

Functional Size Measurement with Business Process Models: the Business Application Domain

Carlos Monsalve^{1, 2}, Alain Abran², Alain April²

¹CIDIS – FIEC – Escuela Superior Politécnica del Litoral

²GELOG – École de Technologie Supérieure

monsalve@espol.edu.ec, {[alain.abran](mailto:alain.abran@etsmtl.ca), [alain.april](mailto:alain.april@etsmtl.ca)}@etsmtl.ca

Abstract:

Functional size measurement (FSM) is an important component of a software project, as it provides information for estimating the effort required to develop the measured software. For planning purposes, FSM should be performed during the early stages of the project. Considering that a common use of business process models is to gather requirements from the early stages of the project, this raises the opportunity that a business process model could be a valuable source of information for FSM.

This paper analyzes the feasibility of such an approach for the business application domain, including the derivation of the modeling rules for measurement purposes and the rules to map the modeling constructs of a business process modeling language into the COSMIC FSM method concepts. The results are compared with those of a previous FSM case study.

Keywords

Functional size measurement, FSM, COSMIC, ISO 19761, business process model

1 Introduction

Measuring the functional size of software is very important when planning, developing, buying, improving, or maintaining software systems [1]. For instance, it provides valuable information for estimating the effort required to develop the measured requirements of the software to be developed. Based on that estimation, software managers can plan resources and estimate costs for the software project [4]. Functional size measurement (FSM) can be performed based on the project specifications or a posteriori (i.e. based on the finished product). The former is desirable for planning a software project successfully; therefore, the measurement of functional size should ideally be performed during the early stages of the project. There are several proposed methods for FSM, one of which is the COSMIC FSM method [7]. COSMIC has been 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 [7], and it has been reported to be easy to learn and use. COSMIC has been accepted since

2003 as international standard ISO/IEC 19761:2003 “Software engineering – COSMIC-FFP – A functional size measurement method” [13].

Business process models (BP models) are designed to be useful for documenting, communicating, or improving organizational business processes. BP models are also used by software engineers and business analysts to gather the software and system requirements during the development process [9, 12, 16].

Therefore, a BP model may also be a valuable source of information for FSM. However, no research on the feasibility of applying such an approach could be identified. This paper identifies the candidate rules for mapping the modeling constructs used by typical BP models to the various COSMIC concepts, as well as the modeling rules to take into consideration for functional size measurement in the business application domain. Finally, to verify the value of this proposed approach, this paper compares the results obtained to those of a previous FSM case study.

The structure of this paper is as follows. Section 2 reviews related works. Section 3 describes the methodology used in this research. Section 4 presents the results of the use of a BP model for FSM in the business application domain. Section 5 discusses the results obtained. Finally, section 6 presents a summary of the contributions of this research, its limitations, and future work.

2 Related Works

The use of conceptual models for FSM has been studied and analyzed in the research literature, and Marin *et al.* [15] offer a complete survey of related works, including their own. After the publication of that survey, Lavazza and Del Bianco [14] studied the use of Unified Modeling Language (UML) [3] diagrams (use case, component, and sequence diagrams) for modeling real-time software to be measured using the COSMIC FSM method. Sellami and Ben-Abdallah [18] studied the potential relationships between the measurements obtained from UML use case diagrams and those obtained from other UML diagrams. From all these works, only two have included the use of some kind of BP model: Demirors and Gencel [8] used an extension of the Event Driven-Process Chain (EPC) [17] to model a military application. In this case, the EPC diagrams were used as part of the requirements elicitation methodology. The second work [19], proposes the use of UML activity diagrams as one of the possible options for representing the behavioral aspects of the software being modeled. However, neither of these two latter works presents a rule for mapping between the BP models and the concepts of the measurement methods being studied. Moreover, the emphasis of these two works is not related to the feasibility of using only BP models for FSM.

3 Methodology

The methodology used in this research is presented in Fig. 1. To test the feasibility of the proposed approach, the February 23, 2008, version of the C-Registration System case study is used [11]. The system is modeled using the Qualigram language [6]: Qualigram is a management-oriented BP modeling language which is based on the results of an international research project as part of the European Strategic Program for Research and Development in Information Technology (ESPRIT) [5, 10].

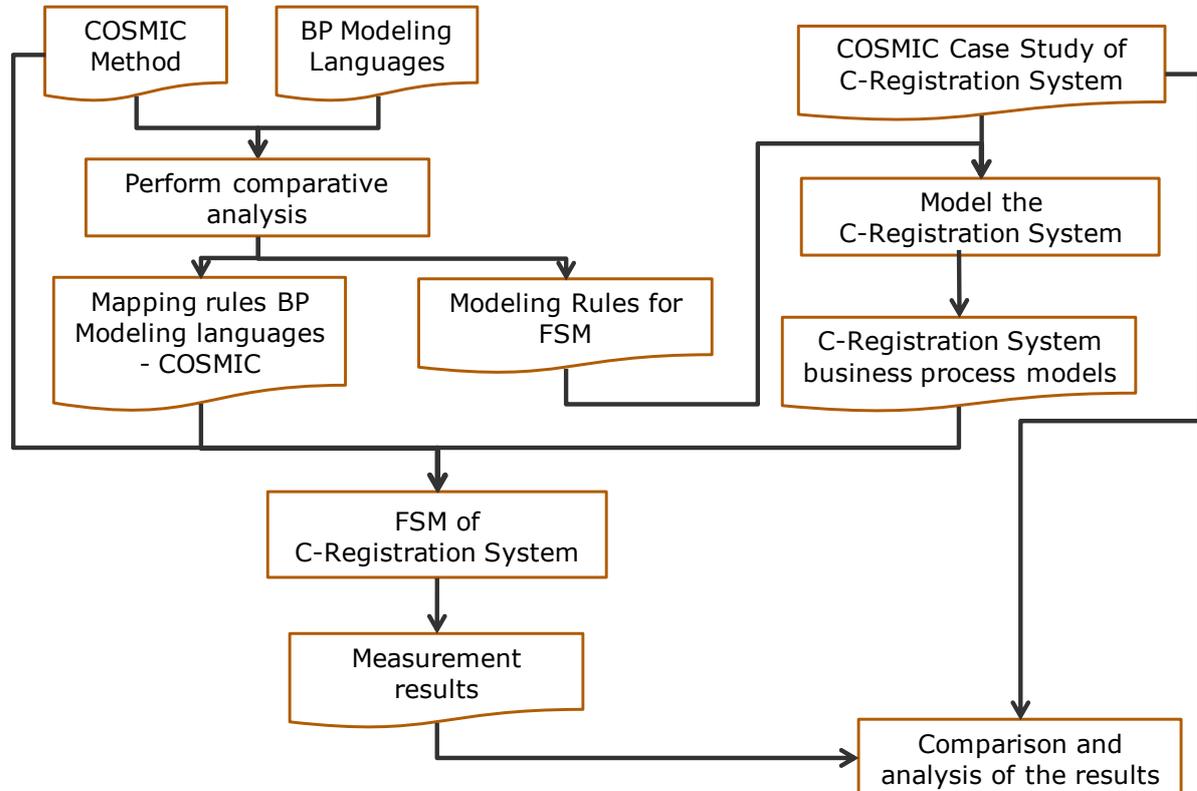


Figure 1: Methodology for the business application domain

Qualigram proposes three levels of abstraction. The top-level (strategic level) models the processes. The intermediate-level (tactical level) models the procedures. Finally, the lowest-level (operational level) models the work instructions. “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”.

It is important to point out that Qualigram’s conception of a process model (i.e. top-level model) is somehow different to the mainstream notion of a business process model. Actually, a Qualigram procedure-model (i.e. intermediate-level model) is closer to what typically is understood as a business process model. In this paper the term BP model is used as a generic term to encompass the different variations and levels of detail that each modeling language or each author may

prefer to use to represent the processes of an organization. Therefore, the reader should not infer that a Qualigram process-model has exactly the same general scope than a BP model. Moreover, because this paper uses the term “BP model” as a generic term, both Qualigram process-models and Qualigram procedure-models are considered as BP models with different levels of abstraction.

Based on the definitions of the various modeling constructs offered by Qualigram, and the definitions of the COSMIC concepts, a mapping table between Qualigram and COSMIC is generated. Also, as a result of the comparison, a set of specific modeling rules is identified to allow the Qualigram models to be used for FSM. The mapping table and the generated BP models are used to measure the functional size of the C-Registration System [11], a case study documented by the COSMIC organization. Finally, the measurement results are compared to those presented in the C-Registration System case study.

4 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 models generated using Qualigram, in order to analyze the feasibility of using BP models as the source of information for FSM.

Therefore, the **scope** of this measurement is given by all the functional user requirements (FUR) of the C-Registration System as described in [11]. The C-Registration System is business application software that belongs to the “**application layer**” of the “typical layered software architecture” [7]. In the next subsection, the modeling rules for producing Qualigram models suitable to be used for FSM are identified. In addition, in order to be in agreement with the **level of granularity** expected by the COSMIC FSM method, the appropriate level of abstraction of these models is determined. In subsection 4.2, the mapping rules between COSMIC and Qualigram are defined to ultimately measure the functional size of the C-Registration System.

4.1 Modeling Rules for Measurement Purposes

At the top level of abstraction (i.e. strategic level), Qualigram [6] models the strategy of the organization, asking the questions: “why” and “where to” (i.e. the main goals of the organization and the relevant external actors). At this level, therefore, Qualigram represents those processes that are directly related to the goals and external actors of the organization. The external actors are either the destination of the products or services produced by the organization, or the important partners whose services or products are required to achieve the goals. It is also possible to represent the relations between the various processes, and between the processes and the external actors.

Modeling Rule BA1: At the top level of abstraction (i.e. strategic level), represent the software to be measured as a process.

Modeling Rule BA2: Following the COSMIC principles, consider any external software component that interacts with the measured software as an external actor.

Qualigram allows some of the processes to be more detailed at the top level of abstraction by representing their sub-processes and main procedures. The procedures constitute the elements that are further detailed at Qualigram’s intermediate level of abstraction (i.e. tactical level). Any procedure represented at that level must present at least one input and one output of information. Qualigram prohibits representation of any internal actor of the organization at the top level of abstraction.

Modeling Rule BA3: Consider any logical instruction-set that is worth detailing in more depth as a procedure.

Modeling Rule BA4: Represent any user of the software who allows representation of the inputs and outputs of the procedures modeled as an external actor.

Fig. 2 shows the top-level model of the C-Registration System, based on the requirements of the system [11] and the annotated modeling rules.

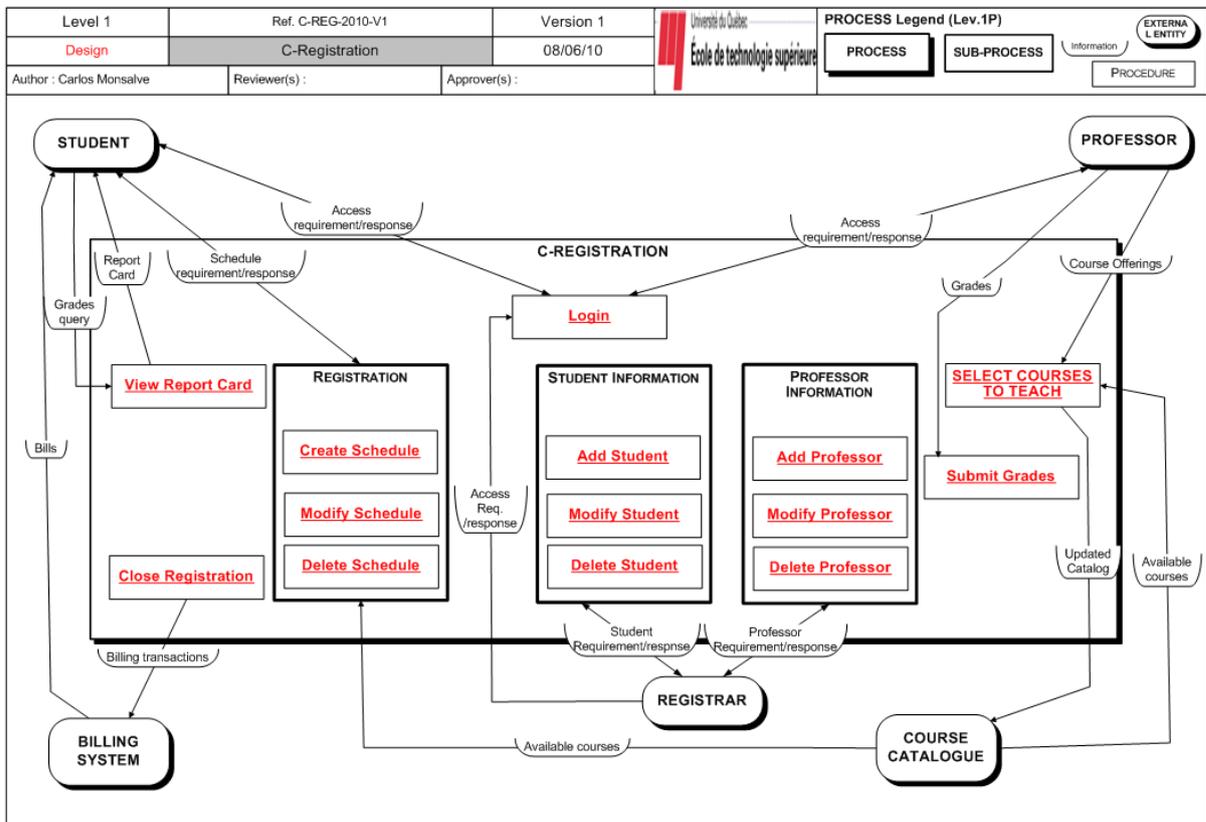


Figure 2: Qualigram’s top-level model of the C-Registration System

The requirements mention the registrar as an actor. From an organizational point of view, the registrar should be considered as an internal actor who would not appear in the top-level model. However, in order to represent the inputs and outputs

of some of the procedures modeled, the registrar is represented as an external actor.

Qualigram uses more specialized terms at its intermediate level of abstraction (i.e. tactical level). The terms “internal role”, “external role”, and “unit” are used to identify specific types of actors. For modeling and analyzing the procedures at this level, the modeler tries to answer the questions “who is doing what” and “what is being done” (i.e. the various instructions to be executed as part of the procedure, and the different actors involved in the procedure).

According to Qualigram rules, a procedure requires a minimum of two actors and five instructions. Moreover, the login procedure, or the “create student/professor” procedure, would not typically be considered as a candidate to be modeled using a BP modeling language. From the organizational point of view, a typical BP crosses different functional departments of the organization, and that is not the case for this type of procedure. For example, the login procedure contains only an interaction between a user and the information system, and the login requirement does not cross any functional department of the organization. Something similar happens with the create, modify, update, or delete types of procedures.

Modeling Rule BA5: At the intermediate level of abstraction (i.e. tactical level), represent the software being measured as an internal role.

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

COSMIC requires identifying those data movements that retrieve or write information from/to a persistent storage. Qualigram allows representation of the tools that are used or produced by an instruction, which could be of the document type or of the material type. A material tool is used to represent any material resource, such as a piece of software, a machine, a software tool, office material, etc.

Modeling Rule 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.

According to Qualigram, “a procedure is never started spontaneously”, and always requires a “triggering element”. This triggering element might be any organizational event (i.e. a customer requirement), or a requirement coming from another procedure. COSMIC defines a triggering event as “an event that causes a functional user of the piece of software to initiate one or more functional processes.” After identifying a triggering event, the functional user typically initiates the functional process sending a message to the software. This message constitutes a triggering Entry, which is considered by COSMIC as a data movement of the Entry type. Where a functional process has to wait for additional data from the

functional user after having undergone the triggering event, only one Entry has to be considered; this is true even in the case where the functional user requires a prompt message after producing the triggering Entry for entering the additional data. Moreover, “in the business application domain, control commands shall be ignored.”

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

COSMIC establishes that “all messages generated and output by software without user data shall be considered” as “a single Exit...within each functional process.”

Modeling Rule 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.

COSMIC determines the Exits and Entries to a functional process by identifying those data movements that cross the boundary of the functional process. The boundary is defined as “a conceptual interface between the software being measured and its functional users.” At Qualigram’s intermediate level of abstraction (i.e. tactical level), each role is confined to a swimlane. If a role A needs to pass the control of the workflow to a role B, then role A needs to send a flow of information to role B crossing the swimlane of role A.

Modeling Rule 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 requirements of the system [11] and the annotated modeling rules, a second-level model for the procedures depicted in Fig. 2 has been produced. Due to space limitations, this paper only presents the models for the Login (Fig. 3) and Add Professor (Fig. 4) procedures.

According to COSMIC, the recommended level of granularity of the FUR is achieved when the functional users: 1) are individuals; and 2) “detect single occurrences of events”. Looking at Figures 3-4, it is possible to conclude that these conditions seem to be satisfied with the intermediate level of Qualigram (i.e. tactical level). Therefore, this research will not look into analyzing the bottom-level of abstraction (i.e. operational level).

4.2 Mapping and Measuring

Before measuring the functional size of the C-Registration System, the mapping rules between the COSMIC concepts [7] and the modeling constructs of Qualigram [6] must be defined. From the analysis in Figures 2-4, it is possible to derive some of these rules. Table 1 shows all the rules that have been defined based on that analysis and a comparison of the definitions of the COSMIC concepts and the Qualigram constructs.

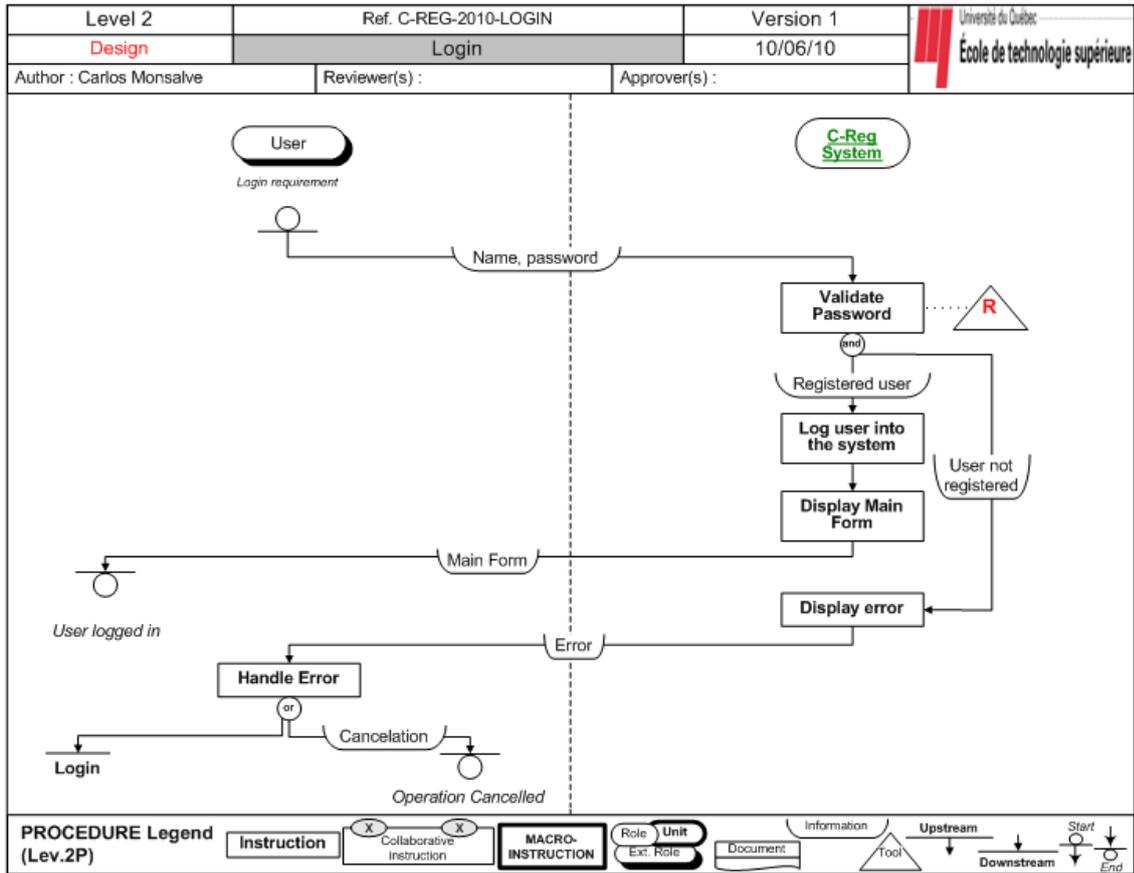


Figure 3: Login procedure

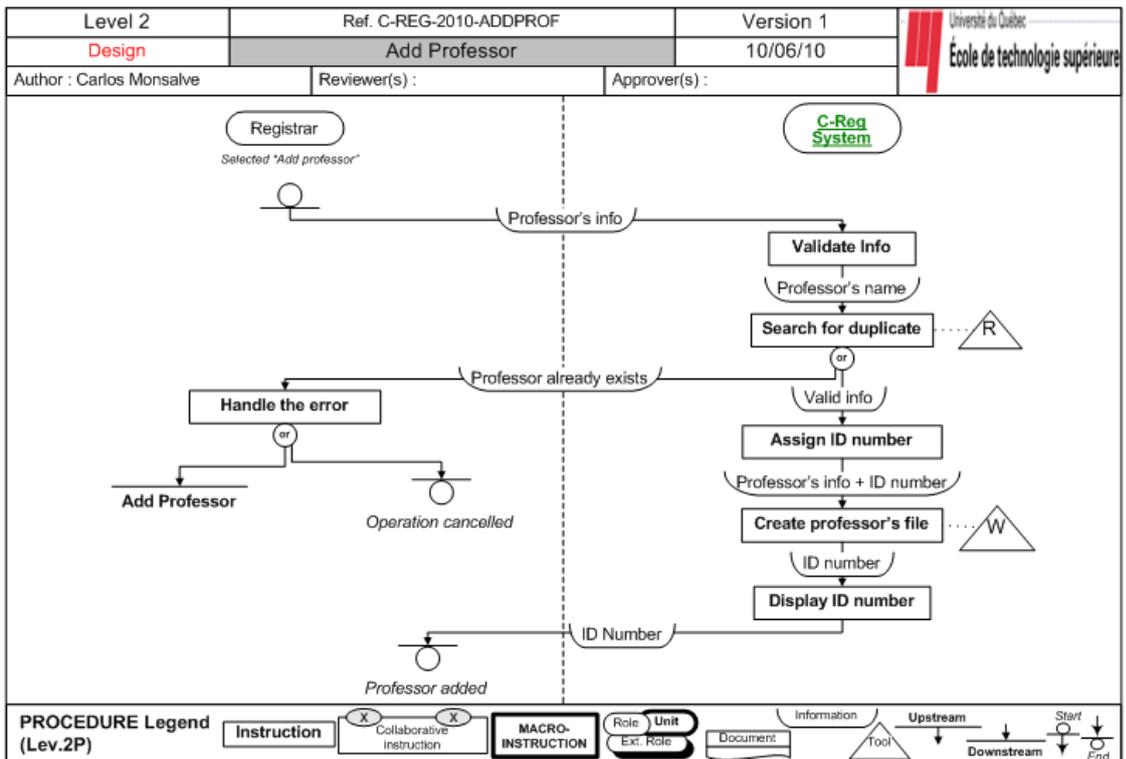


Figure 4: Add Professor procedure

Table 1: Mapping between COSMIC and Qualigram

COSMIC FSM Method V.3.0.1	Qualigram	Comments
Functional User	Role	Only those roles that interact with the software
Boundary	The process box that represents the software	Top-level model
	The swimlane 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	

According to COSMIC, “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 Qualigram by means of the flows of information between roles. For example, observe the “Add Professor” procedure in Fig. 4: the first flow of information from the registrar to the C-Registration System includes the data group “professor’s info”. Also, a data group may be represented as part of the information describing a material tool that represents a persistent storage. For example, to the Login procedure in Fig. 3, it is possible to add the “user’s data” data group to the “R” description of the triangle that represents the persistent storage.

Fig. 5 shows an example of how to apply the mapping rules to the “Delete Schedule” procedure. Owing to space limitations, only this example is provided here. The measurement results are next obtained by simply adding the different data movements (Entries (E), Exits (X), Writes (W), and Reads (R)) that appear in the models representing the various procedures.

The COSMIC case study [11] presents two sets of results: “step 1” and “step 2”. The first set is obtained after applying the COSMIC FSM method to the “functional user requirements exactly as they are written” in the original specifications

of the C-Registration System. The second set results from modifying “by a further assumption” the FURs of step 1. This paper has only considered the FURs as given by step 1 of the case study.

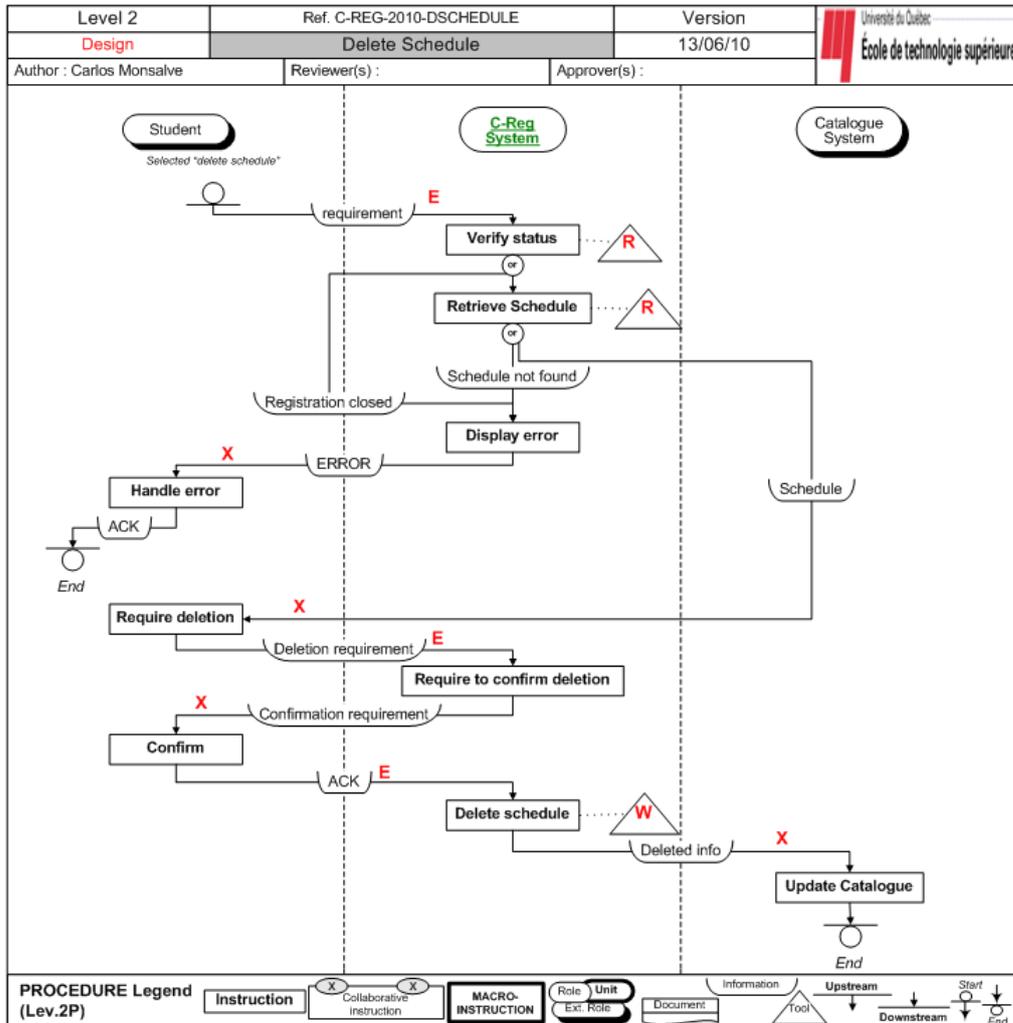


Figure 5: Delete Schedule procedure

Table 2 shows the measurement results obtained by this research compared to those obtained by step 1 of the COSMIC case study. A discussion of these results and the modeling rules is presented in section 5.

5 Discussion of Results

5.1 General Remarks

In the initial work reported here, the modeling rules defined in section 4 embody a slight inconsistency. The same reality (i.e. the measured software) is represented as two different concepts, depending on the level of abstraction. At the top level (i.e. strategic level), the measured software is represented as the main process; however, at the intermediate level (i.e. tactical level), it is represented as an inter-

nal role. Qualigram is a management-oriented language, and does not ask for representation of the information systems supporting the business processes. Therefore, for this research, it has been necessary to provide a modeling rule at each of the two levels of abstraction to represent the software components for which modeling is required for performing FSM.

Table 2: Comparison of measurement results of the C-Registration case study

No.	COSMIC Case Study Step 1					Measurement Results Based on Qualigram						
	Functional Processes	Data Movements				CFP	Procedures	Data Movements				CFP
		E	X	R	W			E	X	R	W	
1	Logon	1	1	1		3	Login	1	2	1		4
2	Add a 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
4	Delete a Professor	2	2	1	1	6	Delete Professor	3	3	1	1	8
5	Select Courses to Teach	4	5			9	Select Courses to Teach	4	6			10
6	Add a 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
8	Delete a Student	2	2	1	1	6	Delete Student	3	3	1	1	8
9	Create a schedule	5	5	1	2	13	Create Schedule	4	5	2	1	12
10	Modify a schedule	5	6	2	2	15	Modify Schedule	4	6	3	1	14
11	Delete a schedule	2	3	1	1	7	Delete Schedule	3	4	2	1	10
12	Close registration	2	5	1	1	9	Close Registration	3	5	2	1	11
13	Submit Grades	4	5	2	1	12	Submit Grades	4	4	2	1	11
14	View Report Card	1	3	2		6	View Report Card	1	2	2		5
	Total	34	44	16	13	107	Total	36	47	20	11	114

5.2 Measurement Results

The inclusion and analysis of the data groups as part of the flows of information between roles may be critical for identifying the Entries (E) and the Exits (X) to be measured. Consider the Login procedure, as depicted in Fig. 3: according to the C-Registration System specifications, the registration software has to send a form at the end of the Login procedure. This requirement has been represented as the information flow that starts after the “Display Main Form” instruction. According to the mapping rules, an outgoing flow of information is considered as an Exit (see Table 1). However, according to the COSMIC measurement rules, a form sent to a user for entering information cannot be considered as an Exit. Consequently, there is a difference of one Exit between the results of the reference case study and those obtained in this paper (see Table 2). 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. Something similar happens with the Delete Professor, Delete Student, and Delete Schedule (see Fig. 5) procedures.

The difference of one Exit (Table 2) for the “select courses to teach” functional process is caused by the fact that the reference case study apparently considers that the course offering information is updated in the Catalogue System every time

this system is consulted about the potential conflicts of 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 one more Read in the procedures (Create, Modify, and Delete) of the “Maintain Schedule” process (see Fig. 2) than in the results of the reference case study. The latter 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 requirements. Even though this may be true, the required verification has been considered in this paper because it has been modeled as one of the instructions to be executed for these procedures. Something similar happens with a verification FUR for the “close registration” functional process. In addition, for the Create and Modify procedures, 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, it has not considered an extra Entry and an extra Write as consequences of a “save schedule” FUR. However, the reference case study has done so.

At the “close registration” functional process stage, the specifications mention that 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 to retry the requirement 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, it has been measured as an Entry here, 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 at the “Submit Grades” procedure stage. After retrieving the list of students and retrieving the grades (two different data groups), the specifications ask for a display of the student’s grades. In the Qualigram model, this is represented by only one instruction, 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 of the two different data groups. Something very similar happens with the “View Report Card” procedure.

6 Conclusions and Future Work

This research effort has looked into the feasibility of using BP models for FSM. A set of modeling rules to represent the software components to be measured using the COSMIC FSM method has been defined for the business application domain. In addition, the rules for mapping between the COSMIC concepts and the Qualigram modeling constructs have been derived. The modeling rules and the mapping

rules have been applied to one case study. The results have been compared to those obtained in a previous work for the same case study. The measurement results show that, following the modeling rules and using the mapping rules, BP models can be used successfully for FSM in the business application domain. Only three of the measurement differences can be attributed to the use of the BP models. In future work, the same case study will be modeled and measured using the Business Process Modeling Notation (BPMN) [2], which has been proposed as a standard for BP modeling by the Object Management Group (OMG). The results will be compared to those obtained using Qualigram to evaluate the impact of the modeling language on the measurement approach. A similar work will be accomplished for the real-time domain to analyze the feasibility of using BP models for FSM in this domain. More case studies need to be reviewed to: 1) generalize the modeling rules and the mapping rules; 2) evaluate the stability of using BP models for FSM; and 3) analyze the advantages and disadvantages of using FSM 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, International Organization for Standardization, Switzerland, 2000.
2. OMG Business Process Model and Notation (BPMN) *Version 1.2*, Object Management Group, 2009.
3. OMG Unified Modeling Language (OMG UML) *Superstructure. Version 2.3*, Object Management Group, 2010.
4. Abran, A. *Software Metrics and Software Metrology*. John Wiley & Sons Interscience and IEEE-CS Press, 2010.
5. Antonellis, V.D. and Zonta, B. A Disciplined Approach to Office Analysis. *IEEE Trans. Softw. Eng.*, 16 (8). 822-828.
6. Berger, C. and Guillard, S. *La rédaction graphique des procédures : démarche et techniques de description des processus*. Association Française de Normalisation, AFNOR, Paris, 2000.
7. 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.
8. Demirors, O. and Gencel, C. A comparison of size estimation techniques applied early in the life cycle. *11th European Software Process Improvement Conference, EuroSPI 2004*, Springer, Trondheim, Norway, 2004, 184-194.

9. Dumas, M., van der Aalst, W. and Ter Hofstede, A. *Process-aware information systems: Bridging people and software through process technology*. Wiley-Interscience, Hoboken, N.J., 2005.
10. Dumas, P. and Charbonnel, G. *La méthode OSSAD: pour maîtriser les technologies de l'information*. Les Editions D'Organisation, 1990.
11. GELOG-ETS. Software Functional Size with ISO 19761:2003 COSMIC-FFP Measurement Method, Proposed Measurement Etalon: C-Registration System, Software Engineering Research Laboratory, Ecole de Technologie Supérieure, Canada, 2008.
12. Indulska, M., Green, P., Recker, J. and Rosemann, M. Business Process Modeling: Perceived Benefits *28th International Conference on Conceptual Modeling - ER 2009*, Springer-Verlag, Gramado, Brazil, 2009, 458-471.
13. ISO/IEC. 19761:2003 Software engineering -- COSMIC-FFP -- A functional size measurement method, International Organization for Standardization, Switzerland, 2003.
14. Lavazza, L. and Bianco, V. A Case Study in COSMIC Functional Size Measurement: The Rice Cooker Revisited. *IWSM '09 /Mensura '09: International Conferences on Software Process and Product Measurement*, Springer-Verlag, Amsterdam, The Netherlands, 2009, 101-121.
15. Marín, B., Giachetti, G. and Pastor, O. Measurement of Functional Size in Conceptual Models: A Survey of Measurement Procedures Based on COSMIC. *IWSM/Metrikon/Mensura '08: International Conferences on Software Process and Product Measurement*, Springer-Verlag, Munich, Germany, 2008, 170-183.
16. Mayr, H.C., Kop, C. and Esberger, D. Business Process Modeling and Requirements Modeling *First International Conference on the Digital Society, 2007. ICDS '07*, IEEE Computer Society, Guadeloupe, 2007, 8-8.
17. Scheer, A., Thomas, O. and Adam, O. Process Modeling Using Event-driven Process Chains. in *Process-aware Information Systems: Bridging People and Software through Process Technology* Wiley-Interscience, Hoboken, New Jersey, 2005, 119-145.
18. Sellami, A. and Ben-Abdallah, H. Functional Size of Use Case Diagrams: A Fine-Grain Measurement. *ICSEA '09: 2009 Fourth International Conference on Software Engineering Advances*, IEEE Computer Society, 2009, 282-288.
19. van den Berg, K.G., Dekkers, T. and Oudshoorn, R. Functional size measurement applied to UML-based user requirements *Software Measurement European Forum SMEF 2005*, Rome, 2005, 69-80.