Guideline for the application of COSMIC-FFP

for

sizing Business applications Software

Arlan Lesterhuis

(Sogeti Nederland B.V.)

arlan.lesterhuis@sogeti.nl

Abstract:

The COSMIC-FFP functional sizing method can be applied to several software domains (such as business applications and 'real-time' software). The Measurement Manual offers a theoretical skeleton, together with examples from those domains. For each specific software domain a Guideline will be developed. A Guideline aims to describe detailed rules and to provide extensive examples for the sizing of software from that specific domain.

The first Guideline to be published gives a characterisation of the business applications domain. To apply the COSMIC-FFP method to business applications software the measurer requires a good understanding of data analysis. The Guideline gives a short explanation of data analysis and its relation with the COSMIC-FFP method. Finally, the application of the COSMIC-FFP method in the domain is explained for the End-user Measurement Viewpoint and the Developer Measurement Viewpoint, as defined in the COSMIC-FFP Measurement Manual.

The Guideline discusses the materials of the COSMIC-FFP method (boundary, layer, object of interest, identification of data movements etc.), their manifestation in the business applications domain together with many examples.

Measurement of maintenance, the extended definition of object of interest and its implications for sizing are treated. Besides, several issues as sizing authorisation, help, logging, menus and layouts are treated.

Keywords

Software, functional size, measurement, COSMIC-FFP, Guideline.

1 The Cosmic Full Function Points Sizing Method

Like Function Points Analysis (FPA), the COSMIC Full Function Points (COSMIC-FFP) sizing method is a method to measure the functional size of software. There is, however, a fundamental difference between both methods. FPA is applicable to size business applications software only. It has been used extensively in productivity analysis and estimation. In contrast, COSMIC-FFP is

not only applicable to size business applications software but also to technical software, system software and real-time software.

Basis of COSMIC-FFP is the assumption that the size of a function is reflected by the number of its 'data movements'. A data movement is a transfer of a data group (a set of attributes of one object of interest, see below). There are four types of data movements. The data movement types Entry and Exit transfer a data group *from* a function's user respectively *to* a function's user. A user may be a human user as well as an other application. The role of the data movement types Read and Write is obvious: they transfer a data group *from* respectively *to* permanent storage.

Any data movement identified (i.e. any E, X, R or W) gives 1 Cosmic functional size unit (Cfsu). The size of a functional process or an application is the number of its Cfsu's.

In business applications software, an object of interest is identified for each 'entity type' (or 'Third Normal Form' relation) found in the normalized data model of the measured software. In COSMIC-FFP, the term 'Object of interest' is used instead of 'entity-type' in order to avoid using terms related to specific software engineering methods.

Objects of interest can be of a *persistent* or a *transient* type. An object of interest is persistent if the software is required to store data about the object of interest concerned. The Read and the Write data movements always relate to one persistent object of interest.

An object of interest is called transient if the object of interest does not survive the functional process using it. Transient Object of interest come into being when analysing (normalising) input and output data groups. A data group in input or output of which the key does not correspond to a persistent object of interest defines a transient Object of interest.

2 Why a Guideline for the Business applications software domain

The basic method definitions of COSMIC-FFP are contained in the ISO/IEC 19761 COSMIC-FFP standard. They are elaborated in the COSMIC-FFP 'Measurement Manual'. This manual contains the definitions, principles and rules on the measurement method expressed in such a way that they are largely independent of the software domain. It also contains explanations and some examples of these basic concepts of several software domains to help understanding.

The COSMIC-FFP Measurement Manual has deliberately avoided many domain-specific examples. For this reason, detailed descriptions of rules and extensive examples for the sizing of software from a specific domain were needed. Another reason is that some methods are relevant for one domain but not for other domains. Entering these may result in a poorly organised Measurement Manual. As an example, to apply the COSMIC-FFP method to Business applications software requires a good understanding of certain data analysis methods.

The Guidelines are intended to provide these domain-specific examples, each for a specific software domain.

The emphasis in the Business applications Guideline (hereafter: the Guideline) is on examples that illustrate the principles and rules of the Measurement Manual and on material interpreting or translating the Cosmic-FFP principles to the business applications domain. There is little duplication of material between the Measurement Manual and the Guideline. The Guideline does not contain definitions, principles and rules. For readers of the Guideline it is therefore necessary to be familiar with the content of the Measurement Manual and any associated 'Method Update Bulletins' (all obtainable from www.lrgl.uqam.ca).

3 Structure of the Guideline

The reader is pointed out that, although the concept of the Guideline for the 'Business applications software domain' is finished, it is now being revised. This means that the final version may differ from the version that has been used for this article.

3.1 What is the 'Business applications software domain'?

The Guideline starts with a description, in general terms, of what the Guideline understands by 'Business applications software domain'.

3.2 Data analysis and its relation to COSMIC-FFP

As data analysis is a key element of COSMIC-FFP it assumes knowledge of data analysis. Two main approaches to data analysis, known as 'Entity-relationship analysis' and Relational Data Analysis are succinctly described in a separate chapter.

The background for the need of data analysis lies in the fundamental notion of 'object of interest' of COSMIC-FFP. An object of interest may be any physical thing, as well as any conceptual object or parts of a conceptual object in the world of the user (as identified from the point of view of the Functional User Requirements) about which the software is required to process and/or store data.

In the Business applications domain, an Object of interest is a synonym of 'entity-type' on an entity-relationship diagram, and has the same meaning as the subject of a relation in Third Normal Form. As the data movements are coupled to these Objects of interest, it is necessary to identify the entities.

Data analysis is important not only to identify the persistent data structures. The principles of data analysis hold when identifying transient data that may appear in the input and output components of functional processes. This is especially the case for transient Objects of interest appearing in the output of management reports or ad hoc enquiries. So it is essential to understand and to be able to apply data analysis to the input and to the output of a functional process in order to identify the different data movements that may make up the input and the output.

There is another reason why data analysis is important. The background is an old problem: by data analysis, 'code tables' may emerge as candidates for Objects of interest. Code tables are entities with often only two attributes, for instance country ID and Country name. The question is: do code tables represent Objects of interest or not? Recall, an Object of interest is a 'thing' in the world of the user about which the software is required to process and/or store data.

The Guideline gives the decisive answer: if there is a functional process that *enters* data about the 'thing' or *creates* data about it and exits that data from the software, the 'thing' corresponds with an Object of interest. Loosely speaking, only if data about a 'thing' are maintainable by a user, the thing represents an Object of interest. The definition does not demand the functional processes which enter or exit data about an Object of interest to be within the Scope of the software being measured.

4 Identification of functional processes

For the identification of functional processes, the Guideline gives some valuable hints.

4.1 CRUDL

First, the idea is not new that each possible transition from one stage to another in the life-cycle of every persistent Object of interest should correspond to a functional process. This rule is summarised by the acronym 'CRUDL' where C = Create, R = Read, U = Update, D = Delete and L = List. Every Object of interest must be created, is read, updated, deleted and listed.

The Guideline states that a separate functional process must be identified for any of these CRUDL transactions, if mentioned in the Functional User Requirements.

4.2 What is a functional process?

Often, the functional processes in the Functional User Requirements (FUR) each fill a paragraph. A point however that is often misunderstood is that, conversely, not every paragraph in the FUR describing what looks like a functional process corresponds with a functional process in the CFFF sense. The Guideline gives the decisive answer: for an elementary component of a FUR to be a functional process it is necessary to be both

- independently executable and
- triggered by an event in the world of the users.

4.3 One or more functional processes?

Sometimes, the measurer may be in doubt whether a function consists of one or more functional processes. The Guideline states that whenever the user has to make a decision (i.e. an event occurs in the realm of the user) this implies a separate trigger, and so a separate functional process is identified.

4.4 Identification of data groups and data movements

This part of the Guideline is a central section. It presents three steps to identify data groups and data movements.

The first step consists of identifying, via the entity-types or relations, the persistent object of interests. This is the basis of identifying the Read or Write data movements in any functional process in which a persistent objects of interest are retrieved or made persistent.

In the second step, the objects of interest in the input part of a functional process are identified for the purpose of identifying Entries. By identifying the separately-keyed data groups of the input whose key corresponds to a persistent object of interest one Entry is identified. For each transient data group in the input one transient object of interest and hence one Entry is identified.

In the third and last step, the objects of interest in the output part of a functional process are identified for the purpose of identifying Exits. By identifying the separately-keyed data groups of the output whose key corresponds to a persistent object of interest one Exit is identified. For each transient data group in the output one transient object of interest and hence one Exit is identified.

4.5 Other measurement conventions

In a separate section several measurement problems are treated. They are summarized here, together with their solution. For details, see the Guideline:

- Does COSMIC-FFP take navigation control data into account and if so, how? In general, control data is ignored.
- Some functional processes seem to have no triggering Entry, how is this considered? A functional process has always a triggering Entry. If there is no apparent data group to be transferred yet there is a 'Start function' Entry with its associated data manipulation.
- How does COSMIC-FFP measure batch functions? There is no principal difference in measurement between batch and other functions.
- If a data movement has multiple sources/destinations and formats, is this one or more data movements? In general, if output to two physical devices or destinations is identical, only one data movement is counted. If there are any differences beyond the completely trivial, two outputs are counted.
- How does COSMIC-FFP treat menus, GUI elements and layouts of lists, screens and menus? Menu choices, displaying an 'empty' data entry screen, showing header and footer data are not measured, unless data related to objects of interest are moved.
- How does COSMIC-FFP treat authorisation, help and log functionality? The usual data movements are identified. Calling or exploiting existing functionality is not measured.
- How does COSMIC-FFP consider the different kinds of error messages? One Exit for all error messages in any one functional process are measured.

4.6 Measurement of the size of changes to software

One of the important topics in measuring the functional size of software is to determine the size of a *change* to software. Sizing changes is so important because often more than 80% of the time software engineers *change* existing software instead of developing new software.

The Measurement Manual states that the size of any required functional change to a piece of software is by convention the arithmetic sum of the functional sizes of all the added, changed and deleted data movements of that piece of software.

It is clear what is meant by an 'added' or 'deleted' data movement. But when is a data movement changed? In the Guideline this is answered. The Measurement Manual states that a data movement has two relevant aspects: it moves a single data group and it may fulfil some specific data manipulation(s). The Guideline now defines a change to a data movement as a change to a) the data group and/or b) to its data manipulation. If one or more such changes apply to a data movement, one changed Cfsu is measured for this data movement.

The Guideline concludes by stating that a *data group* is changed if

- attributes are added to this data group and/or
- removed from the data group or
- one or more existing attributes are changed, e.g. in meaning or format

or if its *data manipulation* is changed, i.e.

- the specific formatting, presentation and/or validation of the data are changed or
- the data manipulation (i.e. processing or computation) associated with the data movement is changed.

4.7 Developer Measurement Viewpoint

The last big step forward is the specification of measurement with the Developer Measurement Viewpoint. The Measurement Manual is not very specific about this viewpoint: it contains not much more than the definition of the viewpoint and some examples.

In the Developer Measurement Viewpoint *all* functions in the FUR of the software to be developed have to be measured. In the End-user Measurement Viewpoint only the Business applications layer is seen and any peer-item structure of the application is invisible. In the Developer Measurement Viewpoint multiple layers and peer-items within layers become visible. The Scope of the measurement will define which software items and their interactions must be considered. Boundaries must now be defined between each item of software within the Scope and each of their respective Users.

Furthermore, the concept of User is interpreted strictly as defined in the Measurement Manual (‘any person or thing that interacts with the software at any time’). Hence, in the Developer Measurement Viewpoint Users now become the devices that the application software must interact with e.g. the keyboard, mouse, screen, printer, etc. And the application software may be divided into peer components, where each component is a User of the components it interacts with.

Functional processes of Business software must be identified and distinguished exactly as above for the End-user Measurement Viewpoint. But there will be additional functional processes in the Developer Measurement Viewpoint because we now ‘see’ control and usability functions (for example a ‘page up/down’ command) that have to be provided. Also, interactions between the software being measured and any peer items become visible. Rules for identifying Objects of interest, data groups and data movements, and for measuring changes to them

are the same as for the End-user Measurement Viewpoint. But there will be additional Objects of interest as a result of considering control and usability functionality.

5 Conclusion

In the Guideline for the Business applications software domain, much experience with COSMIC-FFP has been collected and many questions answered. It is to be expected that the Guideline will be a big step forward to prepare access to the method and unify measurement in the domain.

References

A public domain version of the COSMIC-FFP Measurement Manual and other technical reports, including translations into other languages can be found on the Web at www.cosmicon.com or www.lrgl.uqam.ca/cosmic-ffp.html.

1. Symons, Charles and Lesterhuis, Arlan. Guideline for the application of COSMIC-FFP for sizing Business/MIS Application Software. Soon to be published.

2. Measurement Manual, The COSMIC Implementation Guide for ISO/IEC 19761:2003, Version 2.2, January 2003. COSMIC Measurement Practices Committee.

3. ISO/IEC 19761:2003, Software Engineering – COSMIC-FFP – A functional size measurement method.