Survey of Automation Tools Supporting COSMIC-FFP – ISO 19761

Anabel Stambollian[1], *Alain Abran*[2]

[1, 2] École de Technologie Supérieure-ÉTS, 1100 Notre-Dame Ouest,
Montréal (Québec) Canada H3C 1K3

[1]anabel.stambollian.1@ens.etsmtl.ca, [2]alain.abran@etsmtl.ca

Abstract:

Many software tools have been developed to support the implementation of the ISO-19761 COSMIC-FFP standard on functional size measurement. This paper presents a reference framework made up of the set of functions that is of interest to practitioners who implement ISO functional size measurement standards. It also includes a 2006 survey of COSMIC-related tools available both on the market and in the research community. Finally, a gap analysis is presented in which the functions that still need to be addressed by tool vendors are identified.

Keywords

COSMIC-FFP, Functional Size Measurement, FSM Support Tools, ISO 19761.

1. Introduction

ISO 14143-1:1998 [14] defines the *functional size* of a software as the size "*derived by quantifying the functional user requirements*". Also specified in this ISO standard is the fact that the '*functional user requirements*' (FUR) represent "user practices and procedures that the software must perform to fulfil the user's needs. They exclude quality requirements and any technical requirements" [9].

The functional size measurement (FSM) methods based on FUR can be used at any phase of the software life cycle. At the end of the development phase, for example, FSM results can be used to derive productivity ratios and to develop estimation models. When used in the beginning of the life cycle phase, at the requirements phase, for example, the measurement results can be used in estimation models to forecast software project effort. The software size information can then provide valuable information for requirements control, project estimation and project productivity analyses.

Four FSM methods are now recognized as ISO standards:

- ISO 19761: COSMIC-FFP [15];
- ISO 20926: Function Point Analysis (FPA) (e.g. IFPUG 4.1, unadjusted function points only) [16];
- ISO 20968: Mk II [17];
- ISO 24570: NESMA [18] (a Dutch interpretation of FPA, version 4.1).

The first generation of FSM was designed by Alan Albrecht of IBM in 1979 [4] and is referred to as "Function Point Analysis" (FPA); it is now supported by the "International Function Point Users Group" (IFPUG) [9]. The FPA, MarkII and NESMA methods were primarily designed to measure business application software.

COSMIC-FFP is referred to as a second generation of FSM and is designed to handle other types of software as well, such as real-time, telecommunications and infrastructure software. The evolution of this method is managed by an International Advisory Committee of functional size measurement experts from 14 countries.

All the steps and procedures prescribed by FSM methods to measure the functional size of a software product are initially performed manually by a measurer. This makes the use of FSM methods time-consuming and prone to errors, especially if performed by inexperienced measurers.

There have been numerous attempts to automate FSM methods, in particular for the first generation of these methods. In 1996, Mendes *et al* [22] proposed a framework to classify the functions that could be automated in the measurement of functional size.

On the market today, we see the emergence of such software support tools, which have been developed for second-generation measurement methods to help make the COSMIC-FFP method easier to use.

This paper presents a survey of COSMIC-FFP-related tools and a list of related resources available. The tools identified here have been grouped into two sets: those that are already available on the market followed by those still under development in the research community. Finally, a gap analysis of functions that tool vendors have not yet addressed is presented.

The paper is organized as follows: section 2 presents a classification framework of functions that can benefit from automated support; section 3 presents an overview of the COSMIC-FFP method framework and fundamentals; section 4 presents the results of the April 2006 survey of COSMIC-FFP support tools available in the marketplace; section 5 presents a survey of research prototypes; and, finally, section 6 identifies gaps and suggests future research directions.

2. FSM Tool Classification Framework

All the FSM methods are described as ways to manually assess software functions and to manually calculate their size. In order to make the measurement procedure easier and to reduce the margin of error in manual measurement, a variety of support tools have been developed.

A classification framework for FSM support tools can help vendors, prospective tool buyers and measurers to position these tools by:

- Identifying key dimensions of FSM;
- Proposing a comprehensive, full-cycle FSM framework;
- Identifying major tool categories;
- Identifying category functions and characteristics;
- Classifying the existing FSM tools in the identified categories.

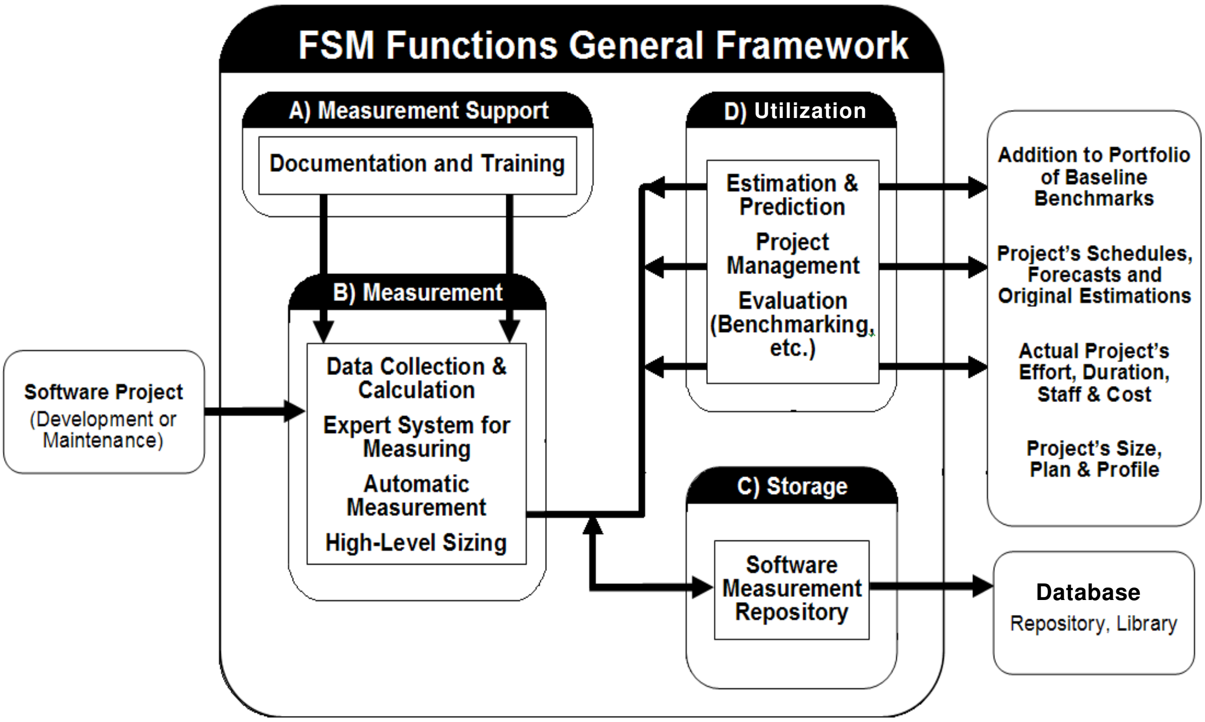
FSM Functions General Framework
A) Measurement Support
Documentation and Training
B) Measurement
Data Collection & Calculation
Expert System for Measuring
Automatic Measurement
High-Level Sizing
C) Storage
Software Measurement Repository
D) Utilization
Estimation & Prediction
Project Management
Evaluation (Benchmarking, etc.)
Software Project (Development or Maintenance)
Addition to Portfolio of Baseline Benchmarks
Project's Schedules, Forecasts and Original Estimations
Actual Project's Effort, Duration, Staff & Cost
Project's Size, Plan & Profile
Database Repository, Library

Figure 1: FSM Functions - General Framework, adapted from Mendes *et al.* [22]

Mendes *et al.* [22] developed such a classification framework, which they call the *FSM General Framework* (Figure 1). It includes 10 tool categories and addresses the following four key dimensions (Table 1):

A) Measurement Support (Documentation & Training);
B) Measurement (manual or automated);
C) Storage (Repository of measurement results);
D) Utilization (Project Planning, Sizing, Estimation & Prediction, Management, Productivity Modeling, Baselining and Benchmarking).

This tool classification framework is useful to identify the coverage of functions included in each FSM support tool, as well as to develop an understanding of the type of support the market offers (Table 2).

This classification framework was initially used in 1996 for a survey of FPA-related support tools [23]; of course, in 1996, the COSMIC-FFP method did not yet exist and was therefore not included in that survey. This paper now addresses this market segment of support tools.

Table 1: FSM Categories and Key Dimensions – adapted from [22]

FSM Key Dimensions (4)	Tool Categories (10)
A) Measurement Support	- Documentation - Training
B) Measurement	- Data Collection & Calculation - Expert System for Measuring - Automatic Measurement - High-Level Sizing
C) Storage	- Software Measurement Repository
D) Utilization	- Estimation & Prediction - Project Management - Evaluation (Benchmarking, etc.)

3. COSMIC-FFP Fundamentals

COSMIC-FFP was designed to measure the functional size of both business application software and real-time software (including: embedded, infrastructure, communications, avionics, etc.) [1] and has been successfully tested. To help classify related support tools, an overview of the COSMIC-FFP method is presented next.

The method uses only the '*Functional User Requirements*' (FUR) (of a software project) as inputs to the COSMIC-FFP measurement process. The process has two phases (Figure 2). The first is the "Mapping Phase", where a COSMIC-FFP model is generated from the software's FUR that are suitable and necessary as inputs to the measurement process. The second, the "Measurement Phase", is where the COSMIC-FFP measurement rules are applied to the FUR model, in order to derive sizes based on the ISO standard of COSMIC-FFP (ISO/IEC 19761) [15].

The COSMIC-FFP method takes into account the fact that the software's FUR can be broken down into a set of functional processes. Each of these functional processes can have a unique set of sub-processes which perform either data movements or data manipulations. This decomposition represents the basis of functional size measurement of the COSMIC-FFP software model (Figure 3). This software model distinguishes four different types of data movements: Entry", "eXit", "Read" and "Write" (Figure 4). For any of the types of movement mentioned, COSMIC-FFP considers that a data movement moves exactly one data group (for example, an object in OO software). An Entry-type data movement distinguishes the data that move from the *user* (outside the software boundary) across the boundary (through the external software interface) to the inside of the functional process (inside the software boundary). In contrast, the *eXit*-type data movement only considers the data group that moves from the inside of the functional process, across the boundary, to the user. The *Read*- and *Write*-type movements are characterized by the data that are moved from, and to, a persistent data storage (for example, a database).

It is from these data movements that the functional size of a software product is determined in the COSMIC-FFP method. To each data movement identified in the software product being measured, a single unit of measure is assigned, which is, by convention, equal to 1 "Cosmic functional size unit" (Cfsu). Therefore, the total functional size of a measured software product corresponds exactly to the sum of all the assessed data group movements. For the detailed measurement rules, see Abran *et al* [2].

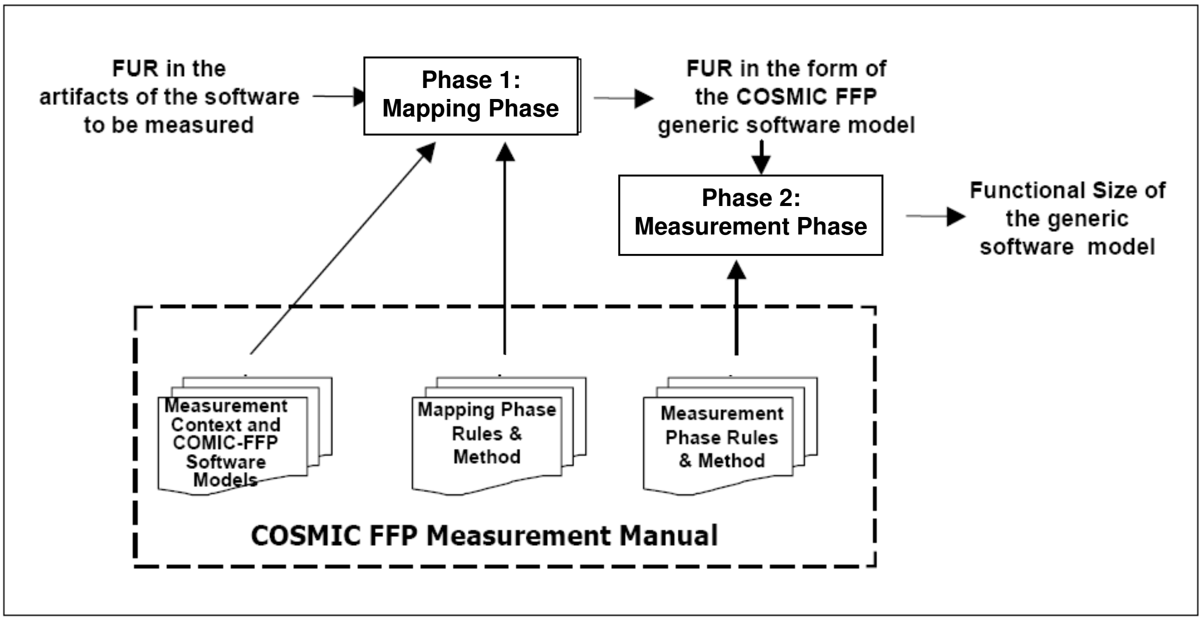
FUR in the artifacts of the software to be measured
Phase 1: Mapping Phase
FUR in the form of the COSMIC FFP generic software model
Phase 2: Measurement Phase
Functional Size of the generic software model
Measurement Context and COMIC-FFP Software Models
Mapping Phase Rules & Method
Measurement Phase Rules & Method
COSMIC FFP Measurement Manual

Figure 2: COSMIC-FFP measurement process model, adapted from [2]

4. 2006 Market Survey Results

A number of support tools are available on the market for COSMIC-FFP which help make the method easier to use. In terms of FPA support, COSMIC-FFP support tools vary considerably from one to another. The survey has shown that they cover different aspects of the method itself, the method's phases and the method's outputs for different purposes. Table 2 gives a high-level overview of the functions of the seven tools surveyed. For each of these tools, an X fills the cell where a tool addresses (partially or fully) a specific tool category. The tools are presented in Table 2 in decreasing order of their number of functions. If two tools have the same number of Xs, no particular order was used.

The seven tools surveyed are:

- COSMICXpert
- ISBSG
- MeterIT-Cosmic, MeterIT-Project, PredictIT;
- ExperiencePro;
- KnowledgePlan;
- SIESTA.

All the tools mentioned above address more than one of the four key FSM tool dimensions (presented above: Measurement Support, Measurement, Storage and Utilization).

These tools openly specialize in their respective chosen key dimensions, as they do not purport to do more than they are designed to do, nor do they plan to (see Table 2). All the tools included in this survey are briefly described next, based mostly on information provided by their suppliers.

COSMICXpert

COSMICXpert is a Web-based expert system which helps practitioners to understand COSMIC-FFP rules, apply them correctly and make correct decisions when measuring software. Developed by Desharnais [9] in 2004, this tool offers functionalities which help practitioners find the correct measurement answers (solutions) to address the following issue: “How do I interpret my software artifacts in relation to COSMIC-FFP rules, in order to derive the correct functional size?”.

Desharnais [9] has identified four distinct steps that measurement practitioners must go through when measuring functional size on the basis of FURs. These steps are (in order of execution): understand, interpret, use and solve (e.g. find the correct measurement result).

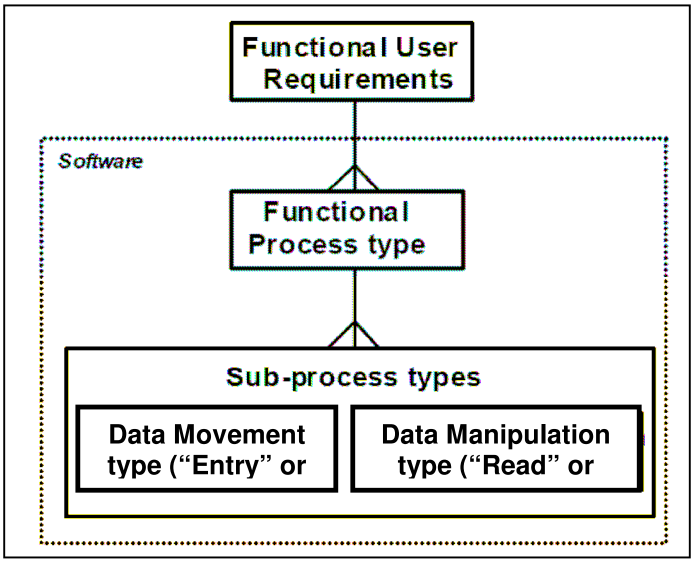
Functional User Requirements
Software
Functional Process type
Sub-process types
Data Movement type (“Entry” or
Data Manipulation type (“Read” or

Figure 3: COSMIC-FFP measurement process model, adapted from [2]

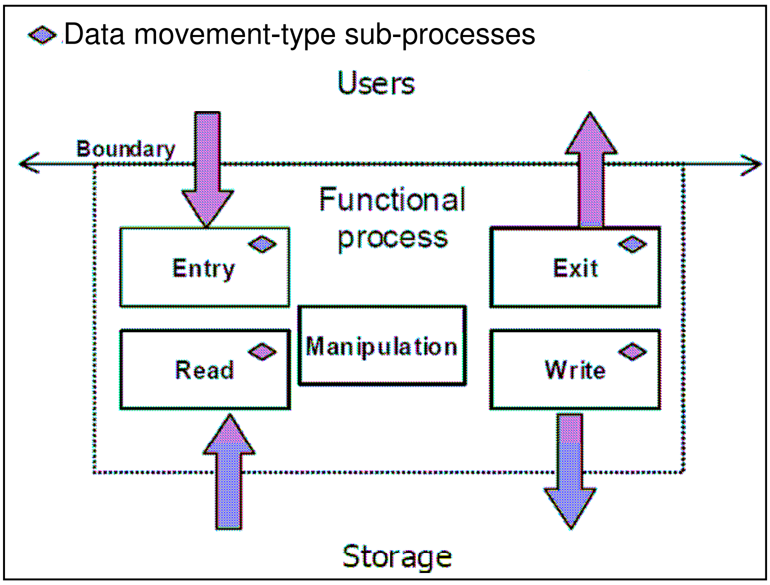

Figure 4: COSMIC-FFP sub-process types and some of their relationships [2]

Table 2: Classification of COSMIC-FFP – ISO 19761 software support tools

FSM Key Dimensions (4)	Tool Categories (10)	COSMIC Xpert	ISBSG	MeterIT-Cosmic MeterIT-Project PredictIT	Experience Pro	Knowledge Plan	SIESTA
A) Measurement Support	Documentation	X		X			
	Training	X		X			
B) Measurement	Data Collection & Calculation			X	X	X	X
	Expert System for Measuring	X					
	Automatic Measurement						
	High-Level Sizing			X	X	X	X
C) Storage	Software Measurement Repository		X	X	X	X	
D) Utilization	Estimation & Prediction			X	X	X	X
	Project Management			X	X	X	X
	Evaluation (Benchmarking, etc.)		X	X	X	X	

When a practitioner enters a key word[1], *COSMICXpert* presents the lists of concepts associated with that key word, followed by a number of "problem cases"[2] associated with the concepts on the lists. Finally, it reveals the various topics that correspond to the topological concepts on those lists. The practitioner can then reduce the number of "problem cases" proposed by *COSMICXpert* by eliminating the topological concepts that he considers not to be relevant to him. He can also reduce the number of topics by eliminating the non-relevant problem cases iteratively, until only one problem case remains (interpretation). Lastly, the practitioner can propose a response to the various topics (use) and find the appropriate recommendation (solution). He can also consult the references related to all the *COSMICXpert* concepts, such as the key words, the topological concepts, the problem cases, the topics and the recommendations.

COSMICXpert can be used free of charge, and is available at http://cosmicxpert.ele.etsmtl.ca

ISBSG Release 9

The International Software Benchmarking Standards Group (ISBSG) recently marketed its Release 9, which includes a repository of over 3,000 software projects and two software tools: *ISBSG Estimation Reality Checker Tool, Release 3,* and *ISBSG Comparative Estimation Tool, Release 5.*

The *ISBSG Estimation Reality Checker Tool, Release 3,* was designed for the most part for two purposes:

1. To compare software project estimates that have already been calculated against the project history of the *ISBSG Release 9 CD data* (to obtain a reality check);
2. To obtain an initial ball park estimate for a particular project, again based on the project history of the *ISBSG Release 9 CD data.*

The other tool, the *ISBSG Comparative Estimating Tool, Release 5,* is an automated MS Excel-based tool designed for early software project life cycle estimation and software project benchmarking. It is used to generate estimates of software project effort, delivery rate, duration and speed of delivery. The tool provides a number of parameters to select similar projects from the *ISBSG Release 9 CD data* and derived ranges of estimates.

At first, it could not be determined whether this tool's estimating equations (or those of the *ISBSG Estimation Reality Checker Tool, Release 3*) were based on projects measured with COSMIC-FFP alone or on alternate sizing methods,

[1] The term "key word" as used here comes from the software engineering domain and the COSMIC-FFP method terminology. Key words are used and retained to help a measurer close in on the environment of his problem;

[2] *COSMICXpert's* knowledge base contains more than 100 "problem cases" [9].

including COSMIC-FFP. But further investigation of the *ISBSG Repository, Release 9 (extract)* (ISBSG project data, extracted and implemented in both tools) reveals that the function points seem to include only those measured with the IFPUG method, even though *ISBSG Release 9 CD data* does include software projects sized using the COSMIC-FFP method.

Nevertheless, software benchmarking and estimating with COSMIC-FFP projects can be performed with ISBSG's latest release, either manually or by using other automated COSMIC-FFP support tools.

ISBSG Release 9 is commercially available at this address: www.isbsg.org. The ISBSG Web site[3] also mentions third party support tools that can be used with the *ISBSG Release 9 CD data*, such as: *ExperiencePro* (www.sttf.fi) and *KnowledgePlan* (www.spr.com). These third-party support tools are described later in this paper, as they support the COSMIC-FFP method.

<u>MeterIT tool suite</u>

The tool suite consisting of *MeterIT-Cosmic, MeterIT-Project* and *PredictIT* is distributed by Telmaco (www.telmaco.com). It supports users in measuring software, in measuring projects and in estimating projects.

The latest version of *MeterIT-Cosmic* (v1.5-March 2006) [32] is the main tool considered in this section. It delivers software sizes to the *MeterIT-Project* tool (project measurement and software benchmarking tool), which in turn provides calibration information to the *PredictIT* tool (project estimation tool). Table 2 does, however, consider the complete tool suite, since it offers, as a whole, functionalities that cover more key dimensions and tool categories than *MeterIT-Cosmic* alone.

MeterIT-Cosmic is a stand-alone tool which allows practitioners to size existing software products, as well as to estimate software product size (early or late in the software life cycle). Estimation of software size is made possible in *MeterIT-Cosmic* by making it possible to maintain and quantify, throughout the software life cycle, software requirements as they evolve, and by updating the estimates as the process goes on.

MeterIT-Cosmic accepts a variety of product architectures, such as stand-alone applications, large distributed software products, hosted, embedded or real-time applications. It also accepts different product types, such as new software or developments, as well as baseline or enhancement software projects (as does the COSMIC-FFP method). The tool offers product-wide measurement environment which enables a practitioner to measure a piece of software within its product context. This means that a practitioner who wants to measure a complex

[3] www.isbsg.org

software product using *MeterIT-Cosmic* is offered the choice of including the software product's whole system, layers of it or other items in it. There is even the option of measuring a stand-alone software item without having to include the overall software system's overhead.

According to the product manufacturer's description [32], the Telmaco tool also offers a means to maintain the COSMIC-FFP components, a feature which can be useful for the more experienced practitioner. This means that each of the standard measurement components (such as System, Layer, Software Item, User, Triggering Event, Object of interest, Data Group, Functional Process and Data Movement) can be created, updated or deleted in a controlled fashion, so that any software product structure can be modeled and measured according to the practitioner's needs and wishes. *MeterIT-Cosmic* also offers functionalities that enable the practitioner to automatically create and upgrade a measurement database.

MeterIT-Project provides reports of the detailed sizing information of software items and aggregated sizing information of software products, and delivers this information to the tool when needed. It also includes a tutorial.

Telmaco offers a 30-day free trial of *MeterIT-Cosmic* at http://www.telmaco.com.

<u>ExperiencePro 3.1</u>

ExperiencePro is a commercially distributed, stand-alone software tool, developed by Software Technology Transfer Finland (STTF). This tool was designed to help practitioners with software scope management, project estimation and control, change management, software measurement and data collection [11].

ExperiencePro can use ISBSG project data (see the *ISBSG Release 9* description presented earlier) as well as the data collected by STTF. STTF [27] identifies roles for practitioners using *ExperiencePro*, which are the project manager, the project customer representative, the project office people of the *ExperiencePro* user company, the database administrator of STTF, the *ExperiencePro* tool and, lastly, the service provider.

The tool is based on recognized measurement methods (including COSMIC-FFP) and it offers functionalities that support software assessment and software measurement and estimation, throughout the software life cycle (from early estimation, useful in the earliest life cycle phases, to final reviews, useful in the latest phases).

In addition to supporting project manager needs, *ExperiencePro* offers several features for project benchmarking and research activities, including [11]:

- Early estimation;

- Improved estimates;
- Progress control;
- Advanced scope management;
- Change management;
- Version comparison reports;
- Project closure and post-mortem analysis;
- Project portfolio management and integration facilities.

The tool also provides interfaces to other related systems and tools. It enables data transfers from *ExperiencePro* to other project management tools such as MS-Project, Niku Portfolio Manager and Artemis, or even MS-Excel for statistical studies. *ExperiencePro* also supports data collection from other project data sources, such as the ISBSG, or the most popular, which are Gartner and Compass [27].

ExperiencePro 3.1 is commercially available at www.sttf.fi.

KnowledgePlan 4.2

Software Productivity Research (SPR) developed *KnowledgePlan 4.2*, which offers features to help in tracking milestones, schedules, resources, actual work effort and the defects found in a software project. The SPR Web site[4] also mentions that *KnowledgePlan* can provide a rational view of tradeoffs among features, schedules, quality and costs. The cost and value implications of additional resources, languages, development tools, methods and technical changes can also be explored with the help of this tool. (Are there redunancies in this paragraph? – ch)

KnowledgePlan also provides interfaces to project management applications to create integrated, full software life cycle solutions, by enabling practitioners to export and import project data, to and from MS-Project and other software project management tools [25]. In addition, reports and output tables can also be exported for analysis and extended reporting. Some of the project management functions (such as critical path scheduling) are also offered.

According to SPR [26], *KnowledgePlan* is a stand-alone tool that offers a base of historical project data where project sizes were measured using the IFPUG and COSMIC-FFP methods. *KnowledgePlan* also uses this project data knowledge base of functional metrics to derive predictive and analytical productivity rates, given a large number of known (or assumed) parameters. The project data contained in the tool are derived from software projects that have been collected by SPR. It also offers modules to create and calibrate other sets of benchmarking data (such as ISBSG, for example).

[4] http://www.spr.com/products/knowledge.shtm

KnowledgePlan is a parametric model, with a project data knowledge base as its source of information. There are four primary inputs to the *KnowledgePlan* model [25]. The first is “Size”, which is the key scaling factor and central to the model. The other three inputs revolve around this key factor, and pertain to configurable variables that can alter the model’s final outputs, which are the “Complexity and Tools” used for the software project, the “Capability and Risk” associated with the software project, and, finally, the “Work processes” that will be used to complete the software project.

The outputs the model are “Deliverable Size”, “Assignment Scope”, “Production Rate”, “Resources”, “Predecessors”, “Work”, “Cost”, “Duration” and “Defects”.

KnowledgePlan 4.2 is commercially available by contacting SPR (www.spr.com).

SIESTA Version 1.2.3

SIESTA (SIzing & ESTimating Application) is a tool developed by Sogeti (The Netherlands) which supports sizing, estimation and control. Sogeti now offers this automated tool commercially. *SIESTA*’s functionalities focus on software sizing to support the most widely used methods of software functional sizing, such as FPA and COSMIC-FFP. Variants of these measurement methods have also been implemented in the tool, in order to enable estimation and budget planning early in the software life cycle.

In addition, maintenance variants of both methods have been implemented in *SIESTA* to estimate what Sogeti calls “functional maintenance”. As *SIESTA’s* user manual specifies [28], the tool integrates more than what is proposed in the COSMIC-FFP model. The difference is that *SIESTA* adds estimation functionalities based on Sogeti’s measurement model components which address the maintenance aspect of a software product. The measurement of maintenance, using COSMIC-FFP, is very similar to the measurement of new development [19]. *SIESTA* differs from COSMIC-FFP only in that the modified data movements are calculated, and specific situations are considered such as deleting, replacing and retesting unmodified functionalities. More precisely, the following three cases are addressed in the *SIESTA* tool:

1. Functionality deletion: the normal COSMIC-FFP size of that functionality is calculated and then multiplied by 0.1;
2. Functionality replacement: the size of the functionality to be replaced is calculated and multiplied by 1.1 (1.0 for “new development” and 0.1 for “deletion” of that functionality);
3. Unmodified functionality retesting: the sizes of the functionalities to be retested are calculated (using the COSMIC-FFP method) and multiplied by 0.1.

The latest version of *SIESTA* has been developed considering different types of practitioners. This has been achieved by assigning specific tool features to specific roles, the "Analyst" and the "Estimator". The tool offers features to enable the *analyst* to name and determine the value of the components of the project being sized, and features to enable the *estimator* to define the budget information by means of which the size of the project can be translated into effort (in hours). According to Sogeti [28], *SIESTA* can only be used one role at a time; however, during project size definition or during estimation, one can switch between these two roles.

Sogeti stipulates [28] that, based on product delivery rates per environment (hours/units) and influences (risks and opportunities), the tool can calculate the expected effort automatically. With respect to the estimation process, the historical project data have to be entered manually.

SIESTA is commercially available by contacting Sogeti (www.sogeti.nl).

5. Research Prototypes

XForms-Format

Inspired by the COSMIC-FFP measurement method, Li *et al.* [21] propose a functional size metric for interactive software using the user interface (UI) specifications, which, according to these authors in [21], have proved to be effective for effort estimation in the requirements analysis phase (first phase of the software life cycle). It is a non-commercial support system, and is still under development at this time. The above-mentioned study reports that an automated measurement support system has been developed, based on the *XForms-Format* UI specifications, which will facilitate the process of measuring the functional size of interactive software.

Related studies [20] demonstrate several extensions to the current W3C XForms (XForms is the next generation of Web Forms) for its application in FSM, and propose an *XForms-Format* UI specification that is capable of describing the abstract semantics of user-computer interactions for FSM. It also presents details with regard to the automation of the measurement procedure for measuring Standard Functional Size based on *XForms-Format* UI specifications. Moreover, it describes the architecture of the support system, as well as the conversion and measurement tools incorporated in that system.

In addition, the tool offers a custom software design environment and a set of measurement tools based on the MS-Visio platform. According to Li [20], this system supports the automatic generation of *XForms-Format* UI specifications and enables quick FSM for such specifications. The results outputted by this system have been validated systematically in Li's study, to ensure their correctness and completeness.

In summary, the system provides a UI specification format applicable to FSM, and it is one that enables description of the abstract semantics of user-computer interaction and Xforms. The system also enables automatic measurement based on the proposed UI specification format. The former manual measurement procedure is mapped to an automatic version, making the proposed format useful. Finally, the study stipulates that the system integrates an easy-to-use visual diagramming and measurement tool. This environment facilitates both the generation of UI specifications and automatic measurement to better assist in the measurement process.

A prototype version of this tool is available by contacting Zhen Li (see [20]).

µcROSE, ROOM

µcROSE[5] is a support tool (still in the research stage) which is being designed to automatically measure the functional size of software (as defined by the COSMIC-FFP method) for Rational Rose RealTime (RRRT) models. According to Diab *et al.* [10] *µcROSE*'s goal is to streamline the measurement process, ensuring repeatability and consistency in measurement, while eliminating measurement costs. *µcROSE* was originally developed to be integrated into the RRRT toolset, a CASE tool for designing embedded, real-time and distributed systems. The authors' [10] main motivation for selecting RRRT as the first target for COSMIC-FFP measurement automation was its market penetration and close correspondence with COSMIC-FFP.

First, the tool enables the automation of software measurement: Once the practitioner has identified the subsets of the system to be measured, *µcROSE* handles all the analyses and calculations based on the COSMIC-FFP model by processing the RRRT model. Second, it removes measurement variances and ensures repeatability, tasks which are both ensured by the completely automated measurement algorithm. This algorithm is based on the Diab *et al.* [10] interpretation of the COSMIC-FFP definitions and has been validated by experts who participated in defining the COSMIC-FFP method. This interpretation has been formalized in a mathematical definition that is publicly available, and has been implemented in *µcROSE.*

It is said in [10] that *µcROSE* can assist in building a database of completed development projects (containing actual COSMIC-FFP size, actual cost and other measures) to build cost estimation and defect prediction models and to manage software maintenance and outsourcing, all of which are based on functional software size.

The tool accepts software requirements of any size written in the RRRT notation, and then calculates their functional size. A practitioner can select any

[5] Pronounced "McROSE"

collection of capsules included in an RRRT model or obtain measurement data with a small number of interactions.

More precisely, the main functionalities offered by *µcROSE* are [10]:

1. Visual support of the COSMIC-FFP measurement process;
2. Generation of an RRRT model in XML format;
3. Extraction of RRRT entities required in the measurement process;
4. Analysis of C++ code included in the RRRT model;
5. Identification of functional processes, data groups and data movements;
6. Calculation of COSMIC-FFP and representation of the functional size;
7. Aggregation and reporting of measurement results.

The initial prototype was originally developed in 2001, but none is currently available for the latest version of RRRT. For more information on this study, contact the authors of this paper.

COSMIC-RUP

The *COSMIC-RUP* prototype was developed by Azouz *et al.* [6] to establish a direct mapping between COSMIC-FFP and UML concepts and notation. The objective of this research project was to demonstrate the feasibility of the approach. The prototype's goal and foundation was to extract Rational Rose artifacts to proceed to the software project measurement process. This goal included calculation of the software's functional sizes (once all the specifications had been established, by using RUP early in the life cycle) and provision of early size indicators when only high-level information was available (earlier in the software life cycle, in the requirements phase, for example).

Various artifacts for a software product can be extracted using RUP,, and each can correspond to a different level of detail in the software requirements description. Azouz and Abran [6] identified three levels of measurement, where each has its own level of granularity and its own unit of measurement:

1. At the business modeling and requirements analysis level: the use-case diagrams are used as artifacts from which the functional size can be calculated. The size unit at this level is referred to as a "Use case functional size unit" (Ufsu);
2. At the analysis level: the scenario diagrams are used as artifacts from which the functional size can be calculated. The size unit at this level is referred to as a "Scenario functional size unit" (Sfsu);
3. At the analysis (and sometimes design) level: the detailed scenario diagrams are used as artifacts from which the functional size can be calculated. The size unit at this level corresponds exactly to the ISO 19761 COSMIC functional size unit (Cfsu).

COSMIC-RUP also offers other features that help with estimation, such as an MS-Access database, designed to contain a history of project measurements for all three levels of measurement results (Ufsu, Sfsu and Cfsu). Also, outputted data (measurement results) can either be displayed or sent to an MS-Excel worksheet, where they can then be arranged, saved or printed.

For more details on the 2003 initial prototype, see Azouz [5].

Ontological formalization

Bévo *et al.* [7] proposed an *Ontological Formalization* of the COSMIC-FFP method, in order to eventually design a tool for software FSM. It is based on the identification of all the concepts handled in the COSMIC-FFP method's measurement procedure, as well as the relationships between these concepts (domain ontology). Second, it is based on the identification of all the tasks associated with the method's measurement procedure, as well as the links between those tasks (task ontology).

Bévo *et al.* [7] provide the foundations for an ontological formalization of the procedure. These ontologies can easily be used for the design of measurement tools where the tasks and concepts can be described according to the CommonKADS methodology (the leading methodology that supports structured knowledge engineering)[6].

The authors stipulate in [7] that an XML, followed by XMI translation of this ontology, is to be produced for the development of measurement results analysis and documentation tools, as well as some aspects of measurement result validation. Since XML is appropriate for document exchange, historical databases on projects could also be extended to store more detailed data pertaining to FSM. In that perspective, it is said that the ISBSG could store its project data in either XML or XMI format. This study could also be helpful for data exchange between different FSM tools (in the case where several measurement tools are used for the same FSM method).

The authors stipulate that, eventually, an ontology-oriented approach will be introduced for the complete or partial automation of a method's measurement procedure, because the work on automation is based on the implementation of ontologies.

For information about this prototype, contact Valéry Bévo [7].

[6] http://www.commonkads.uva.nl/frameset-commonkads.html

6. Observations and future work

6.1. Observations

A survey of the COSMIC-FFP support tools has been presented here, and they have been grouped into either commercially available tools or research prototypes. This concise classification of COSMIC-FFP – ISO 19761 software support tools makes it possible to perform a gap analysis of them, as well as to identify the functions that have not yet been addressed up to now.

Table 2 shows that COSMIC-FFP support tools vary considerably, as they cover different aspects of the method or method phase. These tools support one or more of the Mendes *et al.* [22] FSM life cycle phases, but none of them yet fully supports automation of the "Measurement" key function.

The initial version of COSMIC-FFP was published in 1999, and its latest version was released in 2003, the same year it was adopted as an ISO standard. Vendors have therefore only recently started to design tools to facilitate use of the method. This domain of activity is clearly growing rapidly, since almost half the COSMIC-FFP support tools assessed only support that method in their newest versions. It has also been shown that one particular tool category has not been addressed by the functions of the tools assessed: "Automatic Measurement". However, it has also been revealed that an attempt is being made in all the tools being studied at this time to fully automate this particular activity of the COSMIC-FFP method's process.

6.2. Future work

One step would be to test and evaluate all these tools, and to test each of their features and functionalities to better establish their true usefulness as COSMIC-FFP support tools.

It would also be useful to survey the support tools that have not been addressed here, thereby covering, and complying with, the other standardized FSM methods (IFPUG, MkII-FPA and NESMA).

From Table 2, it is possible to evaluate the shortfalls of each of these tools on the one hand, and, on the other, to evaluate the shortfalls inherent in combining all of them, in order to better identify the deficiencies of the market in automating the COSMIC-FFP method's process.

In fact, a tool designed to acquire final FSM results that offers full automation of FSM measurement is viewed by the authors of this paper as not requiring the practitioner's low-level input. For example, a tool that requires inputs such as the number of data movements for one software functionality (for the software being measured) actually requires the practitioner to manually count those data movements. This manual count still demands that the practitioner be an expert in the COSMIC-FFP method for counting, and does not yet significantly reduce the time taken in performing this activity. Ideally, once the requirements (of the software to be measured) are established and documented in some way (early in the software life cycle phase), automatic measurement would consider the related artifacts as FUR inputs to the support tool, without the need for direct inputs on the part of the practitioner.

This clearly reflects a market shortcoming of COSMIC-FFP support tools, although an attempt is being made with the COSMIC-FFP support tools currently under study, to bridge that gap (see section 4.3).

Acknowledgments

The opinions expressed in this report are solely those of the authors. Special thanks are due to all the tool vendors who responded to our inquiries.

References

[1] Abran, A., Symons, C., (2005), *About COSMIC-FFP,* Software Engineering Research Laboratory-GÉLOG, École de Technologie Supérieure (ÉTS), Montréal, Canada, http://www.gelog.etsmtl.ca/cosmic-ffp/index.html, accessed on 02-17-06.

[2] Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D., Symons, C., (2003), *COSMIC-FFP Measurement Manual: The COSMIC Implementation Guide for ISO/IEC 19761*: *2003*, Version 2.2, January 2003, The Common Software Measurement International Consortium, http://www.gelog.etsmtl.ca/cosmic-ffp/manual.html

[3] Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D., Symons, C., (1999), *COSMIC-FFP Technology Transitioning Data Collection Protocol,* Version 1c, December 1999, Software Engineering Management Research Laboratory of the Université du Québec à Montréal (UQAM)

[4] Albrecht, A.J., (1979), *Measuring Application Development Productivity,* Proc. IBM Applications Development Symposium, IBM Corp., Monterey, CA Oct 14-17, p. 83.

[5] Azouz, S. (2003) *Calcul avec ISO 19761 de la taille fonctionnelle des logiciels développés avec Rational Unified Process*, Master degree thesis report, Department of Computer Sciences, Université du Québec à Montréal.

[6] Azouz, S., Abran, A., (2004), *A proposed Measurement Role in the Rational Unified Process and its Implementation with ISO 199760: COSMIC-FFP*, in Software Measurement European Forum - SMEF 2004 , Rome, Italy , 2004

[7] Bévo, V., Lévesque, G., Meunier, J.-G., (2003) *Toward an Ontological Formalization for a Software Functional Size Measurement Methods' Application Process: The COSMIC-FFP Case*, L.R.G.L. – L.A.N.C.I, Université du Québec à Montréal (UQAM), IWSM2003: 13th International Workshop on Software Measurement, Montréal (Québec) Canada.

[8] COSMIC FFP News, (2006), *ISBSG database*, January 2006 Volume 2, Issue 1, http://www.gelog.etsmtl.ca/cosmic-ffp/newsLetter/COSMICNewsJan2006.pdf

[9] Desharnais, J.-M., (2004) *Application de la Mesure Fonctionnelle COSMIC-FFP: une Approche cognitive*, in Département d'informatique, Montréal, Université du Québec à Montréal.

[10] Diab, H., Koukane, F., Frappier, M., St-Denis, R. (2004), *µcROSE: Automated Measurement of COSMIC-FFP for Rational Rose Real Time*, Preprint submitted to Elsevier Science, 29 June 2004. www.dmi.usherb.ca/~frappier/Frappier/Papers/McRose-IST.pdf, accessed on 03-31-06

[11] Forselius, P. (released for STTF), (2006), *The Most Important Features of ExperiencePro 3* - Memo.

[12] IFPUG, (1999), *Function Points Counting Practices Manual – Release 4.1*, International Function Point Users Group (IFPUG), Wisconsin, USA.

[13] ISBSG, (2005), Functional Sizing Methods, © *The International Software Benchmarking Standards Group Limited*, Last Edited on 06/23/2005, http://www.isbsg.org/isbsg.nsf/weben/Functional%20Sizing%20Methods, consulted on the 02-17-2006

[14] ISO/IEC 14143-1:1998, *Information technology - Software measurement - Functional size measurement - Part 1: Definition of concepts*, International Organization for Standardization – ISO, Geneva.

[15] ISO/IEC 19761:2003, *Software engineering - COSMIC-FFP - A functional size measurement method*, International Organization for Standardization – ISO, Geneva.

[16] ISO/IEC 20926:2003, *Software engineering - IFPUG 4.1 Unadjusted functional size measurement method - Counting practices manual*, International Organization for Standardization – ISO, Geneva.

[17] ISO/IEC 20968:2002, *Software engineering - Mk II Function Point Analysis - Counting Practices Manual*, International Organization for Standardization – ISO, Geneva

[18] ISO/IEC 24570:2005, *Software engineering - NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis,* International Organization for Standardization – ISO, Geneva

[19] Koppenberg, T., Dekkers, T., (2004) *Estimating maintenance projects using COSMIC-FFP*, in 14th International Workshop on Software Measurement (IWSM) IWSM-Metrikon 2004, Konigs Wusterhausen, Magdeburg, Germany, Shaker-Verlag. http://www.gelog.etsmtl.ca/publications/pdf/825.pdf

[20] Li, Z., (2003) *Measuring Functional Size of Interactive Software: A Measurement Support System Basing on XForms-Format User Interface Specifications*, Master's degree thesis, in Engineering in Industrial and Management Systems Engineering, Waseda University, Japan

[21] Li, Z., Nonaka, M., Kakurai, A., Azuma, M., (2003) *Measuring Functional Size of Interactive Software: A Support System Based on XForms-Format User Interface Specifications*, Third International Conference On Quality Software (QSIC'03), IEEE Computer Society, p. 368, http://doi.ieeecomputersociety.org/10.1109/QSIC.2003.1319123

[22] Mendes, O., Abran, A., Bourque, P., (1996) *FP Tool Classification Framework and Market Survey*, SEMRL, IFPUG Fall Conference, Dallas, International Function Point Users Group. www.gelog.etsmtl.ca/publications/pdf/83.pdf

[23] Mendes, O., Abran, A., Bourque, P., (1996) *Function Point Tool Market Survey*, Montréal, Université du Québec à Montréal – LRGL.

[24] NESMA, (1997), *Definitions and Counting Guidelines for the Application of Function Point Analysis,* Nov 1997, (see www.nesma.org).

[25] Software Productivity Research, (2006), *SPR KnowledgePLAN Version 4.2 Specifications*, product specification sheet.

[26] Software Productivity Research, (2006), *SPR KnowledgePLAN Version 4 Advertisement*, product brochure.

[27] Software Technology Transfer Finland, (2005), *Product description of the ExperiencePro Tool*, http://www.sttf.fi/eng/products/experience/ExperienceSisalto/ExperienceTool.htm, consulted on 04/12/06.

[28] Sogeti Nederland B.V. (1991 – 2003), *SIESTA: User manual*, July 2003.

[29] St-Pierre, D., Maya, M., Abran, A., Desharnais, J.-M., Bourque, P., (1997-04), *Full Function Points: Counting Practices Manual*, Technical Report, Université du Québec à Montréal, Canada. (See www.gelog.etsmtl.ca/ffp.html).

[30] Symons, C. (released for COSMIC), (2004), *COSMIC: The Common Software Measurement International Consortium*, http://www.cosmicon.com, consulted on 02/17/06.

[31] Symons, C., (1988), *Function Point Analysis: Difficulties and Improvements*, IEEE Trans Software Eng., Jan., vol. 14, no. 1.

[32] Telmaco, (2006) Product Description, MeterIT-COSMIC Software Measurement Tool, http://www.telmaco.com/pdmetco15e.pdf

[33] The International Software Benchmarking Standards Group Limited, (2006) *ISBSG Comparative Estimation Tool,* http://www.isbsg.org/Isbsg.Nsf/weben/Comparative%20Estimation%20Tool, consulted on 03-30-06

[34] The International Software Benchmarking Standards Group Limited, (2004), *ISBSG Reality Checker Tool - Description & Demonstration*, http://www.isbsg.org/html/reality.html, accessed on 03-30-06