availability and quality of the inputs required for measurement purposes; that is, well documented FURs. Business process (BP) models, as they are commonly used to gather requirements from the early stages of a project, could be a valuable source of information for FSM. In a previous article, the feasibility of such an approach for the BA domain was analyzed using the Qualigram BP modeling notation. This paper complements that work by: (1) analyzing the use of BPMN for FSM in the BA domain; (2) presenting notation-independent guidelines for the BA domain; and (3) analyzing the possibility of using BP models to perform FSM in the real-time domain. The measurement results obtained from BP models are compared with those of previous FSM case studies.

Keywords: Functional size measurement; FSM; COSMIC; ISO 19761; business process model; business process modeling; Qualigram; BPMN; requirements modeling.

1. Introduction

The functional size of software is an important input for planning, buying, developing, improving, and maintaining software systems [1]. In particular, it provides valuable information for estimating the effort required to develop the measured software. Based on that estimation, software managers can successfully plan resources and estimate costs for the software project [2]. Functional size measurement (FSM) can be performed *a priori* (i.e. based on the project specifications) or *a posteriori* (i.e. based on the finished software product). The former is desirable for planning a software project and the latter for productivity analysis and benchmarking purposes. For estimation, the measurement of functional size should be performed during the early stages of the project.

Several methods have been proposed for FSM, one of which is the COSMIC FSM method [3]. COSMIC was designed to be applied in various functional domains: (1) business application software; (2) real-time software; and (3) a combination of the two. It is completely open and available in multiple languages, and it has been reported to be easy to learn and use [3]. Since 2003, COSMIC has been accepted as an international standard, ISO/IEC 19761:2003 "Software engineering — COSMIC-FFP — A functional size measurement method" [4]. The COSMIC measurement unit is a COSMIC function point (CFP), which represents one "data movement" [3], and the functional size of software is obtained by adding the data movements identified.

Business process (BP) models are designed to be useful for documenting, communicating, and improving organizational business processes. They are also used by software engineers and business analysts to gather the software and system requirements from the early stages of the development process [5–8]. A BP model may therefore be a valuable source of information for FSM.

It was not until recently that the use of BP models for COSMIC FSM has been studied [9, 10]. This article complements the results given in [9] by identifying the candidate rules for mapping the various COSMIC concepts to the Business Process model and Notation (BPMN) [11] constructs. Considering that COSMIC allows the functional size of real-time software to be measured, this article also analyzes the possibility of modeling real-time software using BP models to measure its functional size. The candidate mapping rules and modeling rules to be taken into consideration for FSM purposes are identified for both the business application software domain and the real-time software domain. To verify the value of the proposed approaches, this article compares the results obtained with those of previous FSM case studies. Finally, this article also compares the results obtained with those from [9] to elaborate a set of general guidelines for modeling BPs for FSM purposes in the business application domain. These guidelines are independent of the modeling notation.

The structure of this paper is as follows. Section 2 reviews related works. Section 3 introduces the BP modeling notations that are explored in this research. Section 4 describes the methodology used in this research. Sections 5 and 6 present the use of a BP model for FSM in the business application domain and in the real-time domain respectively. Section 7 discusses the results obtained. Finally, Sec. 8 concludes the paper with a review of the contributions of this research, its limitations, and future work.

2. Related Works

Two of the research works related to the feasibility of using a BP model for FSM that could be identified before the publication date of [9] do not make use of COSMIC as the proposed FSM method. The first work identified [12] is based on the use of the IFPUG Function Point Analysis (FPA) method, release 4.0 [13]. It proposed an approach for measuring reuse "in the requirements conceptualization phase" of an enterprise resource planning (ERP) software project. To achieve its goal, the approach proposed to map the various modeling concepts of a commercial ERP software development tool to the "base logical components" of the IFPUG FPA method. The FSM method was used to measure "the size of the reusable requirements and the size of the total requirements" as indirect measurements of requirements reuse in an ERP "implementation project". The development tool used the Event-driven Process Chain (EPC) [14] diagrams to represent business requirements as BP models. Therefore, the mapping rules included the mapping of various EPC modeling concepts to the "base logical components" of the IFPUG FPA method. The second work [15] used an extension of the EPC to model a military application. In this case, the EPC diagrams were used as part of the requirements elicitation methodology, but without the need to map the EPC modeling constructs to the concepts of an FSM method.

The use of conceptual models for FSM with the COSMIC method has been studied and analyzed in the research literature, and a complete survey of related works, including its own, is offered in [16]. Following publication of that survey, Lavazza and Bianco [17] studied the use of Unified Modeling Language (UML) [18] diagrams (use case, component, and sequence diagrams) for modeling real-time software to be measured using the COSMIC FSM method. In other work [19], the potential relationships between the measurements obtained from UML use case diagrams and those obtained from other UML diagrams were studied. From all these works, only one [20] has included the use of some kind of BP model. The annotated work proposed the use of UML activity diagrams as one of the possible options for representing the behavioral aspects of the software being modeled; however, it does not provide a rule for mapping between the BP modeling constructs and the COSMIC concepts. Moreover, the emphasis of this latter work is not related to the feasibility of using only BP models for FSM.

A more recent work [9] analyzed the feasibility of using a BP model developed in Qualigram [21] for FSM in the business application domain. Qualigram is a management-oriented BP modeling notation based on the results of an international research project [22, 23]. More details of this notation are provided in section 3. Ten modeling rules for FSM purposes were derived, as well as the necessary rules for mapping the various COSMIC concepts to the corresponding Qualigram modeling constructs. Based on the former rules, the specifications of the February 23, 2008, version of the C-Registration System case study [24] were modeled. The mapping rules were applied in the BP models to identify the data movements, and the measurement results were obtained by adding those data movements (see Table 1). Finally, the measurement results were compared with those obtained in the case study.

Table 1. Measurement results based on Qualigram: Business application domain — The C-Registration System case study [24].

Procedures	Data movements E	X	R	W	CFP
Login	1	2	1		4
Add Professor	1	2	1	1	5
Modify Professor	2	2	1	1	6
Delete Professor	3	3	1	1	8
Select Courses to Teach	4	6			10
Add Student	1	1	1	1	4
Modify Student	2	2	1	1	6
Delete Student	3	3	1	1	8
Create Schedule	4	5	2	1	12
Modify Schedule	4	6	3	1	14
Delete Schedule	3	4	2	1	10
Close Registration	3	5	2	1	11
Submit Grades	4	4	2	1	11
View Report Card	1	2	2		5
Total	36	47	20	11	114

3. Why Test the Proposed Approach with Two BP Modeling Notations?

Two factors that are key to the successful modeling of business processes are: (1) the use of an appropriate BP modeling notation [25]; and (2) the active participation of all the stakeholders, along with a shared vision of business processes [8, 26]. Unfortunately, the evidence shows that business processes are not consistently documented [27], and that management and IT stakeholders tend to use different notations, conventions, and techniques to represent them [26]. Consequently, various notations for modeling business processes have been developed over the last 20 years, most of them responding to an IT-oriented perspective of BP modeling.

For this paper, two BP modeling notations have been selected: (1) the Business Process Model and Notation (BPMN), because of its popularity and because of the considerable effort under way to establish it as a BP modeling standard [11, 28, 29]; and (2) the Qualigram modeling notation [21], because it is a management-oriented BP modeling notation that incorporates different levels of abstraction, and is based on the results of an extensive international research project. With this selection, it is possible to test the proposed approach using both an IT-oriented BP modeling notation and a management-oriented BP modeling notation. Moreover, the comparative analysis of the results obtained with both BP modeling notations leads to the proposal of a set of notation-independent BP modeling rules for FSM purposes. Section 3.1 presents a short introduction to BPMN, and Sec. 3.2 describes the Qualigram modeling notation in more detail.

3.1. *Business Process Model and Notation (BPMN)*

BPMN is currently an Object Management Group (OMG) standard [11]. It was initially developed by the Business Process Management Initiative (BPMI), and in 2004 the BPMN 1.0 specification was released [29]. Later, in 2005, there was a merger between BPMI and OMG, BPMN having been adopted by the latter.

BPMN was created with the idea of providing a unified notation, both for IT and for management stakeholders, that is easy to understand, but at the same time having a formal basis [28–31]. For this purpose, the standard includes a basic set of constructs called the "Business Process Diagram (BPD) Core Element Set" (Core Set), and a more complete set, "BPD Extended Set" (Extended Set). The first set is intended for documentation and communication purposes, and the second set for developing more detailed models that are appropriate for the analysis and automation of business processes. According to [27] and [31], the adoption rate of BPMN is increasing in industry.

BPMN is a modeling notation rich in modeling constructs for representing various types of control flow and events. As a result, BPMN has a high degree of expressiveness, but at the same time is highly complex [32]. According to a recent study [33] based on the analysis of 120 BP models, of the 50 modeling constructs offered by BPMN, a typical BP model uses only 9. Those 9 constructs may vary from BP model to BP model in an arbitrary way. However, only 4 modeling constructs were always used by the 120 BP models, and some of the BPMN modeling constructs were never used.

3.2. *Qualigram*

Qualigram [21] is a management-oriented modeling notation intended for the documentation and communication of business processes. Qualigram proposes three levels of abstraction. The top level (strategic level) models the processes, answering the questions "why" and "where to", and deals with the mission, objectives, and policies of the organization. The intermediate level (organizational level) models the procedures, answering the questions "who" and "what", and describes how to achieve the objectives of the organization. Finally, the lowest level (operational level) models the work instructions, answering the questions "how" and "using what", and deals with the control of specific tasks. "A process is constituted by a set of procedures; a procedure is constituted by a set of work instructions; and an instruction is constituted by a set of elementary operations" [21]. These concepts are depicted in Fig. 1.

It is important to point out that the Qualigram conception of a process model (i.e. top-level model) is somewhat different from the mainstream notion of a business process model. Actually, a Qualigram procedure model (i.e. intermediate-level model) is closer to what is typically understood as a business process model. In this paper, the term "BP model" is generic, and encompasses the variations and levels of detail that each modeling notation or author may prefer to use to represent

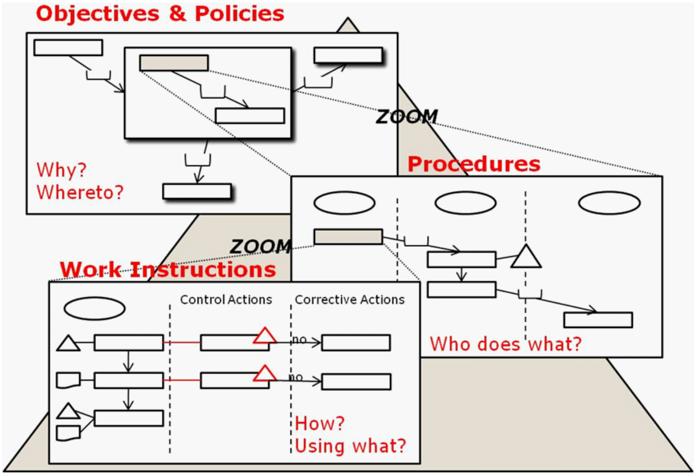

Fig. 1. Qualigram pyramid, adapted from [21].

an organization's process. The reader should not infer, therefore, that a Qualigram process model has exactly the same general scope as a BP model. Moreover, because this paper uses the term "BP model" generically, both Qualigram process models and Qualigram procedure models are considered as BP models with different levels of abstraction.

Another characteristic of Qualigram is its simplicity. The modeling constructs for each level are based on a set of four basic concepts, along with their corresponding graphical forms: (1) action; (2) entity; (3) tool; and (4) information [21]. Variations of the action form are used to represent processes, procedures, work instructions, and elementary operations. Variations of the entity form are used to represent roles (internal and external), units, and external entities. The tool form is used to represent any kind of physical tool or equipment, as well as any kind of document produced or used by an action. The information form is used to represent the input and output flows of information between the various elements modeled. Qualigram claims that its simplicity makes its notation clear enough to be understood by any type of stakeholder of the organization. These concepts are depicted in Fig. 2.

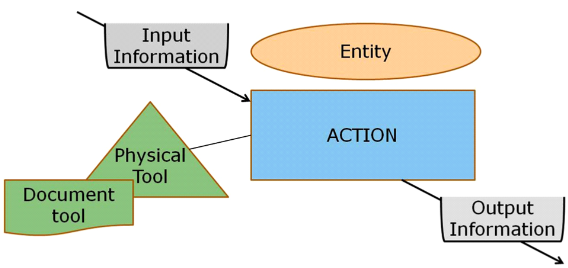

Fig. 2. Basic graphical forms of the Qualigram notation, adapted from [21].

4. Methodology

The methodology used in this research is twofold: (1) The steps to be followed for the business application software domain, as explained in Sec. 4.1; (2) The steps to be followed for the real-time software domain, as explained in Sec. 4.2.

4.1. *Business application domain*

Figure 3 depicts the methodology for the business application software domain. The same methodology is followed for each of the selected BP modeling notations. To test the feasibility of the proposed approach, the version of the C-Registration System case study dated February 23, 2008, and published by the COSMIC Group is used [24]. Based on the definitions of the various modeling constructs offered by the modeling notation, and the definitions of the various COSMIC concepts, a mapping table of the COSMIC concepts and the modeling constructs is generated. Also, as a result of the comparison, a set of specific modeling rules is identified to allow the BP models to be used for FSM. The C-Registration System is modeled following these modeling rules. The mapping rules and the BP models are used to measure the functional size of the system. Finally, the measurement results are compared with those presented in the C-Registration System case study.

In addition, the results obtained using each of the BP modeling notations are analytically compared, in order to generate a set of notation-independent BP modeling guidelines for FSM purposes.

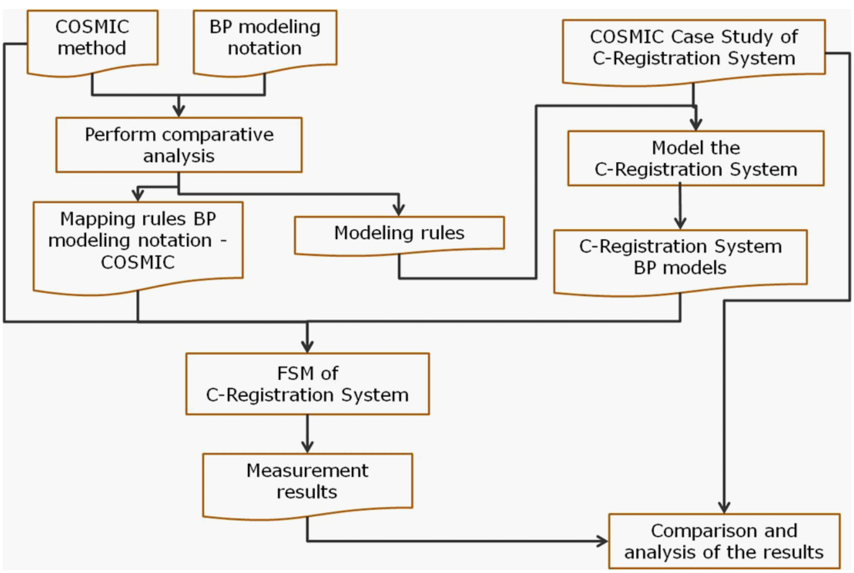

Fig. 3. Methodology for the business application software domain.

4.2. *Real-time domain*

The methodology for the real-time software domain is very similar to that depicted in Fig. 3. The main differences are the case study to be analyzed and the way the feasibility of the proposed approach is tested. The May 22, 2008, version of the Rice Cooker case study [34] is used to illustrate the approach. To verify the value of the approach, the results obtained are compared with those obtained by [17] for the same case study. From the two selected BP modeling notations (i.e. BPMN and Qualigram), only Qualigram is used for analyzing the real-time domain. Since a BP modeling notation is typically intended to model the business processes of an organization, and the case study corresponds to a real-time software controller, it is very likely that some specific modeling rules for FSM purposes have to be derived. Finally, as in the previous case, a set of mapping rules is elaborated and used to measure the functional size of the software components of the Rice Cooker system. This methodology is depicted in Fig. 4.

5. FSM based on a BP Model: The Business Application Domain

The **purpose** of this section is to measure the functional size of the C-Registration System, based on a set of BP models, in order to analyze the feasibility of using them as the source of information for FSM. Therefore, the **scope** of this measurement is

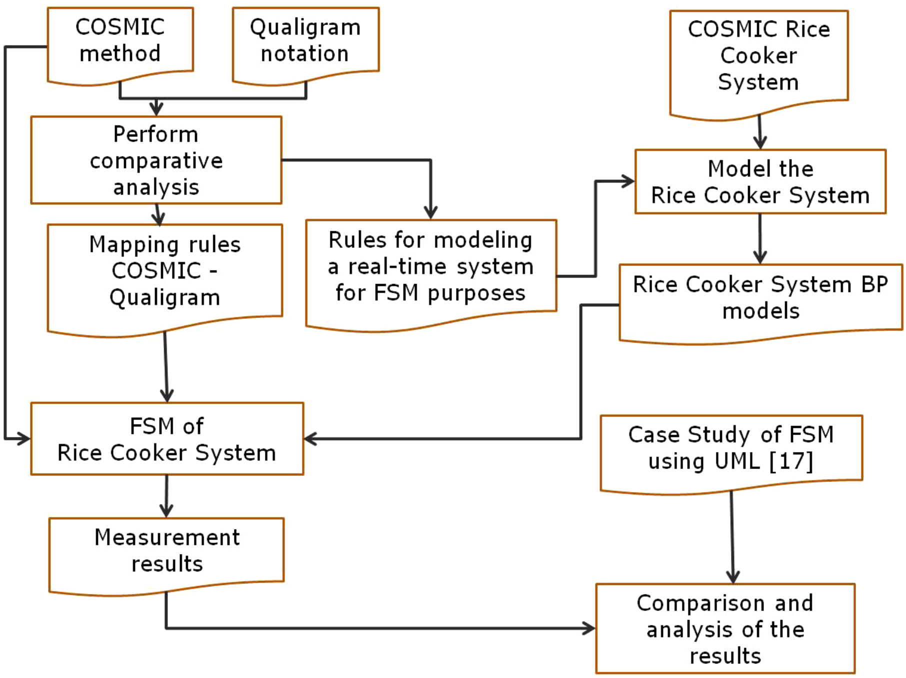

Fig. 4. Methodology for the real-time software domain.

given by all the functional user requirements (FURs) of the C-Registration System, as described in [24]. The C-Registration System is business application software that belongs to the "application **layer**" of the "typical layered software architecture" [3].

Two BP modeling notations are used: Qualigram and BPMN. In the next subsection, the results of using the Qualigram models that were obtained in [9] for FSM purposes are summarized, including the determination that the appropriate Qualigram level of abstraction is in agreement with the level of granularity expected by the COSMIC FSM method. In Sec. 5.2, the specific modeling rules for producing BPMN models suitable for use for FSM are identified. In Sec. 5.3, the rules for mapping between COSMIC and BPMN are defined to ultimately measure the functional size of the C-Registration System. Finally, in Sec. 5.4, a comparative analysis of the results obtained with Qualigram and BPMN is performed in order to derive a set of notation-independent BP modeling guidelines for FSM purposes.

5.1. *Results obtained with Qualigram*

This subsection is entirely based on the results obtained in [9]. Ten modeling rules for producing Qualigram models suitable to be used for FSM were identified, the first four to be applied at the top level of abstraction (i.e. the strategic level) and the last six at the intermediate level of abstraction (i.e. the organizational level). According to COSMIC, the recommended **level of granularity** of the FURs is achieved when the functional users: (1) are individuals; and (2) "detect single occurrences of events". According to [9], these conditions seem to be satisfied with the intermediate level of Qualigram (i.e. the organizational level).

Modeling Rule Q.BA1. *At the top level of abstraction* (*i.e. the strategic level*), *represent the software to be measured as a process.*
Modeling Rule Q.BA2. *Following COSMIC principles*, *consider any external software component that interacts with the measured software as an external entity.*
Modeling Rule Q.BA3. *Consider any logical instruction set that is worth detailing in more depth as a procedure.*
Modeling Rule Q.BA4. *Represent any user of the software who allows representation of the inputs and outputs of the procedures modeled as an external entity.*

Based on these four modeling rules, the C-Registration System was represented with a top-level Qualigram model (i.e. a strategic level model), as depicted in Fig. 5.

Modeling Rule Q.BA5. *At the intermediate level of abstraction* (*i.e. the organizational level*), *represent the software being measured as an internal role.*
Modeling Rule Q.BA6. *At the intermediate level of abstraction* (*i.e. the organizational level*), *represent any peer software component that interacts with the software being measured as an external role.*

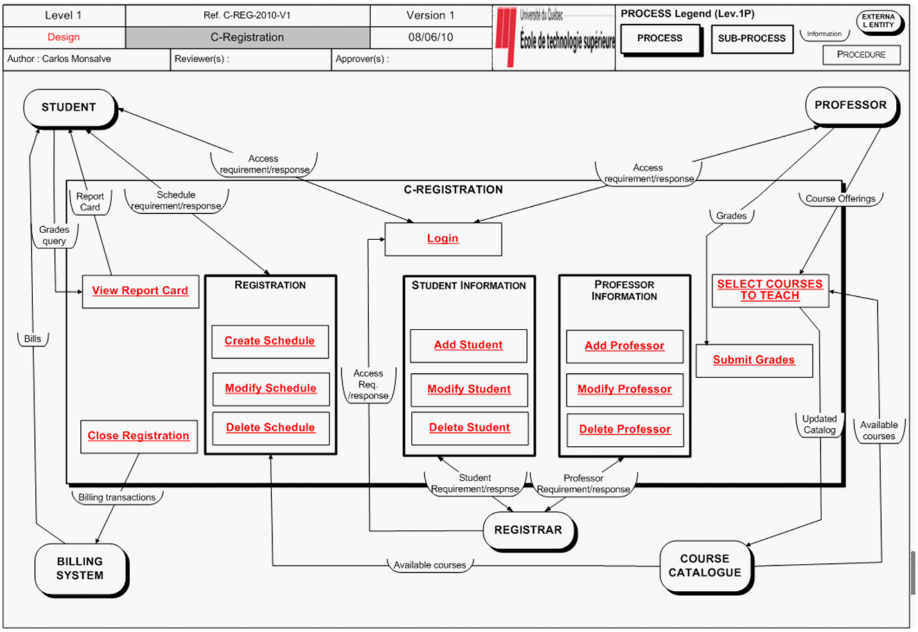

Fig. 5. Top-level Qualigram model of the C-Registration System.

Modeling Rule Q.BA7. *Any instruction that requires retrieving or writing relevant data from/to a persistent storage should be associated with a material tool. That tool has to be labeled, indicating the type of operation to be applied to the persistent storage: R for retrieve, W for write.*

Modeling Rule Q.BA8. *If the procedure being modeled requires, at its inception, information to be entered by the role that triggered it, represent the initial submission of information as the triggering event.*

Modeling Rule Q.BA9. *All the error conditions identified by a role must be collected by a unique instruction executed by the same role before reporting them to another role.*

Modeling Rule Q.BA10. *Avoid representing flows of information between roles when those flows are only aimed at indicating a possible end to the workflow.*

Based on the specifications of the system [24] and the annotated modeling rules, each of the functional processes of the C-Registration System were modeled at the Qualigram intermediate level of abstraction (i.e. the organizational level). For example, Fig. 6 presents the model corresponding to the "Add Professor" functional process.

From the analysis of the models generated in Qualigram, the rules for mapping between the COSMIC concepts and the Qualigram modeling constructs were defined. Table 2 shows all the mapping rules that were applied to each of the models representing the functional processes of the C-Registration System. The

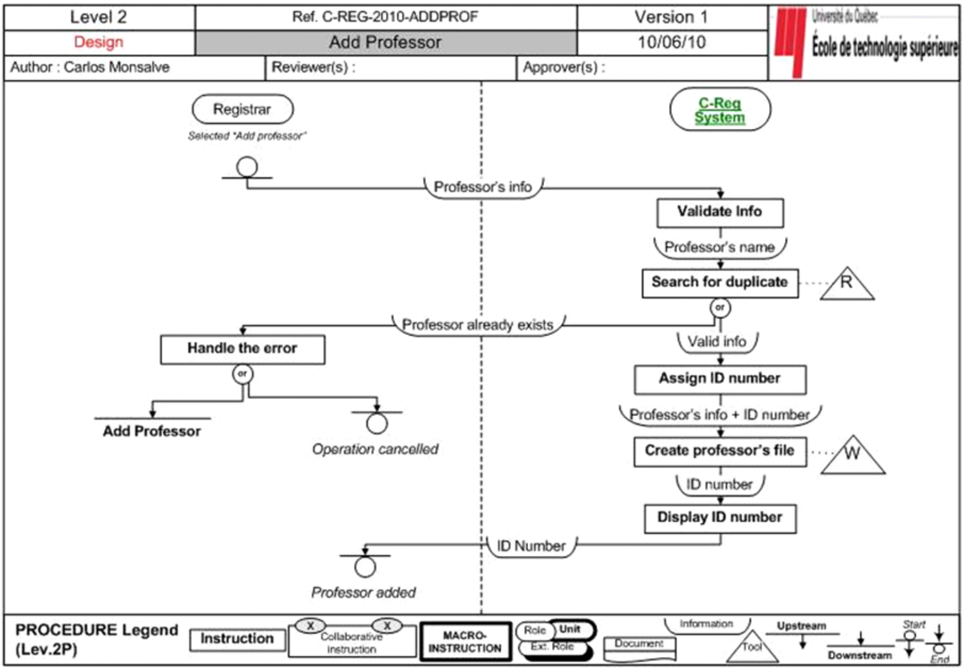

Fig. 6. Qualigram model corresponding to the "Add Professor" functional process.

Table 2. Rules for mapping between COSMIC and Qualigram.

COSMIC FSM method V.3.0.1	Qualigram notation	Comments
Functional User	Role	Only those roles that interact with the software
Boundary	The process box that represents the software	Top-level model
	The swim-lane of the role that represents the software	Intermediate-level model
Functional Process	Procedure	The procedures included in the process box of the software
Triggering Event	Triggering element	
Data Group	May be provided as part of the information flow	Between roles
	May be provided for describing the material tool	For an instruction that requires access to a persistent storage
Entry	An incoming flow of information	
Exit	An outgoing flow of information	
Read	Description (R) given in a material tool	
Write	Description (W) given in a material tool	

measurement results were obtained next, by simply adding the various data movements (Entries (E), Exits (X), Writes (W), and Reads (R)) that appeared in the models.

Table 1 shows the measurement results obtained in [9]. Finally, the measurement results were compared with those obtained in the case study [24]. The COSMIC case study presents two sets of results: "step 1" and "step 2". The first set is obtained after applying the COSMIC FSM method to the FURs "exactly as they are written" in the original specifications of the C-Registration System. The second set results from modifying the FURs in step 1 "by a further assumption". This paper has only considered the FURs as given in step 1 of the case study. The comparison of the results is presented in Sec. 5.3.

5.2. *Modeling rules for BPMN*

This section uses BPMN version 1.2 (BPMN 1.2) [11] for modeling the specifications of the C-Registration System and for deriving the modeling rules for FSM purposes. There is a version 2.0 of BPMN (BPMN 2.0) [35], but it is still considered as a Beta 2 version at the time of writing this paper. When the term BPMN is used in this paper without any reference to either of the two versions, it has to be understood that, for understanding the authors' argument, the BPMN version does not affect the meaning of their assertion.

BPMN does not offer the possibility of representing the C-Registration System by a model with similar characteristics to the one depicted in Fig. 5. In BPMN, it is always necessary to represent the workflow of the business process; i.e. each business process has at least one clear start event that triggers the first activity (task or sub-process), after which a finite set of activities is executed following a predetermined flow that finishes at a clear end event [11]. A business process may have multiple end events. In BPMN 1.2, a BP should be contained in a pool, and, even if it is not drawn, it is "implied by default" [36]. A BP can interact with any external participant (customer, provider, external actor, other BP) through sending and receiving messages [11]. In these cases, the external participant is considered as an external BP and may be represented as a pool in the BPMN diagram. In order to differentiate between the pool of the BP and the pool of any external participant, this research will refer to them as "main pool" and "secondary pool" respectively. A pool may be partitioned into lanes, which are used to represent any organization or categorization of activities [11]. Typically, lanes are used to represent "performer roles or organizational units" [36].

Modeling Rule BPMN1. *Consider any logical instruction set that is worth detailing as a separate BP.*

Modeling Rule BPMN2. *Represent the software to be measured as a lane in the main pool.*

Modeling Rule BPMN3. *Represent any external software component that interacts with the measured software as a secondary pool.*

Modeling Rule BPMN4. *Represent any user of the software as a secondary pool (external user) or as a lane in the main pool (internal user).*
Modeling Rule BPMN5. *All the error conditions identified within the lane that represents the software to be measured must be collected by a unique event or a unique activity before reporting them to another lane or pool.*

Figure 7 presents the application of these modeling rules for representing the "Select Courses" functional process based on the requirements of the system [24].

Modeling Rule BPMN6. *Avoid representing a sequence flow between lanes or a message between pools when that flow or message is only aimed at indicating a possible end to the workflow.*
Modeling Rule BPMN7. *Any modeling construct that requires retrieving or writing relevant data from/to persistent storage should be associated with a data object.*
Modeling Rule BPMN8. *Use link events when the lane of the software to be measured must be crossed in order to return to an activity (e.g. for representing a feedback).*

Figure 8 depicts the application of these modeling rules for representing the "Modify Professor" functional process based on the requirements of the system [24].

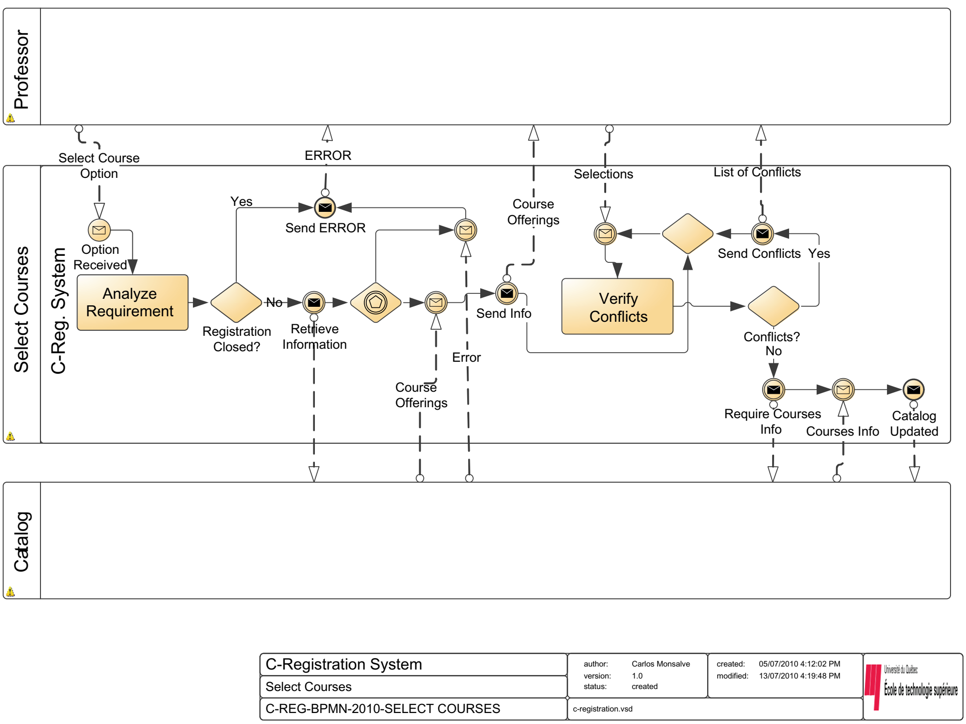

Fig. 7. BPMN model of the "Select Courses" functional process.

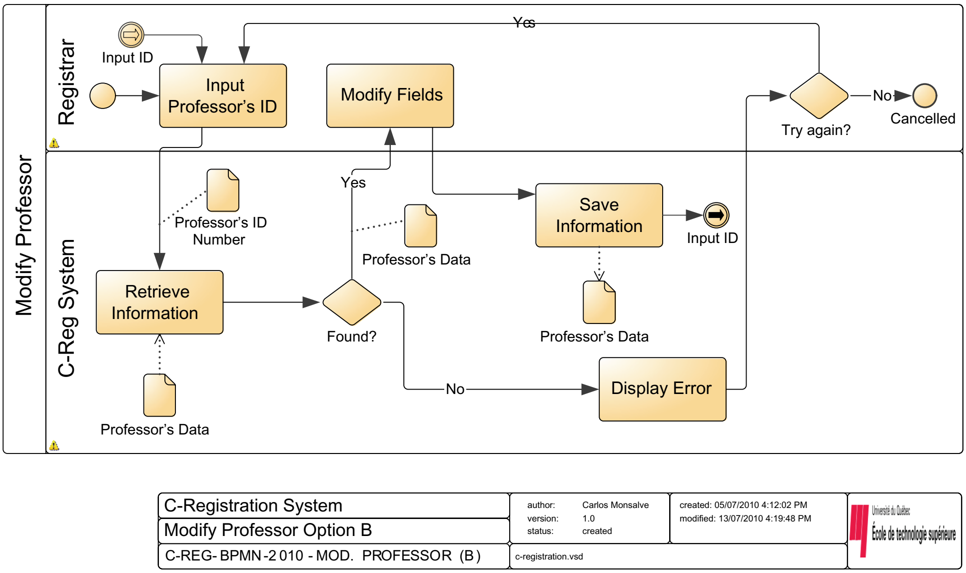

Fig. 8. BPMN model of the "Modify Professor" functional process.

Looking at Figs. 7 and 8, it is possible to conclude that the COSMIC conditions [3] for the recommended **level of granularity** seem to be satisfied with the level of detail of the BPMN models.

5.3. *Mapping and measuring based on BPMN*

According to COSMIC [3], "a data group is a distinct, non empty, non ordered and non redundant set of data attributes" that describes an "object of interest", the latter being "anything that is identified from the point of view of the functional user requirements." A data group may be represented in BPMN by means of the messages exchanged between pools. For example, observe the representation of the "Select Courses" functional process in Fig. 7: the last message sent to the catalog pool by the C-Registration System includes the data group "Catalog". Also, a data group may be represented in BPMN by the information describing a data object that represents a persistent storage. For example, the first data object in the representation of the "Modify Professor" functional process (Fig. 8) shows the "Professor's ID number" data group.

Before measuring the functional size of the C-Registration System, the rules for mapping between the COSMIC concepts [3] and the modeling constructs of BPMN 1.2 [11] must be defined. From the analysis in Figs. 7 and 8, some of these rules can be derived. Table 3 shows all the rules that have been defined based on that analysis, as well as a comparison of the definitions of the COSMIC concepts and the BPMN constructs.

Table 3. Mapping between COSMIC and BPMN version 1.2.

COSMIC FSM method V.3.0.1	BPMN 1.2	Comments
Functional User	Lane and pool	Those that interact with the lane of the software to be measured
Boundary	The lane that represents the software to be measured	
Functional Process	Pool	Those that contain the lane that represents the software to be measured
Triggering Event	Start Event	
Data Group	Name of a message	Between pools
	Data Object	When a persistent storage must be accessed
Entry	An incoming message or sequence flow	
Exit	An outgoing message or sequence flow	
Read	An upstream association with a data object	
Write	A downstream association with a data object	

Figure 9 shows an example of how to apply the mapping rules to representing the "Close Registration" functional process. Owing to space limitations, only this example is provided here. The measurement results are obtained by simply adding the various data movements (Entries (E), Exits (X), Writes (W), and Reads (R)) that appear in the BPMN models representing the various processes.

Table 4 shows the measurement results obtained by this research compared with those obtained in step 1 of the COSMIC case study [24] and with those obtained using the Qualigram notation [9]. A discussion of these results is presented in Sec. 7.

5.4. *Deriving notation-independent modeling guidelines and mapping rules*

Based on the analytical comparison of the results obtained in Secs. 5.1–5.3, a set of notation-independent modeling guidelines for FSM is derived first, and then a general set of mapping rules is proposed. Both the modeling guidelines and the mapping rules are intended for use in the business application software domain.

It is critical when performing FSM based on BP models to choose the correct level of abstraction for modeling the FURs of the software to be measured. Doing so will ensure that the specifications will be represented with the right level of granularity. For example, it would be extremely difficult to obtain all the required information for FSM from a high-level BP model, like the one depicted in Fig. 5, with the Qualigram notation. From the BP models presented in Secs. 5.1 and 5.2,

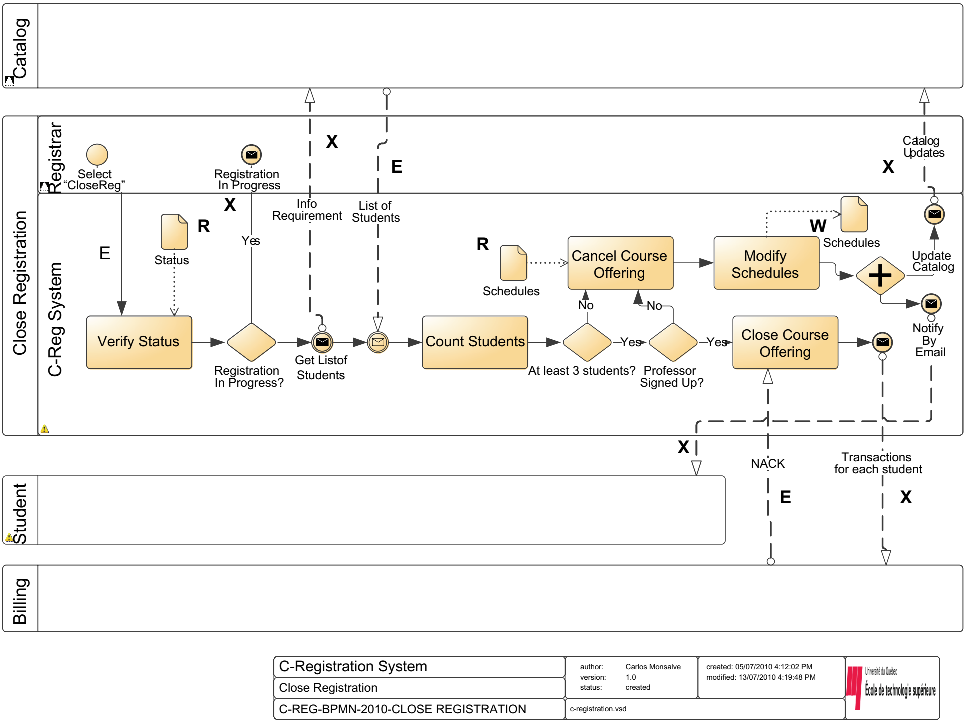

Fig. 9. Application of the mapping rules to the “Close Registration” functional process.

it is possible to conclude that a good level of granularity is achieved when modeling at what Qualigram calls the “organizational level” of abstraction.

Guideline 1. *If the selected BP modeling notation offers various modeling levels of abstraction, choose one that allows depiction of the BP workflow, including its activities, roles, events, and flow of information.*

Modeling rules Q.BA2, Q.BA6, and BPMN3 are related to the same concepts and can be generalized as follows:

Guideline 2. *Consider any peer software component that interacts with the measured software as an external participant (i.e. external role).*

Modeling rules Q.BA3 and BPMN1 are very similar, and can be generalized as follows:

Guideline 3. *Represent any logical instruction set that is worth detailing as a separate BP workflow.*

Modeling rules Q.BA4 and BPMN4 share some concepts, and can be generalized as follows:

Guideline 4. *Represent any user of the software, external to the organization, as an external participant (i.e. external role).*

Table 4. Comparison of the measurement results: C-Registration System.

No.	COSMIC case study step 1						Measurement results based on Qualigram						Measurement results based on BPMN v.1.2					
	Functional processes	Data movements				CFP	Procedures	Data movements				CFP	Processes	Data movements				CFP
		E	X	R	W			E	X	R	W			E	X	R	W	
1	Logon	1	1	1		3	Login	1	2	1		4	Login	1	2	1		4
2	Add a professor	1	2	1	1	5	Add Professor	1	2	1	1	5	Add Professor	1	2	1	1	5
3	Modify a professor	2	2	1	1	6	Modify Professor	2	2	1	1	6	Modify Professor	2	2	1	1	6
4	Delete a professor	2	2	1	1	6	Delete Professor	3	3	1	1	8	Delete Professor	3	3	1	1	8
5	Select Courses to Teach	4	5			9	Select Courses to Teach	4	6			10	Select Courses to Teach	4	6			10
6	Add a student	1	1	1	1	4	Add Student	1	1	1	1	4	Add Student	1	1	1	1	4
7	Modify a student	2	2	1	1	6	Modify Student	2	2	1	1	6	Modify Student	2	2	1	1	6
8	Delete a student	2	2	1	1	6	Delete Student	3	3	1	1	8	Delete Student	3	3	1	1	8
9	Create a schedule	5	5	1	2	13	Create Schedule	4	5	2	1	12	Create Schedule	4	5	2	1	12
10	Modify a schedule	5	6	2	2	15	Modify Schedule	4	6	3	1	14	Modify Schedule	4	6	3	1	14
11	Delete a schedule	2	3	1	1	7	Delete Schedule	3	4	2	1	10	Delete Schedule	3	4	2	1	10
12	Close Registration	2	5	1	1	9	Close Registration	3	5	2	1	11	Close Registration	3	5	2	1	11
13	Submit Grades	4	5	2	1	12	Submit Grades	4	4	2	1	11	Submit Grades	4	4	2	1	11
14	View Report Card	1	3	2		6	View Report Card	1	2	2		5	View Report Card	1	2	2		5
	Total	34	44	16	13	107	Total	36	47	20	11	114	Total	36	47	20	11	114

In addition, modeling rule BPMN4 includes some relevant considerations that can be generalized as follows:
Guideline 5. *Represent any user of the software, internal to the organization, as an internal participant (i.e. internal role).*

Modeling rules Q.BA5 and BPMN2 can be generalized as follows:
Guideline 6. *Represent the measured software as an internal participant (i.e. internal role).*

Modeling rules Q.BA7 and BPMN7 present some concepts in common and can be generalized as follows:
Guideline 7. *Anytime relevant data must be retrieved from or written to persistent storage, represent that type of action as a resource or as a data object used in the BP. Associate the resource or data object with the corresponding modeling construct, and then differentiate a retrieval action from a writing action in an appropriate way.*

Modeling rule Q.BA8 is relevant and should be generalized:
Guideline 8. *If the BP being modeled requires, at its inception, that information be entered by the user triggering it, represent the initial submission of information as the triggering event.*

Modeling rules Q.BA9 and BPMN5 are very similar, and can be generalized as follows:
Guideline 9. *All the error conditions identified by the internal participant (i.e. internal role) representing the measured software must be collected by a single modeling construct associated with the same internal participant, before reporting those conditions to another participant (i.e. role).*

Modeling rules Q.BA10, BPMN6, and BPMN8 share common concepts, and can be generalized as follows:
Guideline 10. *Avoid representing flows of information between participants (i.e. roles), whether they are internal or external, when those flows are only aimed at indicating a possible end to the workflow, or a repetition of it.*

Comparing Tables 2 and 3, it is possible to generalize the mapping rules for the business application software domain, as presented in Table 5.

The COSMIC data group concept presents two mapping options, as described in Table 5. The first option is to map a data group to the information provided as part of a flow. This option is valid for the data groups that are exchanged between the measured software and the functional users. The second option is to map a data group to the name of a resource or data object. This option is valid for the data groups that are retrieved from, or moved to, a persistent storage by the measured software.

Table 5. Rules for mapping between COSMIC and BP modeling notation.

COSMIC FSM method V.3.0.1	BP modeling notation
Functional User	Construct that represents a role or participant
Boundary	The swim-lane of the measured software
Functional Process	Business process
Triggering Event	Start Event
Data Group	Information provided as part of a flow
	Name of a resource or data object
Entry	An incoming flow
Exit	An outgoing flow
Read	A resource or data object representing the retrieval of data
Write	A resource or data object representing the writing of data

6. FSM based on a BP Model: The Real-Time Domain

The **purpose** of this section is to measure the functional size of the software components of the Rice Cooker Controller [34] based on a set of BP models, in order to analyze the feasibility of using them as the source of information for the FSM of real-time software. Therefore, the **scope** of this measurement is given by all the software requirements of the Rice Cooker Controller case study of the COSMIC Group, which is a real-time system. All its software components are at the same hierarchical level, and at a similar **level of decomposition**. Therefore, in this article, we consider that all the software components of the Rice Cooker Controller belong to a single software **layer**. In the next subsection, the specific modeling rules for producing Qualigram models of real-time software for FSM purposes are presented. In addition, the appropriate level of abstraction of the models generated is determined, in accordance with the **level of granularity** expected by the COSMIC FSM method. In Sec. 6.2, the mapping rules between COSMIC and Qualigram for the real-time domain are defined, in order to arrive at a measure of the functional size of the software components of the Rice Cooker Controller.

6.1. *Modeling rules for the real-time domain*

The modeling rules presented in Sec. 5.1 for the business application domain can be adapted as follows:

Modeling Rule Q.RT1. *Represent the various software components of the real-time system as one process at the top level of abstraction* (*i.e. the strategic level*).
Modeling Rule Q.RT2. *Consider any hardware interacting with the software as an external entity.*
Modeling Rule Q.RT3. *Consider as a procedure any software requirement that*: (1) *presents an autonomous functionality* (*i.e. does not depend on other software components*); *and* (2) *can be detailed more deeply.*

Figure 10 shows the top-level model of the software components of the Rice Cooker Controller.

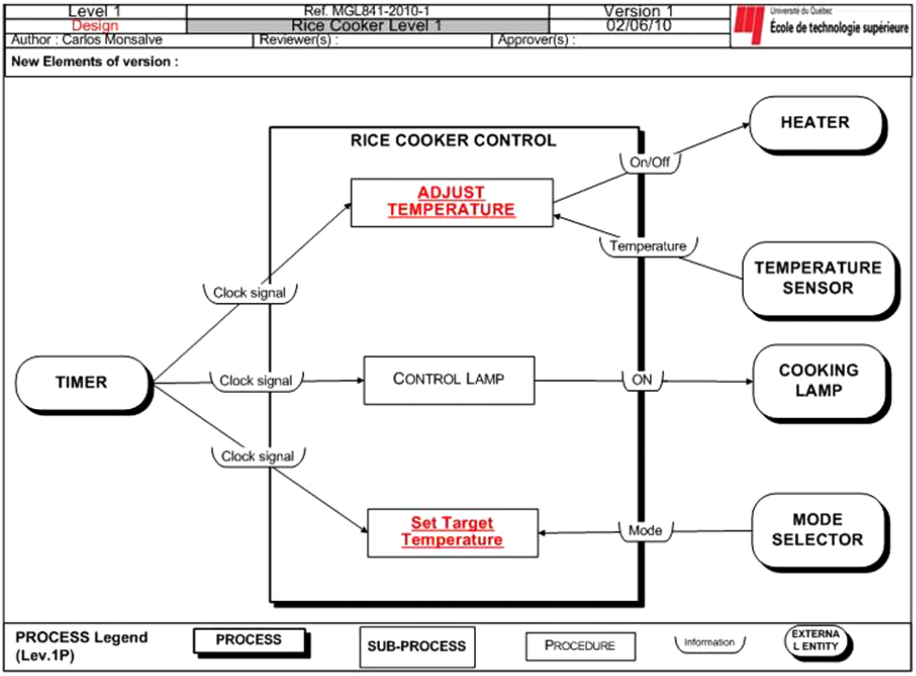

Fig. 10. Top-level Qualigram model of the Rice Cooker Controller.

Modeling Rule Q.RT4. *Represent the software being measured as an internal role at the intermediate level of abstraction (i.e. the organizational level).*
Modeling Rule Q.RT5. *As any instruction requiring that persistent data be retrieved or written should be associated with a material tool, label every material tool, indicating the type of operation to be applied: R for retrieve, W for write.*
Modeling Rule Q.RT6. *Collect all the error conditions identified by a role by means of a unique instruction executed by the same role before reporting them to another role.*
Modeling Rule Q.RT7. *Avoid representing flows of information between roles when those flows are only aimed at indicating a possible end to the workflow.*

Based on the specifications of the system [17] and the annotated modeling rules, an intermediate-level model for each procedure depicted in Fig. 10 has been produced. Owing to space limitations, this paper only presents the model for the "Adjust Temperature" procedure (Fig. 11). Since every user of the software components modeled is individually represented at the intermediate level as a role, and every procedure responds to a single triggering event, we can conclude that the appropriate **level of granularity** seems to be satisfied with the intermediate level (i.e. organizational level) of the Qualigram notation. We will not, therefore, look into the analysis of the bottom level of abstraction (i.e. the operational level) in this research.

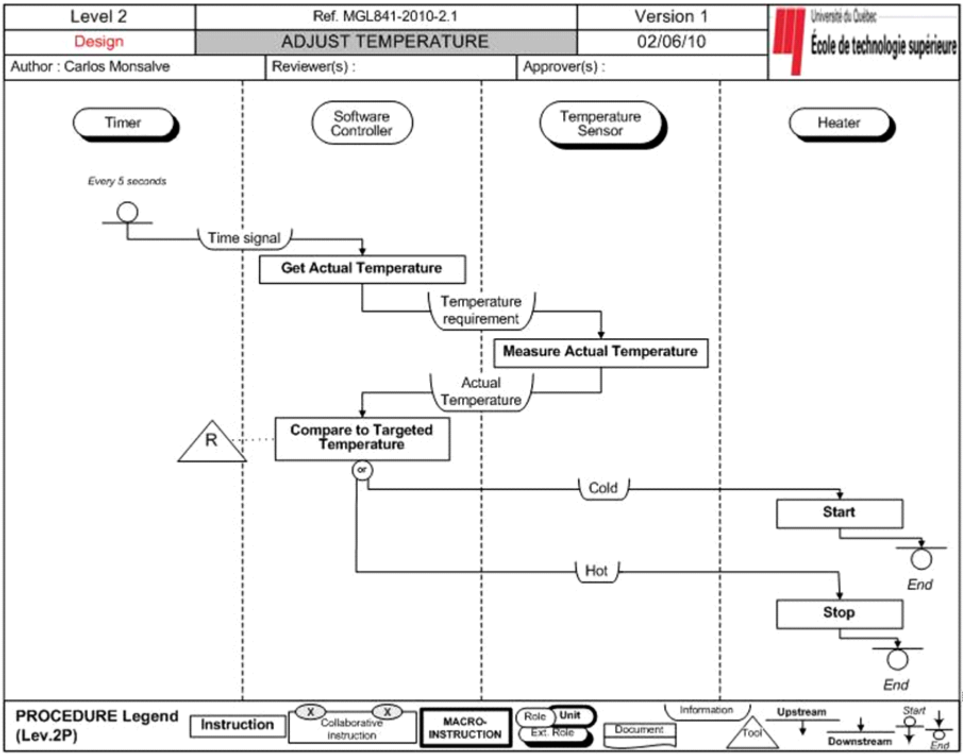

Fig. 11. Qualigram model of the “Adjust Temperature” procedure.

6.2. *Mapping and measuring*

After analyzing the models obtained for the Rice Cooker Controller and comparing the definitions of the COSMIC concepts with those of the Qualigram modeling constructs, we can conclude that the rules defined in Table 2 also apply to the real-time software domain. Figure 12 shows an example of how to apply the mapping rules to the “Set Target Temperature” procedure. Owing to space limitations, only this example is provided here. The measurement results are obtained by simply adding the data movements (Entries (E), Exits (X), Writes (W), and Reads (R)) that appear in the models representing the various procedures. The results are then compared with those obtained in [17]. Table 6 shows this comparison. A discussion of the results is presented in Sec. 7.

7. Discussion of Results

7.1. *Business application domain*

Table 4 shows that the measurement results obtained based on Qualigram are the same as those obtained using BPMN. This result supports the generalization of the

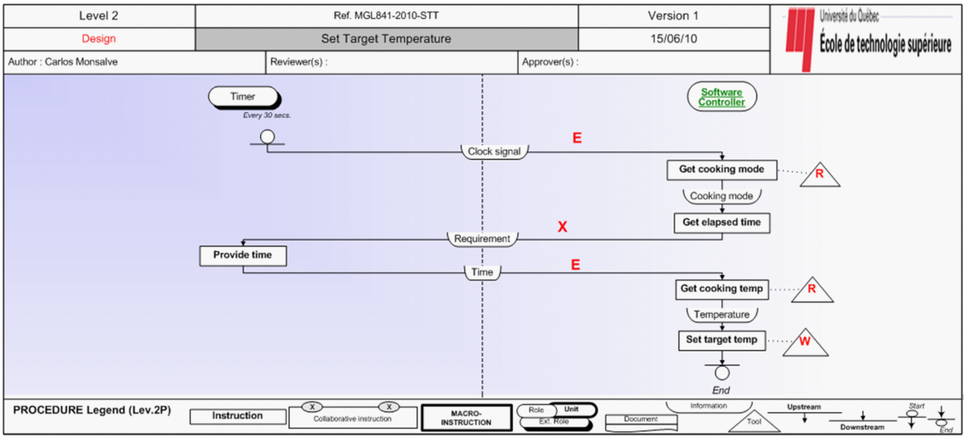

Fig. 12. Applying the mapping rules to the "Set Target Temperature" procedure.

Table 6. Measurement results: Real-time domain.

Results obtained by Lavazza and Del Bianco [17]						Measurement results based on Qualigram					
Process	Data Movements				CFP	Procedures	Data Movements				CFP
	E	X	R	W			E	X	R	W	
Tick (control lamp)	1	1			2	Control Lamp	1	1			2
5 sec. Signal management (control heater)	2	1	1		4	Adjust Temperature	2	3	1	0	6
30 sec. Signal management (set target temperature)	2		2	1	5	Set Target Temp.	2	1	2	1	6
Total	5	2	3	1	11	Total	5	5	3	1	14

approach proposed in this research. The results of using Qualigram for FSM purposes in the business application domain were discussed in [9], and are summarized as follows: (1) there are differences because some modeled information should not be considered as a data group; and (2) there are differences because some modeled details are not considered by the case study for measurement purposes. The inclusion and analysis of the data groups as part of the flows of information between roles were shown to be critical for identifying the Entries (E) and Exits (X) to be measured. The BP models of some of the functional processes required representing flows of information which, according to the mapping rules (Table 2), were considered as Exits; however, according to the COSMIC measurement rules, they could not be considered as such. To address this difference, the flows of information should include the data groups, and it must be determined during the measurement process whether or not each of the information flows corresponds to a data group. Other measurement differences were related to details of the functional procedures that were required to be represented as part of the BP models, even though they were not considered in the interpretation of the specifications in the case study. Most of the rest of this subsection discusses the results based on BPMN 1.2 obtained in Sec. 5.3.

The inclusion and analysis of the data groups as part of the messages between pools, or of the sequence flows between lanes, may be critical for identifying the Entries and Exits to be measured. Consider the "Delete Student" functional process: According to the C-Registration System specifications, the registration software has to send a confirmation requirement after receiving a requirement from the registrar to delete a professor's record, and the registrar must confirm that deletion requirement to the registration software. This deletion handshake is represented in the BPMN model of the "Delete Student" functional process (see Fig. 13). According to the mapping rules, an outgoing sequence flow is considered as an Exit, and an incoming sequence flow is considered as an Entry (see Table 3). However, according to the COSMIC measurement rules, this kind of deletion handshake cannot be considered as a source of data movements. Consequently, there is a difference of one Exit and one Entry between the results of the reference case study [24] and

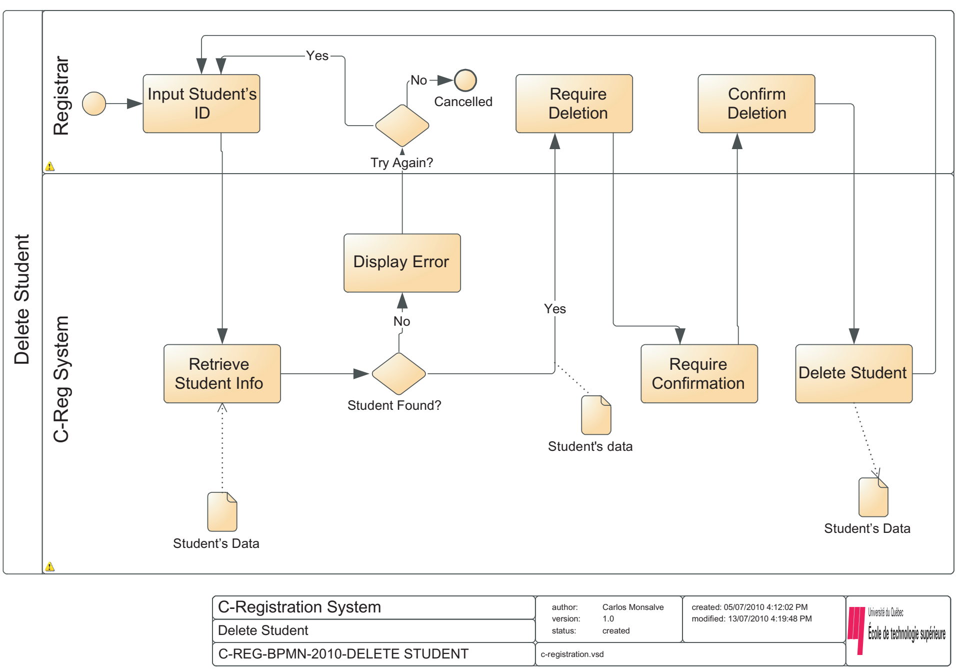

Fig. 13. BPMN model of the "Delete Student" functional process.

those obtained in this paper (see Table 4). To address this difference, the messages and sequence flows should include the data groups, and it must be determined during the measurement process whether or not each of the messages or sequence flows corresponds to a data group. Something similar happens with the "Delete Professor", "Delete Schedule", and "Login" functional processes.

The difference of one Exit (Table 4) for the "Select Courses" functional process (see Fig. 7) is caused by the fact that the reference case study apparently considers that the course offering information is updated in the Catalog System every time this system is consulted about potential conflicts between the offerings selected by the professor. In this research, these two functions have been disaggregated, because the course offerings should be updated only after the professor has resolved the conflicts.

There is a difference of one Read for the "Create Schedule", "Modify Schedule", and "Delete Schedule" functional processes. The reference case study does not consider the FURs associated with verifying the status of the registration process (closed or not closed) before meeting the student's requirement. The reason given by the case study is the poor quality of the specifications. Even though this may be true, we have considered the required verification in this research, because it has been modeled as one of the tasks to be executed for these functional processes. Something similar happens with a verification FUR for the "Close Registration" functional process (see Fig. 9). In addition, for the "Create Schedule" and "Modify Schedule" functional processes, this research has considered that the only way a student can save a schedule is when he or she submits a set of courses to the registration system. Therefore, an extra Entry and an extra Write have not been considered to be a consequence of a "Save Schedule" FUR. It has, however, in the reference case study.

The specifications mention that during the "Close Registration" functional process (see Fig. 9), it is possible that the billing system will not respond to the requirements of the registration system. If that is the case, the specifications ask that the requirement be retried an undetermined number of times. The reference case study has not considered this as a functionality to be measured, probably because there is no data group associated with it. However, in this research, it has been measured as an Entry, because the registration system needs to receive a message from the billing system in order to retry the requirement.

Finally, the impact of the data groups is again evident in the measurement difference that appears for the "Submit Grades" functional process. After retrieving the list of students and retrieving the grades (two different data groups), the specifications ask for a display of those grades. In the BPMN model, this is represented by only one task, which displays the names of the students and their grades, and it counts as one Exit. However, the reference case study considers two Exits, because there are two different data groups. Something very similar happens with the "View Report Card" functional process.

7.2. *Real-time domain*

It is very likely that a BP modeling notation would not be used for modeling real-time software, as the actual purpose of this kind of modeling notation is to represent organizational BPs. However, it is possible to conclude from the results of this research that, following the correct modeling rules, clear and useful models representing real-time software components can be produced.

The first difference between the results obtained in this research and those obtained in [17] (see Table 5) is caused by the way in which the two possible signals to be sent to the heater are represented. Both options are represented as independent flows of information in the Qualigram model (see Fig. 11), and are therefore considered as two Exits. However, in [17], they are considered as part of the same Exit.

Finally, the second difference appears in the "Set Target Temperature" procedure (see Fig. 12). The reference case study [17] considers that the elapsed time is spontaneously sent by the timer to the software controller. In the Qualigram model, the timer sends the time elapsed after receiving a requirement from the software controller. It is therefore considered as an additional Exit in this article.

8. Conclusions, Limitations, and Future Work

This research has shown the technical feasibility of using BP models for FSM with the COSMIC measurement method (COSMIC FSM). A set of modeling rules to represent the software components to be measured using the BPMN 1.2 modeling notation has been defined for the business application domain. The modeling rules for representing real-time software in Qualigram modeling notation have also been defined. In addition, the rules for mapping between the COSMIC concepts and both the Qualigram and BPMN 1.2 modeling constructs have been derived. The modeling rules and mapping rules have been applied to two case studies, one for the business application domain and the other for the real-time domain. The results have been compared with those obtained in previous works for the same case studies.

The modeling rules for the business application domain have been generalized, producing a set of notation-independent BP modeling guidelines for FSM purposes. However, the strengths of these guidelines should be further tested with other popular BP modeling notations. Moreover, to increase the validity of the guidelines, they should be tested with other case studies rather than only testing with the C-Registration System.

The measurement results show that, following the modeling rules and using the mapping rules, BP models might be used successfully for FSM in both domains. Moreover, there is evidence that the measurement results are not affected by the BP modeling notation selected. However, the strength of these results should be further tested with other case studies; preferably case studies where it is possible to cover business processes that are typically modeled in the industry. The results obtained using the mapping rules may be compared to the results obtained by expert COSMIC measurers, if that is the case.

In the business application domain, a small additional effort is foreseen for modeling the BP for FSM purposes when using BPMN. The BPMN models generated in this research do not present important differences with those that are typically generated in industry. A different scenario is foreseen when using Qualigram. The Qualigram intermediate-level models (i.e. procedure models) generated in this research have required representing the software being measured as another role. Organizations using Qualigram typically do not represent any information system as a role in their intermediate-level models. Therefore, using the proposed modeling rules for Qualigram probably earthier will require a change in the modeling paradigm of organizations (for new BP models), or a rework of the intermediate-level models (for already existent BP models). These conclusions are some of the issues that should be tested in the future case studies.

The proposed approach might be very useful at the early stages of a software project; therefore, it needs to be tested against other case studies that are based on high-level specifications typically used at an early stage of the software development process. It will also be necessary to perform more case studies, in order to: (1) validate the generalization of the modeling rules and mapping rules for the business application domain; (2) study the additional effort required for the modeler and the organization for applying the modeling rules; (3) evaluate the stability of using BP models, which typically change in response to the dynamics of the organization, for FSM as a means for estimating effort; and (4) analyze the advantages and disadvantages of using FSM results as a vehicle to estimate effort based on business processes.

References

1. ISO/IEC, 14143-1:1998, Information technology — software measurement — functional size measurement, Part 1: Definition of concepts, Switzerland (International Organization for Standardization, 2000).
2. A. Abran, *Software Metrics and Software Metrology* (John Wiley & Sons Interscience and IEEE-CS Press, 2010).
3. COSMIC, The COSMIC functional size measurement method version 3.0.1, measurement manual (the COSMIC implementation guide for ISO/IEC 19761: 2003). (The Common Software Measurement International Consortium (COSMIC), 2009).
4. ISO/IEC, 19761:2003 software engineering — COSMIC-FFP — a functional size measurement method, Switzerland (International Organization for Standardization, 2003).
5. H. C. Mayr, C. Kop and D. Esberger, Business process modeling and requirements modeling, in *Proc. First International Conference on the Digital Society* (*ICDS '07*), Guadeloupe, 2007, pp. 8–8.
6. M. Indulska, P. Green, J. Recker and M. Rosemann, Business process modeling: Perceived benefits, in *Proc. 28th International Conference on Conceptual Modeling* (ER 2009), Gramado, Brazil, 2009, pp. 458–471.
7. M. Dumas, W. van der Aalst and A. Ter Hofstede, *Process-aware Information Systems: Bridging People and Software Through Process Technology* (Wiley-Interscience, Hoboken, 2005).

8. IIBA, *A Guide to the Business Analysis Body of Knowledge* (*BABOK guide*) (International Institute of Business Analysis (IIBA), Toronto, 2009).
9. C. Monsalve, A. Abran and A. April, Functional size measurement with business process models: The business application domain, in *Proc. International Conferences on Software Measurement* (*IWSM/MetriKon/Mensura 2010*), Stuttgart, Germany, 2010, pp. 270–290.
10. M. Kaya, E-cosmic: A business process model based functional size estimation approach in Department of Information Systems, Ankara (Middle East Technical Universtiy, 2010).
11. OMG, OMG business process model and notation (BPMN), version 1.2 (Object Management Group, 2009).
12. M. Daneva, Measuring reuse of SAP requirements: A model-based approach, in *Proceedings of the 1999 Symposium on Software reusability*, Los Angeles, California, 1999.
13. IFPUG, Function point counting manual, release 4.0, Westerville, Ohio (International Function Point Users Group (IFPUG), 1994).
14. A. Scheer, O. Thomas and O. Adam, Process modeling using event-driven process chains, in Process-aware information systems: Bridging people and software through process technology (Wiley-Interscience, 2005), pp. 119–145.
15. O. Demirors and C. Gencel, A comparison of size estimation techniques applied early in the life cycle, in *Proc. 11th European Software Process Improvement Conference,* (*EuroSPI 2004*), Trondheim, Norway, 2004, pp. 184–194.
16. B. Marín, G. Giachetti and O. Pastor, Measurement of functional size in conceptual models: A survey of measurement procedures based on COSMIC, in *Proc. IWSM/Metrikon/Mensura '08: International Conferences on Software Process and Product Measurement*, Munich, Germany, 2008, pp. 170–183.
17. L. Lavazza and V. Bianco, A case study in COSMIC functional size measurement: The rice cooker revisited, in *Proc. IWSM '09 /Mensura '09: International Conferences on Software Process and Product Measurement*, Amsterdam, The Netherlands, 2009, pp. 101–121.
18. OMG, OMG unified modeling language (OMG UML), superstructure, version 2.3, (Object Management Group, 2010).
19. A. Sellami and H. Ben-Abdallah, Functional size of use case diagrams: A fine-grain measurement, in *Proc. ICSEA '09: 2009 Fourth International Conference on Software Engineering Advances*, 2009, pp. 282–288.
20. K. G. van den Berg, T. Dekkers and R. Oudshoorn, Functional size measurement applied to UML-based user requirements, in *Proc. Software Measurement European Forum* (*SMEF 2005*), Rome, 2005, pp. 69–80.
21. C. Berger and S. Guillard, *La rédaction graphique des procédures*: *Démarche et techniques de description des processus* (Association Française de Normalisation, AFNOR, Paris, 2000).
22. P. Dumas and G. Charbonnel, *La méthode ossad: Pour maîtriser les technologies de l'information* (Les Editions D'Organisation, 1990).
23. V. D. Antonellis and B. Zonta, A disciplined approach to office analysis, *IEEE Trans. Softw. Eng.* **16**(8) (1990) 822–828.
24. GELOG-ETS, Software functional size with ISO 19761:2003 COSMIC-FFP measurement method, proposed measurement etalon: C-registration system, Canada (Software Engineering Research Laboratory, École de Technologie Supérieure, 2008).
25. W. Sedera, G. Gable, M. Rosemann and R. Smyth, A success model for business process modeling: Findings from a multiple case study, in *Proc. Eighth Pacific Asia Conference on Information Systems*, Shanghai, China, 2004.

26. C. Monsalve, A. April and A. Abran, Representing unique stakeholder perspectives in BPM notations, in *Proc. 8th ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2010)*, Montreal, 2010, pp. 42–49.
27. P. Harmon and C. Wolf, The state of business process management 2010, in A BPTrends Report (2010).
28. J. Recker, BPMN modeling–who, where, how and why, Business Process Trends (2008).
29. S. White, Introduction to BPMN, Business Process Trends (2004).
30. H. Smith and P. Fingar, *Business process management: The third wave* (Meghan-Kiffer Press, Tampa, 2007).
31. J. Recker, M. Indulska, M. Rosemann and P. Green, How good is BPMN really? Insights from theory and practice, in *Proc. 14th European Conference on Information Systems* (*ECIS' 06*), Goeteborg, Sweden, 2006.
32. J. Recker, M. Rosemann, M. Indulska and P. Green, Business process modeling: A comparative analysis, *Journal of the Association for Information Systems* **10**(4) (2009) 333–363.
33. M. Zur Muehlen and L. M. Jan Recker, How much language is enough? Theoretical and practical use of the business process modeling notation, in *Proc. CAiSE '08: 20th international conference on Advanced Information Systems Engineering*, Berlin, Heidelberg, 2008, pp. 465–479.
34. COSMIC, *Case study: Rice cooker* (COSMIC group, 2008).
35. OMG, Business process model and notation (BPMN) version 2.0 (Object Management Group (OMG), 2010).
36. B. Silver, *BPMN Method and Style* (Cody-Cassidy Press, Aptos, 2009).