Applying a Functional Measurement Method: Cognitive Issues

Jean-Marc Desharnais
(Software Engineering Laboratory in Applied Metrics)

Alain Abran
(École de Technologie Supérieure – Université du Québec)

Abstract

The application of a software functional measurement method is an intellectual process carried out on a complex abstract artifact: this process includes both a mapping phase between the measurement model and the software model, and a measurement phase for the instantiation of the measurement rules to the derived mapping model. In industrial contexts, two types of difficulties often occur when measuring software: 1- the software models might not be available, and, when they are, they may be incomplete and, as frequently happens, may not use the same modeling approaches and formalisms and 2- applying the specific formalisms of a measurement method to a wide variety of intellectual products which are usually not fully documented can lead to problems of homogeneity (coherence) of measurement results, even among experts. The activity of mapping the functional measurement method to any type of software model, and then instantiating the specific measurement rules, in a specific context, is characteristic of an "expert task"[1] which can itself be modeled in turn within a knowledge system. This paper explores the cognitive issues arising from these mappings, and the necessary context interpretations needed to address these issues, and then proposes an expert-system approach to tackle these cognitive issues in order to achieve consistency in the measurement results.

The functional size measurement method selected to investigate the feasibility of the approach is COSMIC-FFP, and the knowledge system to be used is Help CPR. In this paper, we present: 1 - the various phases of the measurement process; 2 – cognitive issues in the application of COSMIC-FFP in the context of an organization; 3 - the experimental case-based reasoning (CBR) approach proposed to address these issues; 4 - some examples of the resolution of problems using a CBR-type tool within a diagnosis procedure.

1. Measurement process

According to Abran et al. (3), there are four phases[2] in the measurement process, as follows, and illustrated in Figure 1:

1- Design of the measurement method;

2- Measurement method application, in a specific context;

3- Measurement results analysis;

4- Exploitation of the measurement results, for instance in decision-making models, quality models and estimation models.

[1] Human expertise, even in a relatively narrow domain, is often set in a broader context which involves a good deal of common-sense knowledge about the everyday world (11, p. 5).

[2] We replace the word "step" used in the Abran et al. article (2) by the word "phase", in order to avoid confusion with the steps of the COSMIC-FFP method..

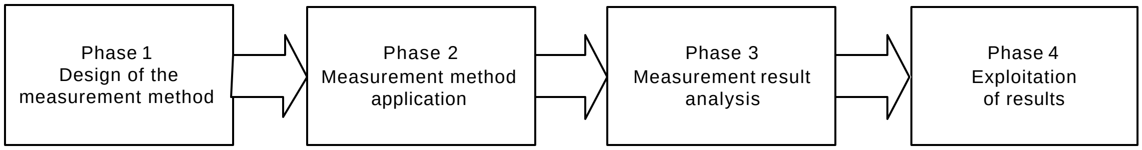

Figure 1: Phases in the measurement process

Of course, Phase 4 (exploitation of results) is the activity of most interest to managers (6); however the quality of decision-making models build and used in this phase depends heavily on the quality of the inputs to such models. Therefore, both the quality of the design of a measurement method (phase 1) and the quality of its application (phases 2 and 3) to ensure that the resulting measures are coherent[3] and of high quality, are important.

The scope of this paper is on phase 2[4] (measurement method application) and on the related challenges in the context of the measurement of software. To investigate this phase, the application of a specific software measurement method, that is COSMIC-FFP, has been selected as the object of study. In Abran et al (2), Phase 2 has been broken down into the following three activities (Figure 2):

1 - Data-gathering[5]

The data-gathering activity is specific to an organization, and, in general, it is not systematized in measurement methods documentation[6]. For example, there are generic texts about the type of documentation which the "measurer" must have, but, in practice, this depends on what is available for a specific project in an organization, and the experience and knowledge of the "measurer".

2 - Model of the software

The software to be measured is modeled using the rules of the measurement method. Several steps [7] are required. In the COSMIC-FFP method, these steps are as follows:

- identification of the layers of the software
- identification of the boundary of each piece of software
- identification of the functional processes

3 - Application of the rules of numerical assignment.

This activity is dependent on the first step of the measurement process, and, more particularly, on the definition of the rules of numerical assignment. hese rules are applied by the measurer, starting with the identification of functional sub processes (which are recognized as the entities being measured, or as the based functional component (BFC) according to the ISO 14143-1 definitions on functional size measurement), i.e. 'measurable' according to the specific model of the measurement method.

---

[3] A measurement is coherent when two measurers with same documentation (or the same information) obtain the same measurement results.

[4] To our knowledge, there has been no systematic theoretical search relating to this phase. Nishiyama et al. (14) tackled this question from the point of view of the quality of the documentation.

[5] The Abran et al. document (3) probably uses the term "data-gathering" in the sense of data as opposed to information.

[6] Documentation is an instance of the word "data". The interviews constitute another instance of the word "data". For Violaine Prince, data are "tout signifiant susceptible d'être capté, enregistré, transmis ou modifié par un agent cognitif de traitement de l'information, naturel ou artificiel" (15, p. 25). In addition, "l'information est un signifié transporté par une donnée" (idem). There may, however, be confusion about what is data and what is information. A document is generally considered as information. In the context of functional measurement, it is data, since information is linked to three factors, according to Prince (idem):
- the task or grids through which the data is decoded
- existing decoding procedures
- the decoding agent and its own cognitive universe

[7] COSMIC-FFP Measurement Guide (2) used the word "step".

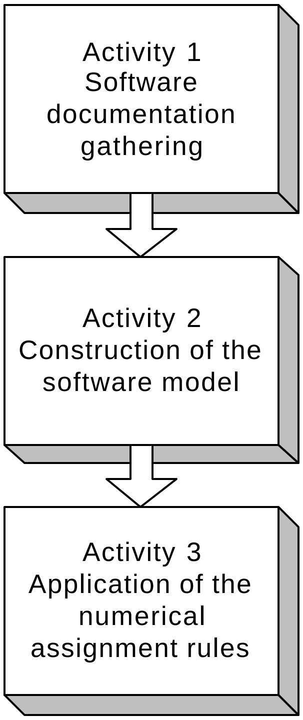

Figure 2: Method application activities

2. The measurement problem

The purpose of phase 2 is to make it possible for measurers to interpret in a coherent way[8] the rules of the measurement method by taking into account the quality and availability of the documentation for the software to measure.

From the point of view of the measurer, applying the rules of measurement ultimately means solving a specific measurement "problem". The measurer must address the following cognitive issues: he must understand the software to be measured, then he must interpret its meaning in order to identify what is to be measured, and, finally, he must use the rules of the measurement method and the rules of numerical assignment in order to arrive at the measurement solution.

Figure 3 shows the cognitive path to be followed by the measurer:

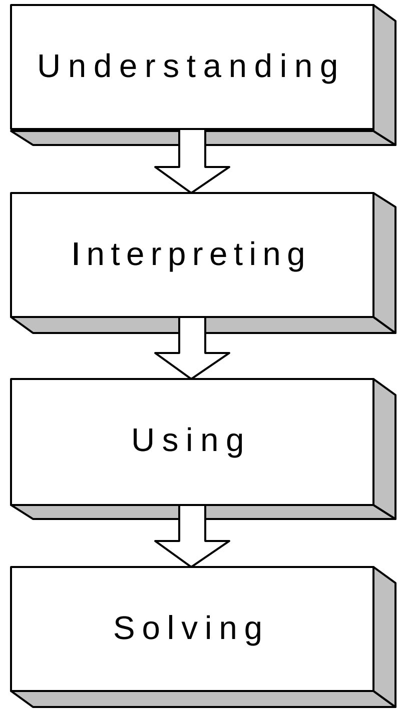

Figure 3: The measurer's cognitive path

By following this path, the measurer can solve a specific "problem" of the measurement application, i.e. the path connects the measurer's understanding of the software to the measurement rules. For example, when a software project documentation provides an entity/relation model in which the measurer can see 10 entities and 2 relations, the measurers understands that this model can help him determine the number of groups of data in the software that he has to measure. He must then interpret what relations and entities mean in the context of the measurement rules. He must next use the rules[9] relating to the groups of data. A solution could be that there are 8 valid groups of data as recognized by the model and rules of the specific measurement method he is using.

The purpose of our research project is provide help for different measurers to arrive at the same solution when using the same set of information as input to a measurement process with the COSMIC-FFP method to ensure the coherence of the measurement results. In the context of software projects, the quality of the input to the measurement process can be impacted by a lack of documentation or by the difficulty in interpreting the documentation. The quality of the documentation has a significant impact on the cognitive path of the

8 From our view, coherence implies repeatability and reproducibility, accuracy and convertibility (10).

9 Using rules means using everything which can be useful in the measurement method in connection with the problem to be solved.

measurer, mainly on his level of understanding and of his interpretation of the documentation. Through practice, we have identified two main factors with an impact on the quality of the documentation:

- poor documentation or, in some case, lack of documentation
- diversity in the representation of software models

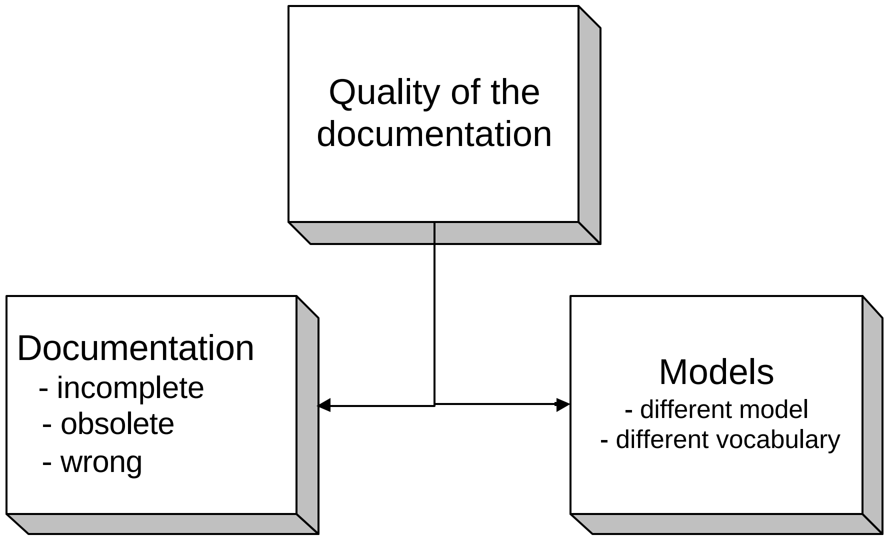

Figure 4: Quality factors of the documentation

Software documentation[10] (which is often expressed via models[11]) is, in practice, frequently incomplete, obsolete, and sometimes even wrong. Documentation problems, during the design and installation of software[12], affect the way in which the software is understood for the purposes of measurement. Moreover, even when the quality of the documentation is good, it is sometimes difficult to compare the models across organizations or organizational units because each organization tends to use its own methodology[13], which means that the modes of representation of the models, as well as the vocabulary, have various degrees of differences in the formalisms of models of representation.

In situations where the quality of the project documentation is not good enough, the measurer has to rely on the software developers to fill in the information that is lacking. The measurer must therefore either trust the documentation, and/or ask the specialists who developed the software, to figure out how to apply the measurement rules.

Figure 5 presents the mapping of phase 2 of the measurement process to the cognitive path of the measurer. Understanding and interpreting correspond to the data-gathering activity, the use and the solution to the software design activity or to the application of numerical assignment rules activity. In the prior

---

10 The words "application", "system" or "information system" are often used as synonyms in business organizations (in banks, government agencies and insurance companies, for example). The word "software" is used more frequently in other industry sectors. We use the word "software" here for preference, except if the context requires the use of a different word, as this is the word used in ISO standard 14143 (9). See also the glossary of the IFPUG Measurement Guide 4.1 (7).

11 "Les systèmes d'information d'ores et déjà modélisés font appel à plusieurs modèles de représentation: modèles de données et de traitement, modèles de connaissances, modèles organisationnels et ergonomiques, modèles de communication"(15, p. 86). For the purposes of functional measurement, we use mainly the data processes and data models.

12 The software itself could be developed in several phases, the name of the phases of which can vary according to the methodology used. There may be one or many phases, some of which address design (e.g. requests, architecture and functional analysis) and one or more phases which address installation (e.g. coding, test, implementation and training). Team composition is not necessarily the same (17).

13 A methodology is "a body of practices, procedures, and rules used by those who work in a discipline or engage in an inquiry; a set of working methods" (See WEB reference, 1).

example, the identification of the groups of data belongs to the software modeling activity. By identifying 8 groups of data, the measurer has found a solution to his problem of applying the measurement method to a specific context, in this instance the application of the measurement rules to the entity/relation model provided by the documentation.

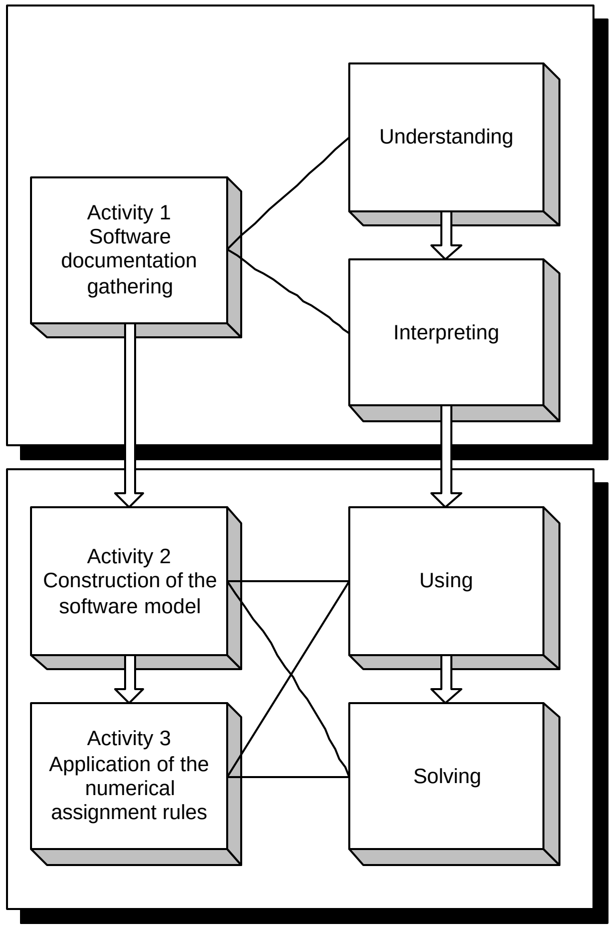

Figure 5: The measurer's path and phase 2 activities

Two additional examples are presented as illustration of information which the measurer can obtain from the documentation or by interviewing a software developer about the software to be measured:

The first example[14] describes a request from a user.

The user wishes to consult a provider's database to obtain a list of the articles purchased during the past month from a specific provider.

The measurement of the functional size of this process involves (based on the data available) a knowledge or understanding of:

- the release process modality
- the results of the process or the group of data read
- the validations and the possible results of the validations

With this information, the measurer applies the measurement rules by progressing through the following cognitive steps:

- Understand what the problem is (release, etc.)
- Interpret the meaning of the user's request
- Use the software design rules (or numerical assignment rules)
- Solve the problem

The second example involves the measurement of a report generator.

14 A similar example was used in (18).

The specifications state that the report generator must provide the user with the capacity to define and create screens and reports by using the data available, based on the groups of data of existing software. The parameters are designed, developed and delivered so that the user can choose:

- Attributes to be displayed or printed
- Selection criteria
- The display or impression format
- Among other rules for display or printing

The documentation specifies that the functionalities of the report generator are distinct from the functionalities of a simple report in the following ways:

- The attributes of the report generator are parameters, and not the attributes included in the group of software data.
- What are delivered are not reports, but the capacity to build reports.
- There may be more than one user: a user who is responsible for creating the reports and another who uses the reports, i.e. the type of user is different (the designer versus the user of a report)
- The architectural environment is different. The design of reports involves choices in terms of the parameters of the report generator and the attributes of the group of software data. There is also the display format and the user of the other display rules.

In this second example, there is more than one "problem", or cognitive difficulty in the execution of a functional measurement. Also, it is possible to uncover a hierarchy of problems, i.e. the solution of one problem can lead to the identification of another problem. This means for any type of cognitive problems a measurer encounters, he could be using Figure 5 path more than once, i.e. one path for each "problem" identified.

The identification of the functionalities of the report generator (there can be one or more processes to identify) constitutes a distinct problem. A number of questions then arise relating to identification of the report generator processes according to COSMIC-FFP[15]. The answers to these questions will lead the measurer to the identification of the processes, and then to results which can in turn lead him to another level of problems which consist in identifying sub processes.

The measurer can identify a problem which has a solution that allows direct numerical assignment, or identify a problem which leads him to the identification of a subsequent problem, and so on. Consequently the measurer must know, that is he must figure out[16], the relationships that can be established between the different problems. This could lead to the construction of a decision tree. However, trying to figure out all the number of possibilities would lead very rapidly to a "computational explosion". For this reason, the expert measurers then turn to "heuristics" to be sufficiently flexible to accommodate these various possibilities. As pointed by Dehn and Schank "the heuristics are rules that suggest way to turn or when to go back and try something different" (20, p. 363).

To address the measurement problems in a specific instance, there are two types of knowledge which the measurer must have:

- An understanding of the software (using the documentation, the models[17] and others artifacts).
- A knowledge of the COSMIC-FFP method (or any other measurement method) to enable him to apply the method and make connections among the various problems that arise at the time of measurement.

3. Proposal of a generic diagnostic procedure

We propose in this paper a generic diagnostic procedure, as a first step and not the only possible one, to help the measurer solve the various measurement problems he encounters. The description of this diagnostic procedure will be, for illustrative purposes,

15 A functional process is a unique set of data movements (entry, exit, read, write) implementing a cohesive set of Functional User Requirements. It is triggered directly, or indirectly via an 'actor', by an Event (-type) and is complete when it has executed all that is required in response to the triggering Event (-type). (2, p. 8).

16 In term of a cognitive approach, in this case, the measurer must use inference to identify new problems. Rieger (16) recommend 16 general classes of inference.

17 A developer can help the measurer to better understand software models, but, for the purposes of measurement, it is still the understanding and interpretation of the measurer that is of primary importance.

the COSMIC-FFP vocabulary[18] and measurement steps. The various COSMIC-FFP steps and sub steps that will form the basis of our topology[19] are described next.

Mapping

Step 1a: On the basis of the requirements and specifications relating to the interaction between the equipment and the software, the measurer must detect whether or not there is more than one layer, and he must formally identify it (or them, if more than one).

Step 1b: On the basis of the requirements and specifications relating to the interaction between the equipment and the software, the measurer must identify the users.

Step 2: On the basis of the requirements and specifications relating to the interaction between the equipment and the software, the measurer must identify the boundary of the software.

Step 3a: From the requirements, the measurer must identify all the functional processes of the software.

Step 3b: From the requirements, the measurer must identify all the triggers[20] of the software.

Step 3c: From the requirements, the measurer must identify all the functional groups of software data.

Measuring

Step 4[21]: Sub processes (4a: Input, 4b: Output, 4c: Read, 4d: Write) must be identified for each software functional process.

Step 5: The functional size of the software is derived from the aggregation of the measurement results, e.g. by adding the sub processes. Because this operation is strictly arithmetic, no particular expertise is required; it is thus not included within the scope of this paper.

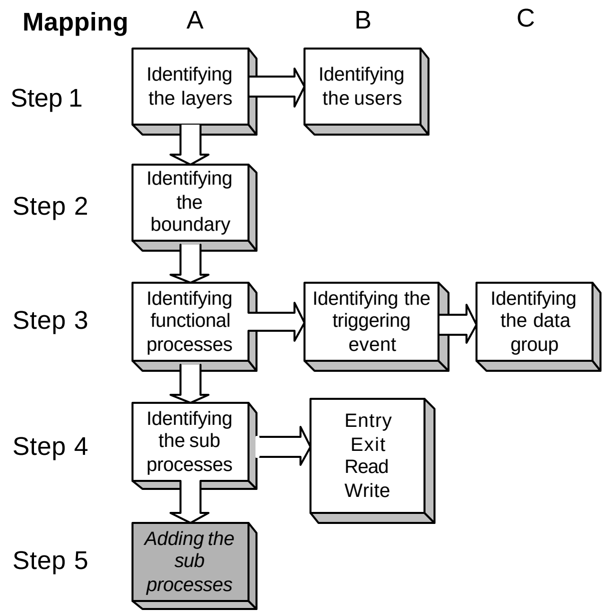

Figure 6:Topology of the COSMIC-FFP method

Based on these steps, it is possible to create a topology of problems encountered by the measurer (Figure 6). There can be multiple types of problems within each block the measurer must seek to solve. Each type of problem in this topology is referred to as a case problem in CBR terminology. The topology helps to locate the various case problems encountered by the measurer. It should also be noted that there is a hierarchy in the topology of the case problems in Figure 6. A case problem at Step 1 is more generic than a case problem at Step 2, and so on.

This topology, linked to the measurer's path, enables us to propose a diagnostic procedure providing the measurer with typical solutions to different measurement problems, and doing so in a coherent way. It is not the only possible topology, nor is it necessarily yet a complete one, for the measurer's practice, but rather a starting point. The encounter, and resolution, of new case problems can enrich this topology when there is a feedback mechanism that permits to register the knowledge being built in the problem resolution process.

The proposed diagnostic procedure is as follows:

---

18 Its vocabulary and its system of classifying the words used (e.g. the layer is more generic than the boundary, which is more generic than the process).

19 Topology: The art of, or method for, assisting the memory by associating the thing or subject to be remembered with a place.

20 The trigger event could also be studied at the sub process level, because the trigger event could be considered to be at the same level as the other sub processes. We chose to look at the trigger event at the process level, since it is used specifically to identify the processes.

21 Step 4 refers to four different problems: identification of "Entry", "Exit", "Read", "Write". We grouped them together to simplify our representation in the schema.

- The measurer must identify the nature of the problem. He might also have to identify additional problems related to the nature of the initial problem identified, if there is more than one problem
- The diagnostic tool locates case problem(s) using the topology and heuristic formula.
- For each case problem, the measurer answers the appropriate questions in order to best understand and interpret the problem.
- The specific answers proposed for the questions will lead to a proposed solution.
- A proposed solution can lead to another case problem, and, with relevant complementary information, contribute to the identification of a typical solution. It can also lead to a new problem, and the identification of a relevant solution.

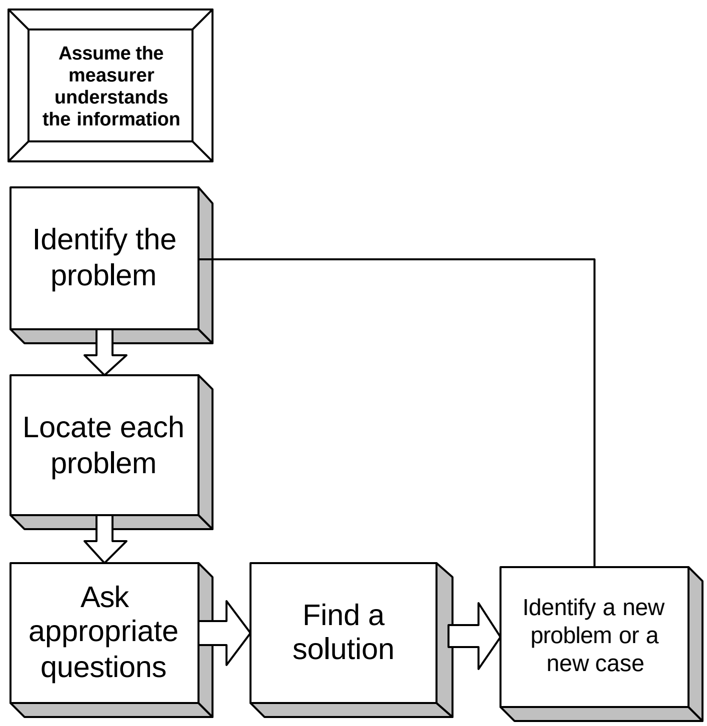

Figure 7:Diagnostic procedure

Research work has begun to implement such a diagnostic procedure by using a cognitive tool, and to demonstrate how it works based on the two examples described earlier. In this initial work, a key difficulty is how to identify what was referred to as an 'appropriate' question in the diagnostic of a measurement problem. Of course, measurers with a lot of experience are better positioned in identifying 'appropriate' questions, while beginners cannot benefit from extensive previous experience. A strategy to help beginners in measurement is to make available to them the expertise of experts through classical expert-based systems, including those referred to as 'case-base reasoning' (CBR) systems.

4. Experimentation with a cognitive diagnostic tool (Help CPR)

4.1 What is a Case-Based Reasoning (CBR) tool?

In the cognitive field, a CBR system is referred to as a tool which "*is able to utilize the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. …. CBR is also an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems*" (1).

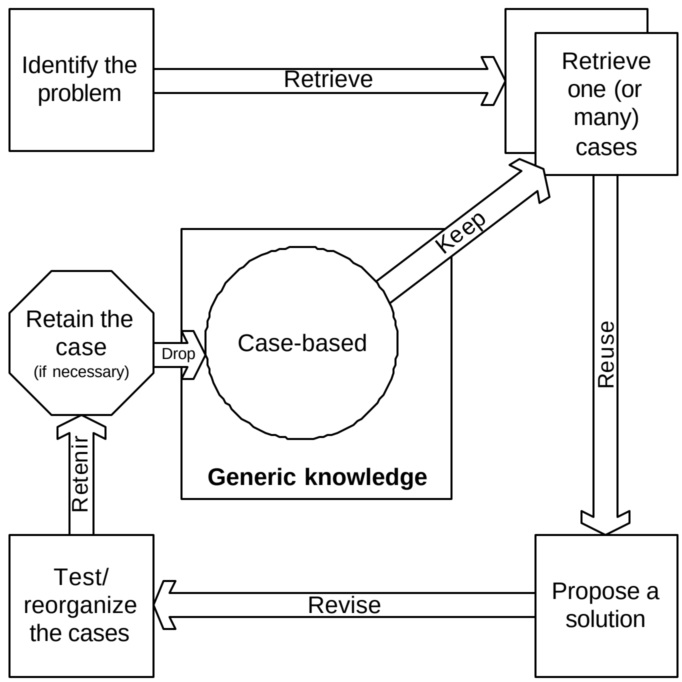

Figure 8: CBR cycle (adapted from Aamodt et al. (1) and Watson (19))

The four phases of the CBR cycle are as follows (Figure 8):

- RETRIEVE the most similar case or cases that will help solve the problem;
- REUSE the information and knowledge from that case to solve the problem;
- REVISE the proposed solution ;
- RETAIN the parts of this case likely to be useful in future problem-solving.

A new type of problem is solved by retrieving one or more previously experienced cases, reusing the case in one way or another, revising the solution based on reusing a previous case, and retaining the useful parts of the new case and incorporating them into the existing knowledge base (case base). Unless the retrieved case is very close to the solution required, the measurer may wish to repeat the process until a more satisfying solution is found.

Prior to discuss how a CBR tool can be useful to a measurer, the following concepts in CBR terminology are briefly discussed:

- What is a case?
- What is the relationship between the measurer's problems and the cases?

For Kolodner (12), a case in a CBR is a contextualized piece of knowledge representing an experience.

According to our interpretation of the work of the authors of (11,12 and 18), a case includes:

- A description of the situation (the problem encountered) surrounding the case;
- Questions, based on the situation, which describe potential solutions which have already been recorded for the case (generally in the form of probabilities);
- A result which provides an indication of the solution.

In the context of software functional size measurement, the steps and sub steps in the problem topology constitute the cases for the measurer. When a measurer has a problem with the application of the measurement method in a specific instance, and he is investigating how to solve it, the measurer using such a proposed diagnostic tool must answer a number of questions identified as relevant to the nature of the problem he recognized as application to his context of measurement. The result proposed then by the diagnostic tool is either a solution or a new case problem.

How can a tool like a CBR be useful for the measurer?

The use of a CBR embodies a number of assumptions, according to Kolodner (12)[22]:

- Regularity: The world is essentially a regular and predictable place. The same actions performed under the same conditions will normally have the same (or very similar) outcomes.
- Typicality: Events tend to be repeated. Thus, a CBR system's experiences are likely to be useful in the future.
- Consistency: Small changes in the world require only small changes to our reasoning, and so correspondingly small changes are needed to our solutions.

The application of a measurement method exhibit these characteristics and by definition, measurement standards require regularity; moreover, the same case problem , or similar ones, recur on a regular basis in various contexts.

4.2 The CBR and the diagnostic procedure

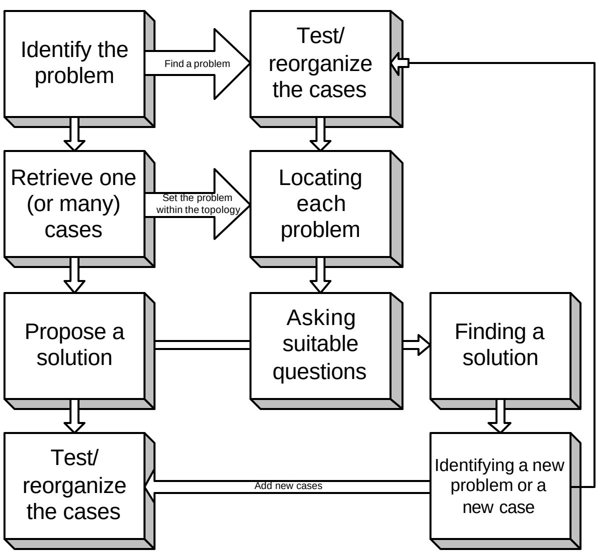

Figure 9: The CBR and the diagnostic procedure

We present in Figure 9 the parallel we have established between the CBR approach (on the left in Figure 9) and the diagnostic procedure (on the right-hand side); the links that can be found between the two approaches are also identified in Figure 9.

In the context of experimenting with the use of a diagnostic procedure for cognitive issues in software functional size measurement, the CBR-related tool sold by the firm Haley Enterprises, "Help CPR" was selected.

4.3 Functional description of Help CPR, a CBR tool.

Help CPR has a menu, making it possible to open and to back up several databases in Microsoft Access format. Help CPR organizes information on the cases into three distinct types of objects: problems, questions and actions. A fourth type of object is the "query". The cases are created by interconnecting these types of objects. It is not necessary to have the object "action" for the resolution of a case, but all the other objects are essential.

It is possible to assign (via the expert interfaces[23]) information on the "problem" level (object "problems") or "action" level (object "actions"), but not on the "question" level (object "questions"). This assignment is the equivalent of a hypertext reference.

[22] See also Watson (18, p. 200).

[23] We do not present the Help CPR Expert interfaces here. These are described in (5).

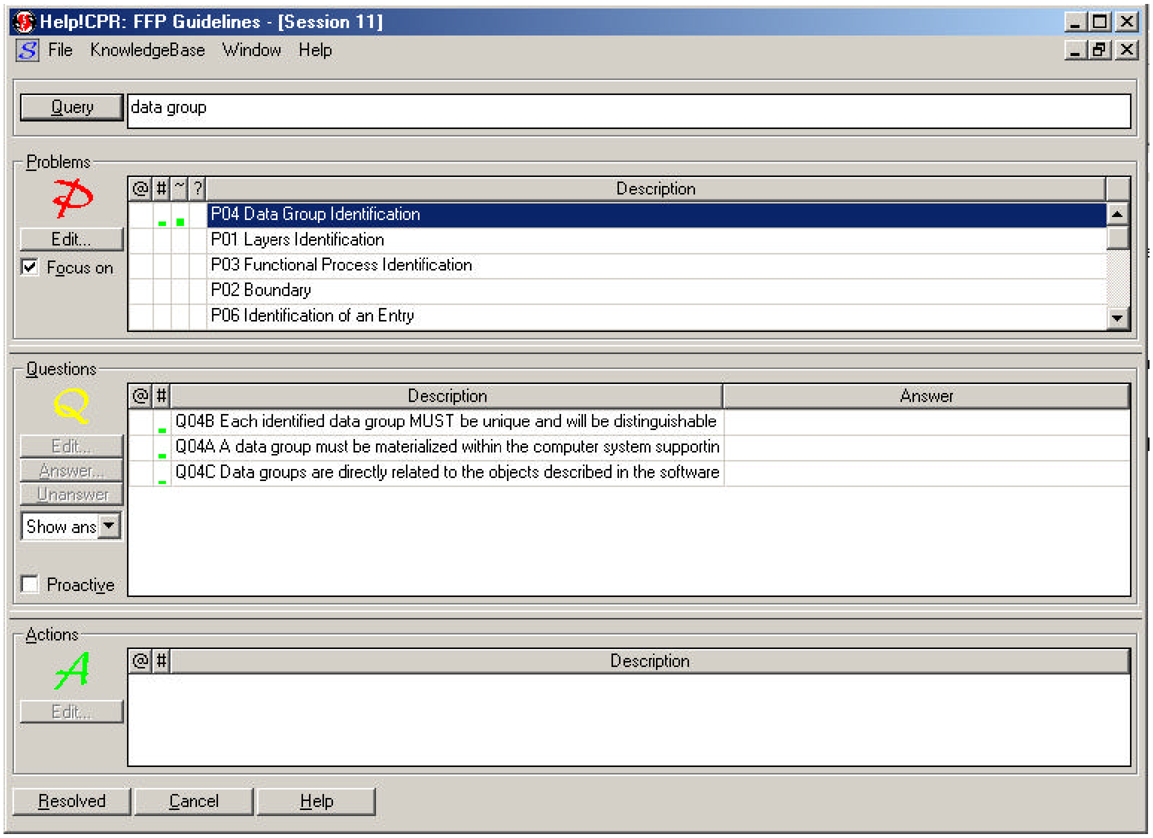

Figure 10: Help CPR user interface

It is possible to choose a particular problem by using the function or object "Query", entering a key word or sentence. Help CPR connects the identified problem and the cases in the case base. It then suggests one or many problems to solve. For example, while entering the term "data group" and the object "Query", one finds in the object "problems", the problem relating to the "data group".

Automatically, in the object "questions", the questions relating to this problem appear. The measurer must then enter a response. The answers provide possible solutions related to the problem to be solved. In this tool, the color green on the left indicates a positive answer, while the color red indicates a negative answer. It is also possible to suggest an action according to the nature of the answers to the questions. The action can point to some other "key words" to add to the knowledge base.

What is the link between Help CPR and the CBR approach? The object "Query" in Help CPR corresponds to the identification of the problem. The problems in the object "problems" list correspond to cases found in the topology. The result (green or red bar) appears on the screen (left side of the object "problems") as the user answers questions (object "questions") corresponding to suggestions for a solution. New questions are added and tested manually via the expert interfaces (not described here).

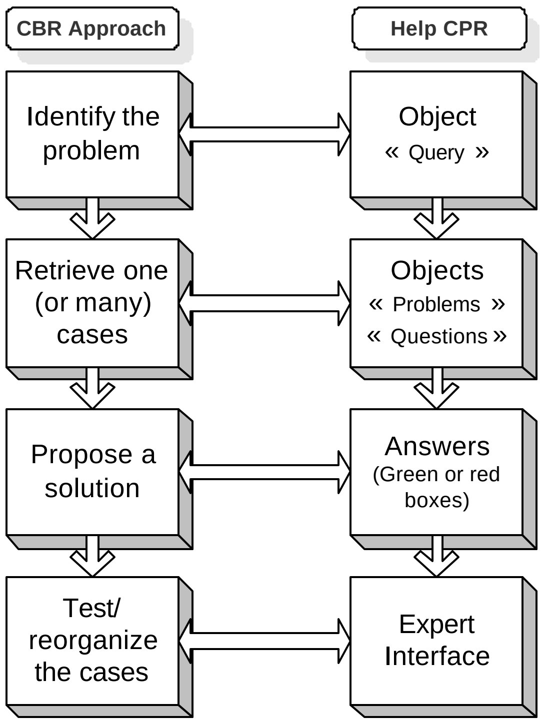

Figure 11: Links between Help CPR and the CBR approach

4.4 Illustration of Help CPR usage

The use of Help CPR in the implementation of our proposed diagnostic procedure is illustrated using the two examples presented earlier

The diagnostic procedure can be summarized as follows:

- Identify the nature of the problem;
- Locate case problem using the topology and heuristic formula;
- Use the appropriate questions to probe a problem;
- Find a solution;
- Identify a new problem or a new case, if necessary.

Example 1 (reminder)

The user asks for permission to question a provider's database to obtain a list of the articles purchased in the past month from a specific provider. Measurement of the size of this process involves (based on the data available) a knowledge or understanding of:

- the release process modality;
- the results of the process or the group of data read;
- the validations and possible results of the validations;

For this example, the measurer needs to identify the nature of the problem. For each problem identified the measurer applies the 5 activities of the diagnostic procedure. For example, the nature of the first problem is the identification of the process, which is step 3a in the topology of the COSMIC-FFP method. Appendix A shows 7 case problems related to identification of the process, identification of the trigger event, etc. For this particular example, each case problem is at the level of the topology, but in the diagnostic tool the case problem will most probably be at a lower level. The third activity is about answering the questions to determine if what the measurer can **understand** and **interpret** from the documentation represents indeed a process. The next activity is coming from the diagnostic tool that suggests a solution, and the proposed solution comes from the heuristics of the tool provided by the expert. Finally, this could lead to the identification of a new problem. For instance, after having identified a process, it is necessary to identify next its first sub processes or the trigger event.

Example 2 (reminder)

It is not necessary for the measurer to go through all the various levels of the topology every time. In the first example, the questions were at the level of steps 3 and 4 of the topology. In the second example, however, the questions start at step 1.

In the CBR diagnostic tool used, the questions do not necessarily follow in the same order, and the formulation can be different. This is why they have been numbered differently. There is a distinction, then, between the formulation of the problem by the user and its formulation in the tool. This is because it is possible to use problems in one example for another example. The experts try to generalize the typical problems (e.g. there are a limited number of questions required to identify an entry). This, of course, requires further investigation.

4.5 Some limitations of Help CPR and future research

The version of the CBR tool used had the following limitations:

- The current system makes it possible to follow, only up to a point, the cognitive path of the user. These limitations will need to be better understood to evaluate the efficiency of the cognitive tool and improve it.

- The interface explaining the "why" of the answers is not available. If the measurer were to know why the expert provided a specific answer in the diagnostic tool, he could understand himself the cognitive path of the expert when measuring, and he could learn faster and increase his own level of measurement expertise.

Further research

Our purpose was identify some of the cognitive issues in the measurement of software and the when going from a software documentation, then through all the cognitive steps required to solve the measurement problem to tackled on such a complex intellectual product. In this context, we initiated an investigation on how a CBR approach could help as a diagnostic tool meets the cognitive needs of the measurer.

We did not tackle in this paper the expert interfaces; several interfaces are already present in Help CPR which we did not describe here for simplicity sake. Further research is required on the cognitive issues relating to these interfaces. There is also a need to analyze the distinction to be made between the formulation of the problem by the user and the formulation of the problem by the expert (as formulated and subsequently structured and recorded in a CBR tool).

The effectiveness of this tool when used by novice measurers also need to be investigated. More research will be necessary to improve the actual topology and the different concepts related to the concept of topology.

5. Conclusion

Application of a software functional measurement method is an intellectual process carried out on a complex abstract artifact: this process includes both a mapping phase between the measurement model and the software model, and a measurement phase for the instantiation of the measurement rules to the derived mapping model.

The measurer must go through the following cognitive steps to address this measurement process: he must <u>understand</u> the software to be measured, then he must <u>interpret</u> its meaning in order to accurately <u>identify</u> what is to be measured, and, finally, he must <u>use</u> the rules of the measurement method and the rules of numerical assignment in order to arrive at the measurement solution.

To tackle these cognitive issues, we proposed a cognitive approach: a diagnostic procedure that follows the Case-Based Reasoning cycle approach linked to a topology based on the measurement method. From there, we illustrated the use of a diagnostic tool to help the measurer solve a specific functional measurement "problem". Finally, with a CBR-type tool, a measurement expert can enrich this topology, at the time of its use, when there is a feedback mechanism that permits to register the knowledge being built in the problem resolution process.

6. Bibliography

1. Aamodt, A., Plaza, E. (1994), Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI Communications, 7.
2. Abran, A.; Desharnais, J.-M.; Oligny, S.; St-Pierre, D.; Symons, C., COSMIC FFP – Measurement Manual version 2.1, Montréal (Canada), May, 2001. http://www.lrgl.uqam.ca/cosmic-ffp
3. Abran A., Jacquet J.-P., Dupuis R., Une analyse structurée des méthodes de validation des métriques, Laboratoire de recherche en gestion des logiciels, Département d'informatique UQAM, 1998.
4. Abran A., Analyse du processus de mesure des points de fonction, thèse présentée en vue de l'obtention du grade Philosophiae Doctor (Ph.D.) en Génie électrique et Génie informatique, mars 1994.
5. Desharnais J.-M., Description sommaire de Help CPR et Help FFP, document produit à l'occasion du cours DIC 9200 pour Alain Abran, été 2000.
6. Desharnais J.-M., Westwood P., Rollo T., The Strengths and Weaknesses of Functional Measure as a Good Negotiation Tool for New Development Projects, ESCOM Conference, April 2001.
7. Function Point Counting Practices Manual, International Function Point Users Group, version 4.1, 1999
8. ISO, Vocabulaire International des Termes Fondamentaux et Généraux de Métrologie, Deuxième édition, 1993.
9. ISO/IEC 14143-1:1997 - Information technology - Software measurement - Functional size measurement - Definition of concepts, 18 December 1996.

10. ISO/IEC 14143-3:1997 – Software engineering – Software measurement – Functional size measurement – Part 3: Verification of functional size measurement methods.
11. Jackson Peter, Introduction to Expert Systems, Addison Wesley, Third Edition, 1998, 542 pages.
12. Kolodner, J.L., Case-Based Reasoning, San Francisco: Morgan Kaufmann Publishers, 1993.
13. Morris P., Desharnais J.-M., Measuring ALL the Software not just what the Business Uses, IFPUG, 1998.
14. Nishiyama S., Furuyama, T., The validity and applicability of function point analysis, EOQ-SC'94, Basel, Switzerland.
15. Prince Violaine, Vers une informatique cognitive dans les entreprises: Le rôle central du langage, Masson, Paris, 1996, 190 pages.
16. Rieger C., Conceptual Memory. In R.C. Schank (Ed.) Conceptual information processing, Amsterdan, 1975.
17. Rivet M., Desharnais J-M., Abran A., A Cognitive Approach and the Implementation of a Measurement Program, ESCOM 2000 Conference.
18. St-Pierre D., Desharnais J.-M., Abran A., Gardner B., Definition of When Requirements Should be Used to Count Function Points in a Project, The Voice 1996 - A publication of the International Function Point Users Group, 1996.
19. Watson Ian, Applying Case-Based Reasoning: Technique for Enterprise Systems, AI-CBR, University of Salford, UK, 1997, 289 pages.
20. Dehn N. and Schank R., Artificial and Human Intelligence, Handbook on Human Intelligence, Robert J. Stenberg, 1982, pp. 352-391.

WEB Reference

1- Grand Dictionnaire Terminologique de l'Office de la Langue Française, http://www.granddictionnaire.com/_fs_global_01.

Appendix A
Problem diagnostic: examples

Problem 1
A) Identification of the problem: is this a process?
B) Location of each problem in the topology: Step 3a.
C) Questions: which the measurer must ask to identify the process and to identify whether or not the function described corresponds to the definition of a process
D) Result: the trigger event must be identified as being a single one, and conform to the definition of a trigger event. The process must be identified as corresponding to the definition of a process
E) Identification of a new problem: it will also be necessary to identify the trigger event and sub process

Problem 2
A) Identification of the problem: identification of the trigger event
B) Locate of each problem in the topology: Step 3b
C) Questions: which the measurer must ask to identify the trigger event
D) Result (or action): the trigger event is identified as being a single one, and conforms to the definition of a trigger event

Problem 3
A) Identification of the problem: identification of the group of data
B) Location of each problem in the topology: Step 3c
C) Questions: each group of data is identified and conforms to the definition of a group of data
D) Result (or action): the group of data is identified

Problem 4
A) Identification of the problem: identification of the entry
B) Location of each problem in the topology: Step 4
C) Questions: which the measurer must ask to identify the entry
D) Result (or action): each entry is identified as being a single one, and conforms to the definition of an entry

Problem 5
A) Identification of the problem: identification of the group of data read
B) Location of each problem in the topology: Step 4
C) Questions: which the measurer must ask to identify the groups of data read
D) Result (or action): each group of data read is identified as being a single one, and conforms to the definition of a group of data read

Problem 6
A) Identification of the problem: identification of the groups of data written
B) Location of each problem in the topology: Step 4
C) Questions: which the measurer must ask to identify the groups of data written
D) Result (or action): each group of data written is identified as being a single one, and conforms to the definition of group of data written

Problem 7
A) Identification of the problem: identification of the exits
B) Location of each problem in the topology: Step 4
C) Questions: which the measurer must ask to identify the exits
D) Result (or action): each exit is identified as being a single one, and conforms to the definition of group of data exits

Example 2
The second example involves several levels[24] of problems. The measurer tries to correctly identify a report generator and wishes to apply measurement rules for a report generator.

Problem (level 1)
A) Identification of the problem: identification of a report generator
B) Location of each problem in the topology: Step 1 (or new element in the topology)
C) Question: to determine whether or not this report generator is based on the characteristics of a report generator.
D) Result (or action): an explanation of what a report generator is, and the referencing of another problem which will make it possible to direct the measurer to the report generator measurement. If this is not a report generator, or the probability of it being so is weak, the reason(s) why it is not a report generator are explained.

[24] By level, we simply means that a more generic problem is at the first level, since a less generic problem is at a lower level. It is possible, in this specific case, to note the relation between the level of the problem and the different steps of the approach.

Problem (level 2)

A) Identification of the problem: to identify each process of the report generator

B) Location of each problem in the topology: Step 3

C) Questions: to help identify the main processes (COSMIC-FFP definition) of the report generator.

D) Result (or action): each process is identified; moreover, to help in checking the results, the measurer can refer to a document explaining the main identifiable processes within a report generator. These processes do not necessarily correspond to the processes identified by the measurer for this particular case, but the reference document serves as a reminder.