
Carlos Monsalve\(^1,2\)
1Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Apartado 09-01-5863. Guayaquil, Ecuador
email: monsalve@espol.edu.ec

Alain Abran\(^2\), Alain April\(^2\)
2Department of Software Engineering
École de Technologie Supérieure
1100 rue Notre-Dame Ouest. Montréal, Canada
email: {alain.abran,alain.april}@etsmtl.ca

Abstract—Functional size measurement (FSM) is an important component of a software project. It provides information to estimate the effort required for developing the measured software. For planning purposes, FSM should be performed during the early stages of the project. On the other hand, one common use of business process models is to gather requirements since the early stages of the project. Therefore, a business process model could be a valuable source of information for FSM at an early stage. This article analyzes the feasibility of such an approach for the business application domain, deriving the modeling conditions, 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; early functional size measurement; business process model; measurement case study; requirements modeling; Qualigram;

I. INTRODUCTION

Measuring the functional size of software is very important when developing, buying, improving or maintaining software systems [1]. It provides valuable information to estimate the effort required for developing the measured software. Based on that estimation, software managers are able to successfully plan resources and estimate costs for the software project [2]. Functional size measurement (FSM) can be done a priori (i.e. based on the specifications) or a posteriori (i.e. based on the finished product). The former is desirable for achieving a successful planning of the software project; therefore, ideally the measurement of the functional size should be performed during the early stages of the software project. There are several proposed methods for FSM; among them, the COSMIC FSM method [3]. COSMIC has been designed to be applied to different functional domains: 1) business application software; 2) real-time software; and 3) a combination of the other two domains. It is completely open and available in multiple languages [3] and it claims to be easy to learn and use, including when an a priori FSM is required. COSMIC has been accepted since 2003 as the international standard ISO/IEC 19761:2003 “Software engineering – COSMIC-FFP – A functional size measurement method” [4].

On the other hand, business process models (BP models) are designed to be useful to document, communicate, or improve organizational business processes. BP models are also used by software engineers and business analysts for gathering the software and system requirements since the early stages of the development process [5-7]. Therefore, a BP model may be a valuable source of information for FSM at an early stage of the project. However, no research works studying the feasibility of applying such an approach could be identified. This article identifies the candidate rules to map the modeling constructs used by typical BP models into the different COSMIC concepts, as well as the modeling conditions to be taken into consideration for the business application domain. Finally, to verify the goodness of the proposed approach, this article compares the obtained results to those of a previous FSM case study.

The structure of this article is as follows. Section II describes the methodology used in this research. Section III presents the use of a BP model for FSM in the business application domain. Section IV discusses the obtained results. Section V reviews related works. Finally, section VI concludes with a review of the contributions of this research, its limitations and future work.

II. METHODOLOGY

Fig. 1 depicts the methodology used in this research. To test the feasibility of the proposed approach, the February 23, 2008 version of the C-Registration System case study [8] is used. The system is modeled using the Qualigram language.
[9]: Qualigram is a management-oriented BP modeling language which is based on the results of an extensive international research project as part of the European Strategic Program for Research and Development in Information Technology (ESPRIT) [10], [11].

Qualigram proposes three levels of abstraction. The top level models the business processes. The intermediate level models the procedures. Finally, the lowest level models the work instructions. Based on the definitions of the different modeling constructs offered by Qualigram, and the definitions of the different concepts of COSMIC, a mapping table between Qualigram and COSMIC is elaborated. Also, as a result of the comparison, a set of specific modeling considerations are identified to allow using the BP models for FSM. The mapping table and the generated BP models are used to measure the functional size of the C-registration system. Finally the measurement results are compared to those presented in the C-Registration System case study.

III. 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 that are elaborated using Qualigram language, 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 of the functional user requirements (FUR) of the C-Registration System as described in [8]. The C-Registration System is a business application software that belongs to the “application layer” if the “typical layered software architecture” [3]. In the next subsection the special considerations for producing BP models suitable to be used for FSM are identified. Additionally, in order to be in accordance with the level of granularity expected by the COSMIC FSM method, the appropriate level of abstraction of the generated BP models is determined. In subsection B the mapping rules between COSMIC and Qualigram are defined to finally measure the functional size of the C-Registration System.

A. Modeling Considerations

At the top level Qualigram [9] models the strategy of the organization: the why? and the where? (i.e. the main goals of the organization and the relevant external actors). Therefore, at this level Qualigram represents those processes that are directly related to the goals and the 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 different processes and between the processes and the external actors.

Consideration BA1: Since the goal of modeling is the use of the produced models for FSM purposes then, at the top level, represent the software to be measured as a BP.

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

Qualigram allows to detail at the top level some of the processes by representing their sub-processes and procedures. The procedures constitute the elements that are further detailed at the intermediate level. Any procedure represented at the top level must present at least one input and one output of information. Qualigram forbids representing any internal actor of the organization at the top level.

Consideration BA3: Consider as a procedure any logical instruction-set that is worth to be detailed deeper.

Consideration BA4: Represent as an external actor any user of the software that allows representing the inputs and the outputs of the modeled procedures.

Fig. 2 shows the top level model of the C-Registration System, based on the specifications of the system [8] and the annotated considerations. The specifications mention the registrar as an actor. From an organizational point of view the registrar should be considered as an internal actor, and he/she would not appear at the top level model. However, in order to represent the inputs and outputs of some of the modeled procedures the registrar has been represented as an external actor.

Additionally, the “Login” procedure, or the “Create Student/Professor” procedure, typically would not be considered as candidates to be modeled using a BP modeling language. From the organizational point of view, a business process crosses different functional departments of the organization, and this is not the case of this type of procedures. For example, in the login procedure there is only an interaction between a user and the information system; 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. Moreover, when Qualigram analyzes and models the procedures at the intermediate level, it tries to answer the who is doing what? and the what is being done? (i.e. it represents the different instructions to be executed as part of the procedure, and the different roles involved in the procedure). According to Qualigram, a procedure requires a minimum of two roles and five instructions.

Consideration BA5: At the intermediate level, represent the software being measured as an internal role.

Consideration BA6: At the intermediate level, represent as an internal role any peer software component that interacts with the software being measured.
COSMIC requires identifying those data movements that retrieve or write information from/to a persistent storage. Qualigram allows representing the tools that are used/produced by an instruction; these tools might 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.

**Consideration BA7:** Any instruction that requires to retrieve or to write relevant data from/to a persistent storage should be associated to a material tool. The material 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”: it always requires a “triggering element”. This triggering element might be any organizational event (i.e. a requirement from a customer), 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”. A triggering event is considered by COSMIC as a data movement of the Entry type. However, in the case that a functional process has to wait for additional data from the functional user after having received the triggering event, then only one entry has to be considered for the whole functional process; this is true, even in the case where the functional user requires for a prompt message for entering the additional data. Moreover, “in the business application domain control commands shall be ignored”.

**Consideration BA8:** If the procedure being modeled requires at its very beginning information to be entered by the role that triggers the procedure, then represent as the triggering event the initial submission of information.

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

**Consideration BA9:** All the error conditions identified by a role must be collected by a unique instruction executed by the same role before notifying them to other role.

COSMIC determines the Exits and the 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”. In Qualigram, each role is confined into a swim-lane. If a role A needs to pass the control of the workflow to a role B, then the role A needs to send a flow of information to role B crossing the swim-lane of role A.

**Consideration BA10:** Avoid representing flows of information between roles when those flows are only aimed at indicating a possible end of the workflow.

Based on the specifications of the system [8] and the annotated considerations, a second level model for each of the procedures depicted by Fig. 2 has been produced. Due to space limitations, this paper only presents the models for the login (Fig. 3a) and the add-professor (Fig. 3b) 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”. Observing Fig. 3, it is possible to conclude that these conditions seem to be satisfied with the intermediate level of the Qualigram language. Therefore, this research will not look into the analysis of the bottom level of abstraction. Also notice that all the procedures can be considered as **peer components** since they belong to the same layer and all have the same **level of decomposition**.
B. Mapping and Measuring

Before measuring the functional size of the C-Registration System, it is necessary to define the mapping rules between the COSMIC concepts [3] and the modeling constructs of the Qualigram language [9]. From the analysis of Fig. 2 and Fig. 3 it is possible to derive some of the required mapping rules. Table I shows all the rules that have been defined based on that analysis and the comparison of the definitions of the COSMIC concepts and the Qualigram constructs.

**TABLE 1. MAPPING BETWEEN COSMIC AND QUALIGRAM**

<table>
<thead>
<tr>
<th>COSMIC FSM Method V.3.0.1</th>
<th>Qualigram Language</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional User</td>
<td>Role</td>
<td>Only those roles that interact with the software</td>
</tr>
<tr>
<td>Boundary</td>
<td>The process box that represents the software</td>
<td>Level 2</td>
</tr>
<tr>
<td></td>
<td>The swimlane of the role that represents the software</td>
<td>Level 1</td>
</tr>
<tr>
<td>Functional Process</td>
<td>Procedure</td>
<td>The procedures included in the process box of the software</td>
</tr>
<tr>
<td>Triggering Event</td>
<td>Triggering element</td>
<td>Between roles</td>
</tr>
<tr>
<td>Data Group</td>
<td>May be provided as part of the information flow</td>
<td>Access to a persistent storage</td>
</tr>
<tr>
<td>Entry</td>
<td>An incoming flow of information</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td>An outgoing flow of information</td>
<td></td>
</tr>
<tr>
<td>Read</td>
<td>Description (R) given in a material tool</td>
<td></td>
</tr>
<tr>
<td>Write</td>
<td>Description (W) given in a material tool</td>
<td></td>
</tr>
</tbody>
</table>

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”. And an object of interest is “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 procedure “Add Professor” in Fig. 3b: the first flow of information from the registrar to the C-Reg 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, in the login procedure in Fig. 3a it is possible to add the data group “user’s data” to the “R” description of the triangle that represents the persistent storage.

Fig. 4 shows an example of how to apply the mapping rules to the “Delete Schedule” procedure. Due to space limitations only this example is provided in this paper. The measurement results are then obtained by simply adding the different data movements (entries (E), exits (X), writes (W) and reads (R)) that appear in the models representing the different procedures.

The COSMIC case study [8] presents two sets of results: “step 1” and “step 2”. The first set is obtained after the application of the COSMIC FSM method at the “functional user requirements exactly as they are written” in the original specifications of the C-Registration System. The second set results after modifying “by a further assumption” the FUR of step 1. This paper has only considered the FUR as given by step 1 of the case study.
Table II 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 considerations is presented in section V.

IV. DISCUSSION OF THE RESULTS

A. General Remark

In the initial work reported here, the modeling considerations defined in section III present 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 the measured software is represented as the main business process; however, at the intermediate level it is represented as an internal role. Qualigram is a management-oriented language and it does not ask for the representation of the information systems supporting the business processes. Therefore, for this research it has been necessary to provide a modeling consideration at each of the two levels of abstraction to represent the software components whose modeling is required for performing FSM.

B. 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. 3a: 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 I). However, according to the COSMIC measurement rules, a form sent to a user for entering information cannot be considered as an exit. As a consequence, there is a difference of one exit between the results of the reference case study and those obtained by this paper (see Table II). To address this difference, the flows of information should include the data groups, and during the measurement process it has to be analyzed if each of the flows of information corresponds to a data group or not. Something similar happens with the delete professor, delete student, and delete schedule (see Fig. 4) procedures.

The difference of one exit (Table II) 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 each 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 solved the conflicts.

The procedures (create, modify and delete) of the “Maintain Schedule” process (see Fig. 2) present an excess of one read (R) with respect to the results of the reference case study. The latter does not consider the FUR of verifying the status of the registration process (closed or not closed) before serving the student’s requirement. The reason given by the case study is the poor quality of the specifications. Even though this is a true reason, this paper has considered the required verification because it has been modeled as one of the instructions to be executed for these procedures. Something similar happens with a verification FUR at the “Close Registration” functional process. Additionally, 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, this research has not considered an extra Entry and an extra Write as a consequence of a “save schedule” FUR. However, the reference case study has considered them.

At the “Close Registration” functional process the specifications mention that it is possible that the billing system does not respond to the requirements of the registration system. If that is the case, the specifications ask...
to retry an undetermined number of times the requirement. The reference case study has not considered this as a functionality to be measured, probably because there is no data group associated to it. However, this article has measured it 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 at the “Submit Grades” functional process. After retrieving the list of students and retrieving the grades (two different data groups), the specifications ask for displaying the students’ grades. In the BP model this has been represented by only one instruction that displays the names of the students and their grades, counting 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” functional process.

V. RELATED WORK

The use of conceptual models for FSM at an early stage has been studied and analyzed in the research literature and Marin, Giachetti and Pastor [12] offer a complete survey of related works, including their own. After the publication of that survey, Lavazza and Del Bianco [13] studied the use of Unified Modeling Language (UML) [14] diagrams (use case, component, and sequence diagrams) for modeling real-time software to be measured using the COSMIC FSM method. Sellami and Ben-Abdelhah [15] studied the potential relationships between the measurements obtained from UML use case diagrams with those obtained from other UML diagrams. From all these research works only two has included the use of some kind of BP models: Demirors and Gencel [16] used an extension of the Event Driven-Process Chain (EPC) [17] to model a military application. The EPC diagrams were used as part of the requirements elicitation methodology. Additionally, the use of UML activity diagrams is proposed in [18] as one of the possible options for representing the behavioral aspects of the software being modeled. However, neither of these two latter works presents any mapping rule between the BP models and the concepts of the measurement methods being studied; the emphasis of these two works is not related to the feasibility of only using BP models for FSM.

VI. CONCLUSIONS AND FUTURE WORK

This article constitutes the first research effort to analyze the feasibility of using BP models for FSM; this approach might be very useful at the early stages of a software project. A set of modeling considerations to represent the software components to be measured according to the COSMIC FSM method has been defined for the business application domain. Additionally, the mapping rules between the COSMIC concepts and the Qualigram modeling constructs have been derived. The modeling considerations and the mapping rules have been applied to one case study. The results have been compared to those obtained by a previous work for the same cases study. The measurement results show that following the modeling considerations and using the mapping rules it is possible to successfully use BP models 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) [19] 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 in the measurement approach. A similar work will be accomplished for the real-time domain to analyze the possibility to generalize the modeling considerations and the mapping rules for the two domains.

REFERENCES


