Verification and validation of a knowledge-based system

Jean-Marc Desharnais[1], Alain Abran[1], Julien Vilz[2], François Gruselin[2], Naji Habra[2]

[1] Département de génie électrique

École de Technologie Supérieure (ETS)

jmdeshar@ele.etsmtl.ca, aabran@ele.etsmtl.ca

[2]Facultés Universitaires Notre Dame de la Paix à Namur

Université de Namur

jvi@info.fundp.ac.be, FGruselin@xpectis.com
nha@info.fundp.ac.be

Abstract

In the real world, a knowledge-based system (KBS) must often accommodate a considerable number of references which support the particular knowledge domain. The size of such a knowledge repository makes its detailed verification challenging and subsequent maintenance onerous. New technology can help improve both the verification and maintenance of these knowledge repositories. To investigate the effectiveness of new technologies for verification and maintenance, we developed two subsequent versions of a KBS designed to improve the consistency of software measurement using ISO 19761 (the COSMIC-FFP measurement method for software functional size) and COSMIC-FFP guide [3].

The COSMIC-FFP KBS consists of a hybrid knowledge system built on case-based and ruled-based approaches. The first prototype was built in 2000 using Microsoft technology (Visual Basic 6, Access 2000, hyperlink facilities for RTF files). This prototype included 105 case problems and almost 800 files (hyperlinks) for the required references. Because of the high number of files, the verification and validation of this KBS was, of course, very challenging. This led us to design a second KBS, a Web-based prototype (XML, XSL, Java Server Page) which is much easier to verify and validate, leading to considerably improved maintainability and expandability.

This paper presents an overview of the selected hybrid KBS approach and of the first and second prototypes. It also illustrates the transitioning and quantitative benefits for the detailed KBS verification and validation process and the lessons learned during this process.

Keywords

verification, validation, knowledge-based, expert system, function point, COSMIC-FFP.

1 Introduction

There are verification and validation (V&V) problems in a KBS related to the considerable number of references in its construction:

- The knowledge of the expert (even if considerable) is not necessarily well defined. The user requirements for a KBS is then ill-defined [2]
- A text approach is sometimes the only way to express the knowledge of the expert; it is then difficult to verify the consistency of each text, when there are many, within others in the KBS
- There are many links to consider between different parts of the knowledge system.

There are a number of publications on verification and validation (V&V) for knowledge-based systems, such as Preece [17, 18] and Mesenger [16]. Related research includes testing by Ayel [4] , a rule-based system by Knauf et al. [14, 15] , case-based reasoning by Klaus-Dieter [13] and the database issue for expert systems by Coenen [6] . The terminology used by these authors is somewhat ambiguous, in that the same terms are used often but in different contexts. They all agree, however, on Barry Boehm's definition of verification (doing the system right) and validation (doing the right system) even though the techniques they used are different [5].

Verification and validation of the KBS is fairly new as a research topic. In 2001, Preece [17] noted that it is still difficult to draw conclusions about the efficiency of different verification and validation techniques because of a lack of available data. Information on knowledge-based systems is often textual and of a semantic nature, and as such, it is recognized that manual inspection can detect anomalies related to the quality of knowledge, formal methods not yet proving to be convenient for most V&V projects. Under a set of conditions, automated support tools for detecting anomalies should be useful, provided that the data is structured enough to allow some level of automation. Hayes and al. [12] have worked on CBR using XML, but their approach is not directly related to the verification and validation process. Instead, they were looking “to extend the incremental CBR approach to network applications, to examine the distributed architecture to such a system and to situate the first two strands as part of a process of creating open standards for case-based network computing…” [12, p. 11]

An overview of the KBS is presented in section 2, and the key features of the first and second prototypes of COSMICXpert in section 3. Transitioning between the two prototypes and the design of verification and validation plans are presented in section 4. Execution of the verification and validation plans is

presented in sections 5 and 6 respectively. Lessons learned are presented in section 7.

2 Overview of the selected hybrid KBS approach

One of the steps suggested by Van Heijst [20] for building a knowledge model[1] is to construct a task model. Figure 1 shows the different steps in the task model used in the design of COSMICXpert, a hybrid KBS developed to improve the measurement accuracy and repeatability of measurers using ISO 19761: 2003 – COSMIC-FFP functional size measurement method.

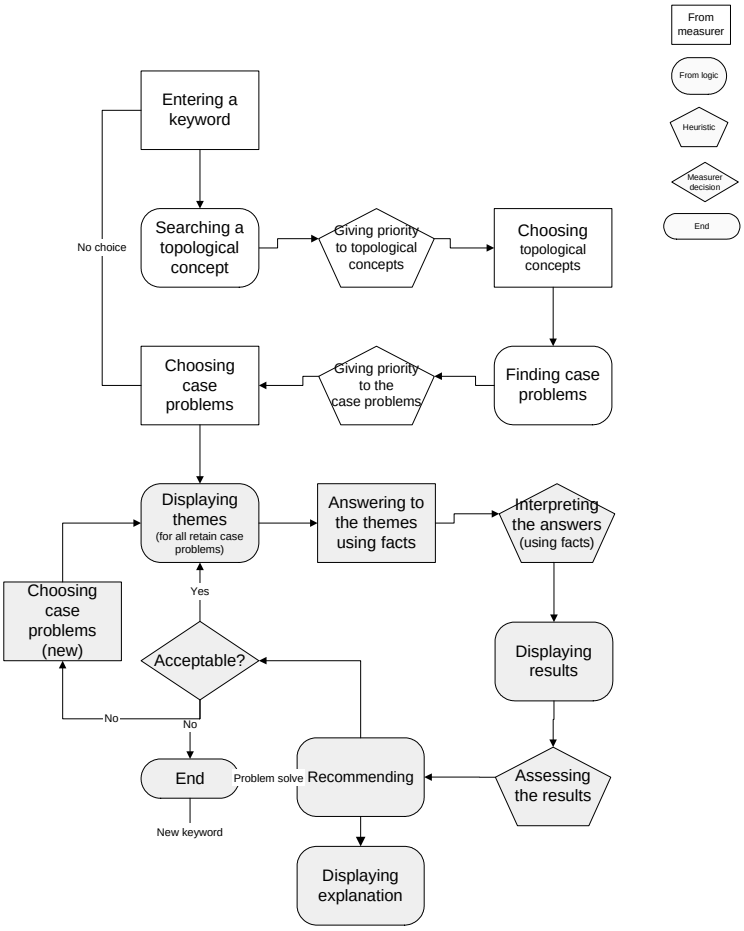

Figure 1: COSMICXpert task model of [10]

Figure 1 shows the dynamics of the role of the measurer performing each task. The square boxes show where the measurer needs to interact with the KBS

[1] In our project, the way we use van Heijst's approach is more specific than proposed by him.

system (entering a keyword, choosing topological concepts, choosing case problems, responding to the themes using facts). The first part is like CBR, because all the tasks contribute to finding a case similar to the one the measurer has to measure. The second part is rule-based, because all the tasks contribute to solving the case. In Figure 1, the heuristics formulae are represented by a pentagon (giving priority to topological concepts, giving priority to case problems, interpreting the answers, assessing the results). Some of them used certainty theory formulae proposed in MYCIN [11].

Table 1 lists and describes each task [9, 10]:

No.	***task***	***Description***
1.	**Entering a keyword**	The measurer will enter a keyword which will help the tool find the topological concepts related to the case problem
2.	**Searching a topological concept**	The tool will present the topological concepts to the measurer
3.	**Giving priority to topological concepts**	The tool will present the topological concepts to the measurer in order of priority
4.	**Choosing a topological concept**	The measurer chooses one or multiple topological concepts
5.	**Finding a case problem**	The tool will find the case problems related to the topological concepts chosen by the measurer
6.	**Giving priority to case problems**	The tool will present the case problems to the measurer in order of priority
7.	**Choosing case problems**	The measurer will choose the case problems corresponding with his/her interpretation of the problem
8.	**Displaying themes**	The tool will show all the themes related to the case problems to the measurer
9.	**Responding to? Themes**	The measurer will find facts for each theme
10.	**Rating facts**	An algorithm will rate the fact chosen
11.	**Displaying results**	The percentage will be presented to the measurer

No.	*task*	*Description*
12.	**Assessing the results**	The tool will assess the results based on heuristics
13.	**Recommending**	The tool will recommend a solution to each case problem, another case problem and/or an explanation as to why the case problem was not solved
14.	**Displaying other case problems**	The tool will suggest one or more new case problems to the user
15.	**Displaying an explanation**	The tool will give an explanation about the solution if necessary
16.	**Acceptable**	The measurer will decide if the recommendation is acceptable
17.	**Choosing case problems (new)**	The measurer will choose another case problem, either one already suggested by the tool or his own.

Table 1: Task list of the KBS tool for functional size measurement

3 Description of the first and second prototype

Two prototypes of the COSMICXpert tool were built initially. The functionality and the design of both prototypes are similar from the point of view of the COSMIC-FFP measurer (Figure 1). What changed is the technology used: Microsoft Visual Basic 6 (language) and Microsoft Access (database) for the first prototype, and a Web approach with XML structure for the second prototype.

The design of the KBS for both prototypes is composed of classes, uses cases and scenarios. Examples of a class diagram, a use case diagram and one scenario are presented in this paper. The full, detailed design appears in [8].

A class is "a description of a set of objects that share the same attributes, operations, methods, relationships, and semantics. A class may use a set of interfaces to specify collections of operations it provides to its environment" . There are a number of classes in our class diagram in Figure 2 (“diagram that shows a collection of declarative – static – model elements, such as classes, types, and their contents and relationships”) [19].

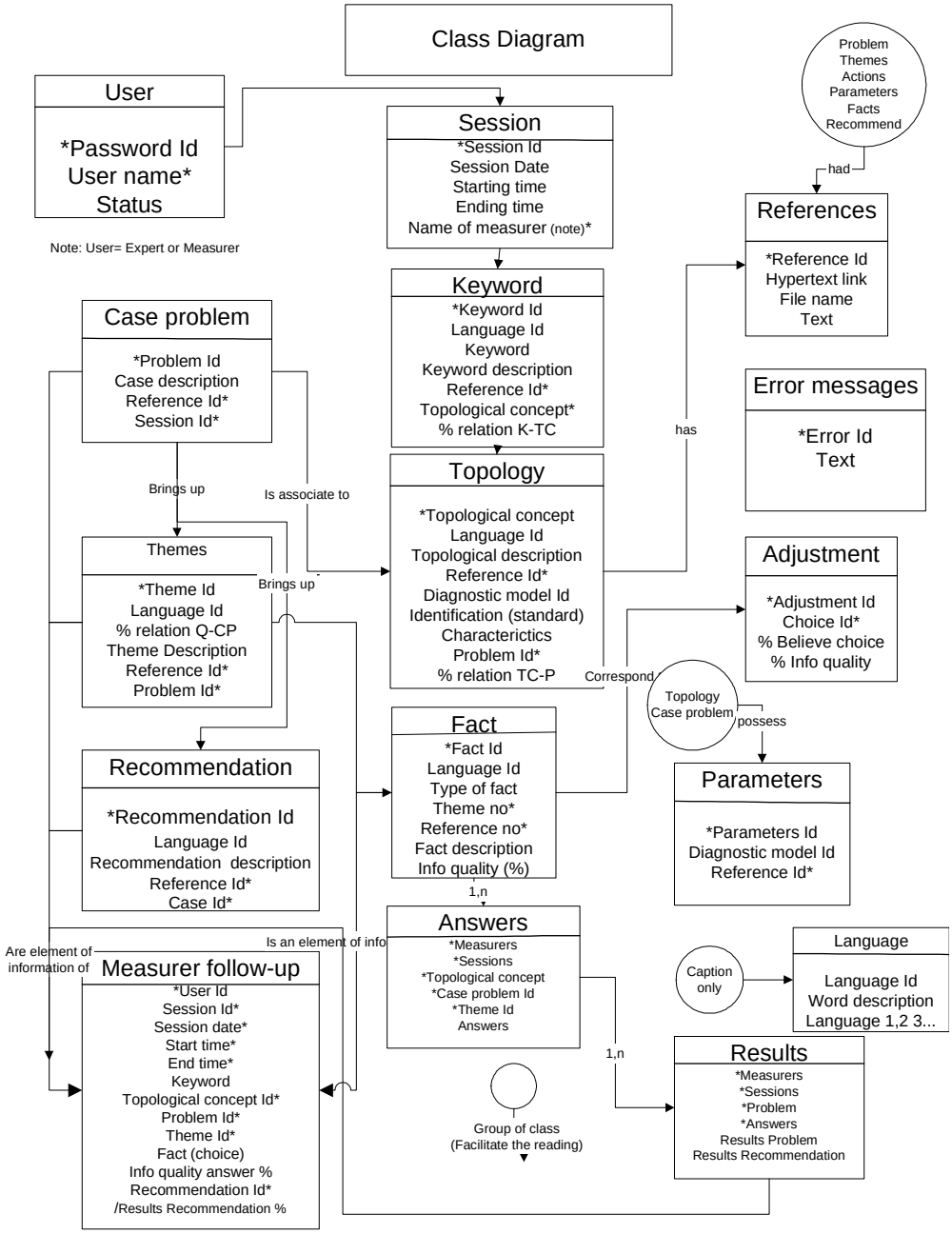

Figure 2: Class Diagram

A ‘use case’ (Figure 3) is “the specification of a sequence of actions, including variants that a system (or other entity) can perform, interactions with actors of the system.” The use case diagram (diagram that shows the relationships among actors and use cases within a system) of COSMICXpert is presented in Figure 3 with three actors (or agents): the measurer, the expert and the administrator. An actor is “a coherent set of roles that users of use cases play when interacting with these use cases. An actor has one role for each use case with which it communicates” [19].

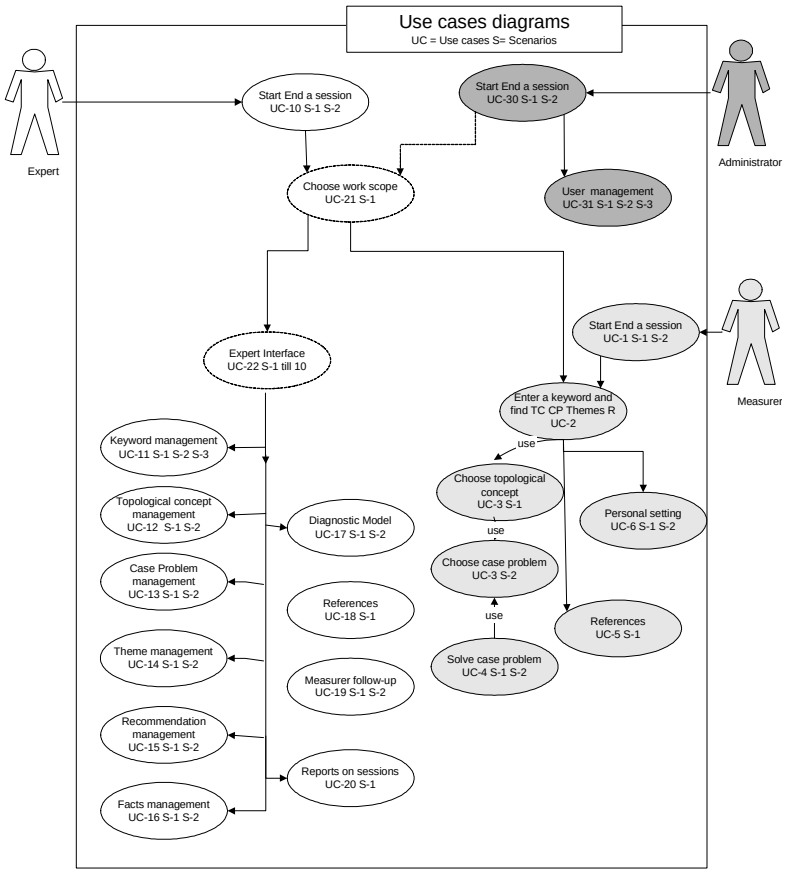

Figure 3: Use case diagram of COSMICXpert

A scenario is “a specific sequence of actions that illustrates behaviours.” As already mentioned, there are many scenarios in our KBS. The scenario in Table 2 describes the registration of a measurer in a session.

Use case 1: Measurer registration in a session
Scenario 1: Session registration
Description: A screen permitting entry of the identification of the measurer and the password Primary education actor: Measurer Secondary actor: No Pre-condition: No Short description: The measurer enters his name (recognized by the software) and his password. The identification of the session is created automatically by the software. Exception: If the name and the password do not correspond to the content of the class password, there is an error message. Post-condition (rules of termination) Access to the software Classes used: Session, measurer Data exchanged: Identification of the measurer, password, identification of the session User interface: see Table 1
Calculation: Yes: No: X

Table 2: Example of a scenario

The interfaces (expert, measurer) evolved considerably from prototype 1 to prototype 2.

The first prototype includes two interfaces:

- An interface for the expert who must put into the diagnostic tool the knowledge required for the establishment of diagnostics, and who must maintain it;
- An interface for the measurer to support him in his measurements tasks.

The measurer interface is built in accordance with the task model presented in Figure 1. An example of this initial interface is presented in Figure 4.

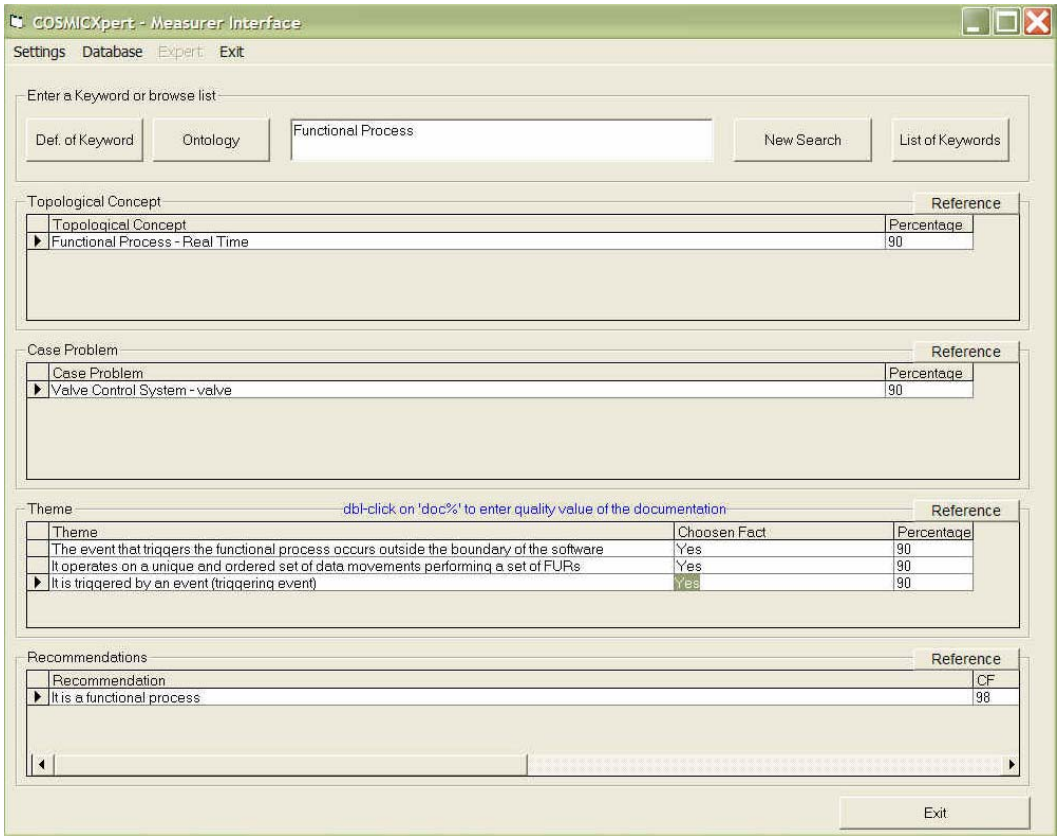

Figure 4: Measurer interface prototype 1

In the second prototype, the measurer interface is Web-based, and the expert interface has been replaced by a single input ([20]) for both the administrator and the expert.

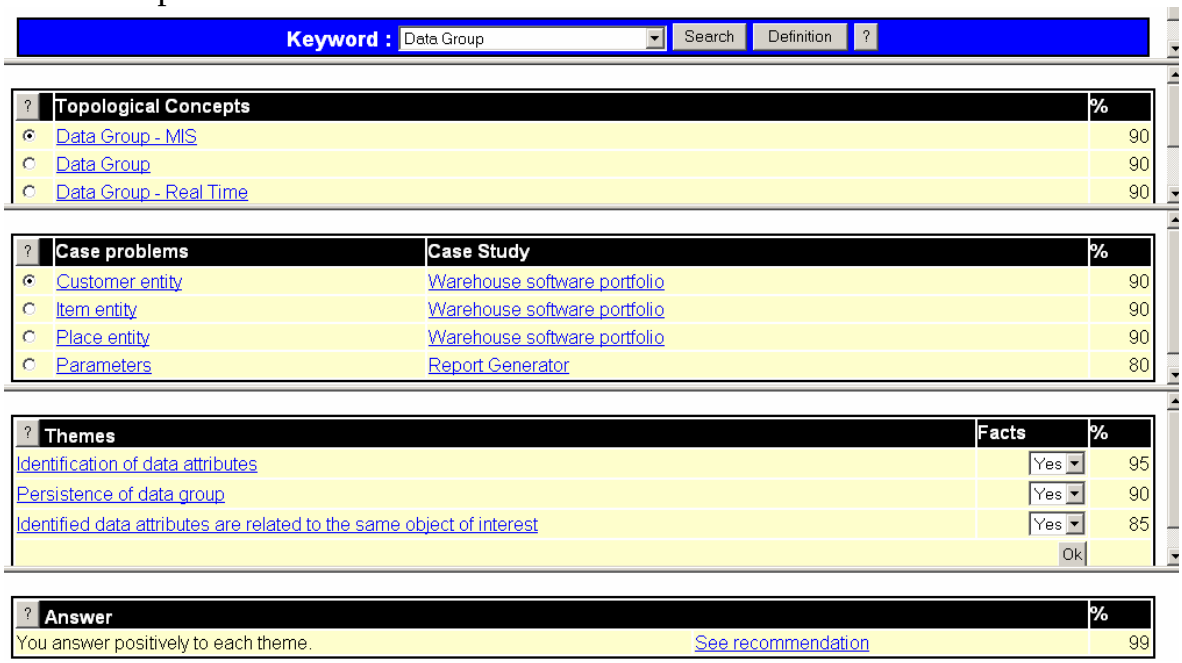

Figure 5: Measurer interface prototype 2

4 Transitioning and V&V processes

4.1 Transitioning

The first prototype required the input of more than a hundred case problems, which produced nearly 800 files, each from half a page to 3 pages in length (for a total of more than a 1000 pages). In the KBS of that prototypes, the knowledge is stored in multiple types of documents (see Figure 6); depending on the type of document, the structure and the content vary.

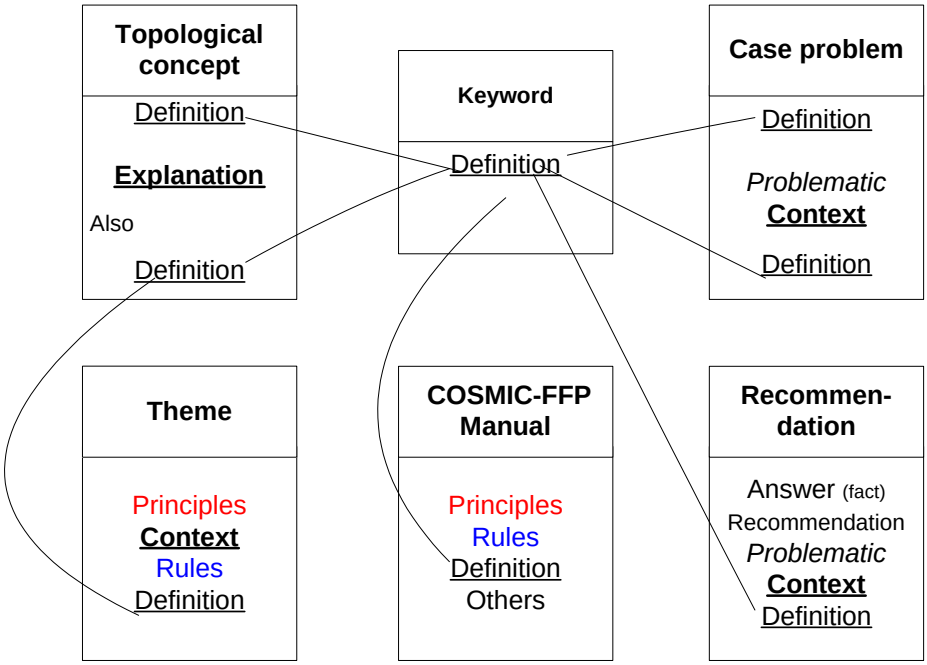

Figure 6: Types of documents (prototype 2)

It can be observed from Figure 6 that some parts of the various types of documents are repetitive within each type of document (e.g. definitions). The number of files for each document type is presented in Table 3, for a total of 779 files:

Document type	*Number of files*
Generic definition	6
Keywords	30
Topological concepts	15
Themes	250
Case problems	82

Document type	*Number of files*
Recommendations	402
Total	779

Table 3: Number of files by document type

4.2 Verification process

Verification must be carried out on the whole KBS, including all the 779 files. A key verification challenge is to ensure that each concept (definition, principles, context, rules, etc.) used in one specific document is used exactly the same way in all the document types. We also need to verify that all the links are used.

Table 4 presents the verification plan of the first prototype:

Criteria	*Techniques*	*Execution*
Coherency	Inspection	Creating XML files matching determined XML schemas (XSD)
Redundancy, reusability	Static verification	XML schema (with XSD) and XSL (reusing part of several XML files in an output)
Completeness	Check the following links: Do we have all the links? Do we have all the recommendations? Production of: - Links between topological concepts and case problems - Links between keywords and topological concepts - Links between case problems and themes - Links between themes and recommendations - Overall links	Automate with a tool to ensure that all links are present (We used ROBOT from Rational Software.)

Table 4: Verification plan

A detailed verification process was designed for each verification criterion. In this paper, we present the detailed process for arriving at the coherency criteria, as well as the verification results for each criterion.

4.3 Validation process

The validation process for this KBS consists of analyzing whether or not the issued recommendations are appropriate. We applied the validation process on the recommendations, validation being related to the user results. This implies that the validation process is not as sensitive to the technology used. However, there is a relation between the verification and the validation. If there is an effective verification process, it is easier to execute the validation process. For example, if the KBS is not working well because many links are missing, it will not be possible to validate all the recommendations.

Table 5 shows the validation plan which can be applied independently for both prototypes.

Validation Criteria	***Validation Techniques***	***Validation Execution***
1- Correctness (to ensure that all the recommendations are correct based on the different possibilities)	Decision tree Inspection	Generate, from all the case problems, all the recommendations
2- Correctness (check that the %s linking criteria to the various concepts are correct)	Decision tree Inspection	Generate a report of the various concepts with all associated percentages
3- Reliability (check that the context of each case problem described was related to the case study)	Manual inspection by an expert Automation not yet possible. Was carried out by a Ph.D. student using UML.	Generate HTML files (through XSL) that use information on the context

Table 5: Validation plan

5 Execution of the verification plan

5.1 Coherency detail process

For Craig and Tracy, “Coherency is related to the compatibility and interconnection of the different elements within a configuration. For example, within a dialog the *coherency* generally means that utterances are connected to each other in some understandable, orderly and meaningful way” [7]

To ensure the coherency of the structure when creating a specific structure, the use of a standard is recommended for all the elements of a configuration. The files were therefore created using XML (eXtensible Markup Language) using a predefined structure for each type of files. To verify this structure, an XML Schema Definition (XSD) was created using XMLSpy software. To create a specific XML file from an RTF file, the following steps were required:

- Definition of a minimal XML structure using a formal description syntax (XML Declaration Syntax or XSD);
- Identification of the various RTF files that need to be converted;
- Conversion of the RTF file into an HTML file using a filter which is partly manual: judgment of individuals who decide which information is useful based on a checklist, and
- partly automated: using Filter Tool 2.0 from Microsoft after converting the RTF file in an HTML file with Microsoft Word;
- Filter Tool 2.0 to keep the graphics in GIF (Graphic Interchange Format);
- Creation of XML files matching an XML schema and reusing parts of converted RTF files;
- For the last step, a schema approach was used. Here is an example of a schema for a concept within COSMIC-Xpert:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by Jean-Marc Desharnais (Université du Québec) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:include schemaLocation="C:\COSMICXpert\XML\textHTML.xsd"/>
    <xs:element name="Concept">
        <xs:annotation>
            <xs:documentation>Definition of concept used within COSMIC-Xpert</xs:documentation>
```

```
        </xs:annotation>
        <xs:complexType>
            <xs:complexContent>
                <xs:extension base="TextHTML">
                    <xs:attribute name="Name" type="xs:string" use="required"/>
                    <xs:attribute name="Ref" type="xs:string" use="optional"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Defining a syntax for each XML reference file made it possible to verify whether or not each file follows the syntax. It is possible then to transform the XML file into HTML format, which will make it easier for the reader to read the text. The presentation of the HTML text is possible through an XSL language or a transformation language. Via the XSL language, the programmer decides how to present the document to the user. The following example was used to define a concept within COSMIC-Xpert:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by Jean-Marc Desharnais (Université du Québec) -->
<?xml-stylesheet type="text/xsl" href="C:\COSMIC-Xpert\XML\Concept.xslt"?>
<Concept xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\COSMIC-Xpert\XML\Concept.xsd" Name="Ontology" Ref="Uschold M., Jasper R., Ontologies for
Knowledge Management, in Knowledge Management - A Micro Level Approach ed.
Rajkumar Roy, Verlag, 570 pages 1st edition, January 2001">
    <p>This is the vocabulary and the structure
of the development process (SWEBOK) and COSMIC-FFP. </p>
    <p>Uschold and Jasper wrote about the
ontology definition: &quot;In the artificial intelligence community, ontology
is generally presumed to consist of set of terms with formal axioms that define
each term's meaning. More recently, the word has been much more widely used to include
sets of terms, the terms of which may have neither explicit definitions nor carefully defined
relationships among them. An ontology may take a variety of forms, but
necessarily includes a vocabulary of terms, and some indication of what the
terms mean&quot;.</p>
</Concept>
```

Table 6: Schema approach

5.2 Quantitative outcomes

There are a number of outcomes for each type of verification criterion:

- Conformity with the structural standard (including statistics on the number of user documents that did not conform to the standard before the transformation process from RTF to XML)
- Conformity with the ontology concept definition of COSMIC-FFP within each document
- Uniformity of the syntax and grammar of each document (user document).

Finally, applying a structural standard using verification criteria significantly reduced the number of documents (from nearly 800 to 150) which could help to reduce the maintenance work on the system.

5.2.1 Conformity with the structural standard

The verification outcomes of each of the five types of XML documents (COSMIC Manual is not included) are presented next - Table 7:

- Keywords: deletion of 1 keyword, addition of 2 others and modifications to the definitions of 7 distinct keywords. We also had to modify or to add 7 references to the definition;
- Topological concept: we added 5 bibliographical references;
- Themes: 9 themes were modified, but only very slightly;
- Themes: 4 definitions were added to 30 different themes;
- Recommendations: more than 500 recommendations were verified. We added 1 to 4 definitions for 288 recommendations.

Keyword	- 1 deleted - 2 added - modification to 7 keywords - modification or addition of 7 references to the definitions
Topological concept	Addition of an example for one of the themes versus topological concepts (what does this mean?)
	- the relations between those 2 concepts were modified for 12% of the cases
Themes	- 9 themes modified - 5 references added
	1 to 4 definitions added to 30 different themes
Recommendations	1 to 4 definitions added to 288 recommendations out of the over 500 verified.

Table 7: Outcomes of the verification of concepts

Using XML format gives us the opportunity to find many errors (most of them small) which would be untraceable using the RTF format for the documents. The number of definitions added to many documents was possible because all the definitions are in one file and each definition could be used as a specific reference within each document.

5.2.2 Conformity with the ontological definitions

All the definitions were verified through keywords. The verification outcomes were:

- modification to the names of 2 definitions, removal of 1 definition (not used by another concept) and addition of 2 definitions;
- modification of the content of 6 definitions (cosmetic) ;
- addition or clarification to the references for 8 definitions.

using of a well-defined structure provided the opportunity to identify a keyword not used by other concepts (orphan keyword). The content and the references of a number of definitions were also improved.

5.2.3 Syntax and grammar for all the user documents

The syntax and grammar correction was performed manually via the Word Grammar tool. Since the number of documents was reduced significantly, it was possible to spend more time on each document to verify the syntax and grammar**.**

6 Execution of the validation plan

Two different validation approaches were used for the execution of the validation plan, one for each of the two prototypes. For prototype 1, the validation was carried out by four COSMIC-FFP experts, while a single COSMIC-FFP expert executed the validation plan for prototype 2. For prototype 1, only criteria 2 and 4 were validated.

6.1 Validation outcomes for prototype 1

Four experts from different countries executed the validation process. They were asked to use the prototype and execute different case problems to determine whether or not they agreed with the recommendations (they did not, however, check the completeness of the recommendations and the %s that link the various concepts). The experts provided comments when they did not agree. They all looked at the same 34 case problems. Only two validating criteria were used (see 2 and 4 in Table 5)

Table 8 presents the validation outcomes on criterion 2: agreement, disagreement and don't know. On the basis of the experts' comments, the disagreement and don't know responses were related to a misunderstanding of the case problems, mainly because they were ambiguous or ill-defined (criterion 4).

Type of answer	Expert 1	Expert 2	Expert 3	Expert 4	Average
% who agree	82 %	79 %	85 %	85 %	83 %
% who disagree	15 %	18 %	3 %	9 %	11 %
% who don't know	3 %	3 %	12 %	6 %	6 %

Table 8: Validation outcomes – Prototype 1 (criteria 2 and 4)

For the majority of the case problems (80%), the experts agreed with the recommendations. The range of variation between the experts is low for the agreements (between 79% and 85%), and higher for the disagreements and "don't know".

6.2 Validations outcomes for prototype 2

The validation plan was executed by a single expert for prototype 2. The same four validation criteria were used as for verification (see Table 5). Essentially, the expert agreed with the recommendations for the 104 case problems (criterion 2), but proposed some changes in the content of many of the recommendations (for example, he proposed adding definitions to 120 recommendations) (criterion 4). He also established the maximum number of recommendations (1480) and suggested using only a subset (545) of these, because many recommendations were be very similar (criterion 1). He also checked that the %s linking to the various concepts were correct. There was no modification. Because there was only one expert, the recommendations should be cross-validated by additional experts.

7 Benefits and lessons learned

The main benefits are:

- the number of anomalies in our KBS were reduced considerably;
- the KBS will be easier to maintain in the future, not only because the number of files are reduced, but also because there is a structural link between the different parts of the KBS;
- the information in the KBS is consistent because there is only one source of information;
- the information in the KBS is non redundant for the same reason;
- we know that the expert agree with the recommendations.

In summary, the verification and validation of the same KBS, but constructed with two different techniques, made it possible to demonstrate the efficiency of one technique over the other one. Our research is in sync with the emergence of using XML in the CBRS domain [1].

8. References

1. XMLSpy Enterprise Edition, Altova, 2002.
2. Aamodt, A. and Plaza, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, *7*. 27.
3. Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D. and Symons, C. Measurement Manual 2.2, UQAM, 2002, 83.
4. Ayel, M. Validation, verification and testing of knowledge based systems, England, 1991.
5. Boehm, B.W. *Software engineering economics*. Prentice-Hall Inc., Englewood Cliff, New Jersey, 1981.
6. Coenen, F., Verification and validation issues in expert and database systems: the expert systems perspective. in *Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on*, (1998), Practical, 16-21.
7. Craig, C.E. and Harris, R.C. Total Productivity Measurement at the Frim Level *Sloan Management Review*, 1973, pp. 13-29.
8. Desharnais, J.-M. Application de la mesure fonctionnelle COSMIC-FFP: une approche cognitive *Informatique*, UQAM, Montreal, 2003, 187.
9. Desharnais, J.-M., Abran, A., Mayers, A., Buglione, L. and Bevo, V., Knowledge Modeling for the Design of a KBS in the Functional Size Measurement Domain. in *KES 2002*, (Crema, Italy, 2002), IOS Press, 7.
10. Desharnais, J.-M., Abran, A., Mayers, A. and Küssing, T., Design of a diagnostic tool to improve the quality of functional measurement. in *Proceedings of the 12th International Workshop on Software Measurement*, (Magdeburg, Germany, 2002), Shaker-Verlag, 52-60.
11. Durkin, J. *Expert system: Design and Development*. Prentice Hall, New York, 1994.
12. Hayes, C., Cunningham, P. and Doyle, M. Distributing CBR using XML, Department of Computer Science, Trinity College, Dublin, Technical Report 1998.
13. Klaus-Dieter, A. Validation of Case-Based Reasoning Systems, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, 1997.

14. Knauf, R., Gonzalez, A.J. and Abel, T. A framework for validation of rule-based systems. *Systems, Man and Cybernetics, Part B, IEEE Transactions on, 32* (3). 281-295.

15. Knauf, R., Gonzalez, A.J. and Jantke, K.P., Validating rule-based systems: a complete methodology. in *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, (1999), 744-749 vol.745.

16. Meseguer, P. and Preece, A. Expert system validation through knowledge base refinement. *Knowledge Engineering Review, v 10* (n 4). 331-343.

17. Preece, A. Evaluating Verification and Validation Methods in Knowledge Engineering. in (ed), I.R.R. ed. *Micro-Level Knowledge Management*, 2001, 123-145.

18. Preece, A. and Decker, S. Intelligent web services. *Intelligent Systems, IEEE [see also IEEE Expert], 17* (1094-7167). 15-17.

19. Rational Software Corporation UML Semantics Appendix M1-UML Glossary.

20. Van Heijst, G., Schreiber, A.T. and Wielinga, A. Using Explicit Ontologies in KBS Development, University of Amsterdam, Department of Social Science Informatics, Amsterdam, 1997.