A Knowledge-Based System for Knowledge Management Capability Assessment Model Evaluation

JAVIER ANDRADE, JUAN ARES, RAFAEL GARCÍA,
SANTIAGO RODRÍGUEZ, SONIA SUÁREZ
Software Engineering Laboratory, Communications and Information Technologies
University of A Coruña
Campus de Elviña s/n, 15071, A Coruña
SPAIN
{jag, juanar, rafael, santi, ssuarez}@udc.es http://www.tic.udc.es

Abstract: - It is now commonly accepted that high quality knowledge management programmes lead to competitive advantages for the organizations. Several knowledge management maturity models have been proposed with the aim of evaluating the quality of knowledge management programmes in organizations. These models are classified into two large groups: CMM-based models and models that are not CMM-related. One of the best known CMM-based models is the Knowledge Management Capability Assessment (KMCA) model. Even so, the acquisition of a knowledge management level may imply a considerable amount of audits. It is therefore very interesting to minimise the costs by paying only for the truly indispensable audits. This article proposes a Knowledge-Based System that makes it possible to evaluate an organization at a KMCA maturity level. It limits the services of an auditor to those cases in which the system's response complies with the requested knowledge management maturity level. This clearly implies an important cost reduction for audits with negative results. The design of this system is based on the CommonKADS methodology, and its implementation was carried out with the Clips tool.

Key-Words: - Audit, Clips, CommonKADS, Knowledge-Based System, Knowledge Management, Maturity Model

1 Introduction

It is now commonly accepted that an important share of the real value of an organization resides in its own knowledge, in particular the one that comes from experience. Knowledge plays an increasingly larger role in organizations and may consider it the most important factor of production in a knowledge economy [1]. This situation has lead to a new discipline, known as Knowledge Management (KM) [2], whose objective consists of gaining the most from the knowledge which all the organizations possess by means of its adequate and explicit management. Specifically, KM can be defined as the discipline that works providing not only the right information and knowledge to the right people, but also in the right moment and in the right form for taking the best decision for a specific problem [3]. This commonly accepted situation and the growth of the KM discipline have caused organizations to dedicate numerous efforts in order to retain and to institutionalise the knowledge they possess [4].

In recent years, investments in KM initiatives have grown and the related literature counts a fair number of success stories [5, 6, and 7]. Principles and practices for the implementation of KM programmes have been developed, and researchers have proposed maturity models as the main way to represent the KM programmes development process.

As a result, there is a growing number of the so-called Knowledge Management Maturity Models (KMMM): researchers and developers have proposed maturity models as the formal way of capturing the development process of Knowledge Management, evaluating the aspects that define and control it and make it more efficient [8].

Maturity models are applicable to this field because they describe the development of an entity in a certain lapse of time, the entity being any matter of interest: the functioning of an organization, a process, etc.

Maturity models exist in numerous areas, going from the Capability Maturity Model (CMM), widely used in software development, to maturity models in the field of psychology, e.g. the Markov pyramid [9].

For the purpose of this paper, the KM is the entity of interest for the maturity models. Thus, a

KMMM describes the growth stages that an organization can expect to have to go through in the course of a KM program.

These types of models are classified into two large groups: CMM-based models and models that are not CMM-related. We have taken the CMM model as a reference because it is one of the most prestigious and popular models in software development.

The purposes of the CMM are twofold: on the one hand, evaluate how organizations develop software (i.e., the quality of the processes they follow and the mechanisms they use), and on the other hand serve as a guide towards continuous improvement [10]. This model contemplates five maturity levels that represent the quality of the software development process in the organization. These levels are, ranged from minor to major, the following: initial, repeatable, defined, managed, and optimised, with a major CMM level implying more quality.

As shown in Table 1, each maturity level, with the exception of the first, defines a series of Key Process Areas (KPAs) with which the company must comply in order to be positioned in that level, as well as a series of skills that must exist in the organization in order to implant the software processes appropriately. These skills typically involve resources, organizational structures, and trainings.

Level	KPAs
1. Initial	None
2. Repeatable	1. Requirements Management 2. Software Project Planning 3. Software Project Tracking & Oversight 4. Software Subcontract Management 5. Software Quality Assurance 6. Software Configuration Management
3. Defined	1. Organizational Process Focus 2. Organizational Process Definition 3. Training Program 4. Integrated Software Management 5. Software Product Engineering 6. Intergroup Coordination 7. Peer Reviews
4. Managed	1. Quantitative Process Management 2. Software Quality Management
5. Optimizing	1. Defect Prevention 2. Technology Change Management 3. Process Change Management

Table 1. CMM Levels & KCAs.

It is worth mentioning that CMM has evolved towards what is now known as CMMI. CMMI emerged from the accumulated experience with CMM and is based on the same principle: the quality of a product or system is mainly the consequence of the quality of the process applied during its development and maintenance [11]. Nevertheless, for the classification purposes of this work the original CMM model is enough.

As a consequence, the KMMMs that are based on CMM have certain maturity levels (similarly to CMM) and, like CMM, present a staged representation. They also possess a series of key process area, and each maturity level is described by a set of features. Examples of these initiatives are the following: Knowledge Management Capability Assessment (KMCA) [12, 13], the model proposed by Infosys [14], the model proposed by Ehms y Langen [15], and the Pyramid Model [16].

On the other hand, the models that are not CMM-related do not present a common feature, each adopting a particular vision for the elaboration of the model. Examples of these initiatives are: the KMf and KM3 models [17], the KPMG Knowledge Journey [18], the model proposed by Gabor Klimko [18], and the Tata 5iKM3 [20].

Regardless of the model that is followed, the satisfaction of a high KM maturity level by an organization tends to increase its prestige and competitive advantages. It may however be very expensive to evaluate whether or not an organization disposes of the necessary conditions to obtain a given KM level: this implies the repeated services of an auditor before and after the correction of any detected insufficiencies. Also, once the necessary conditions are acquired, the auditor must intervene once more to evaluate the organization at the desired KM level. The entire process may turn out to be very costly.

This paper proposes a Knowledge-Based System (KBS) for the evaluation of an organization at a specific KM maturity level. The application of this system will substantially reduce the need for expensive audits because these will only take place after the system has issued a positive report on compliance with the level—a report that has many possibilities to coincide with the auditor's review but is not necessarily identical, since the auditor may weigh certain aspects that are not considered by the KBS. These differences between expert and KBS can actually be exploited, since the inclusion of new knowledge will lead to the improvement of the system.

We selected the KMCA model, proposed by Uday Kulkarni and Ronald Freeze [12, 13], because its structure and evaluation process are well detailed.

The KMCA model also contemplates six maturity levels. These levels are, ranged from minor to major, the following: difficult, possible, encouraged, enabled, managed, and continuously improved, with a major KMCA level implying more KM quality.

Each maturity level implies the following Knowledge Capability Areas (KCAs): expertise, lessons learned, knowledge documents, and data.

The expertise KCA refers to the management of the knowledge of the organization's individuals that are considered to be experts in a certain domain, and could be seen as the highest level of organizational knowledge. The transfer of this experience takes place through consultation, collaboration, observation, and personal interaction. Managing the availability and location of the experts implies a clear improvement in organizational performance.

The lessons learned KCA refers to the management of the learning obtained from previous experience, both in terms of success and failure.

The knowledge documents represent explicit knowledge in the shape of project reports, technical reports, policies and procedures, publications, diagrams, audio and video files, etc. These documents constitute the knowledge base that can be referred to by workers to increase their understanding of their work, and are rather permanent.

Finally, the data make up the most explicit part of the organizational knowledge. These data are stored in data warehouses and constitute a constant source of knowledge to detect patterns, model business situations, etc. The design of its structure and the description of the own data (metadata) determine the level of functionality of this type of knowledge.

Even though the authors of the KMCA model recognize the existence of a certain overlap between the previously described KCAs, they consider that the differences between them are substantially significant and that the knowledge needs of the organization can be better understood with this distinction.

Each KCA has its own properties, but all imply the same management steps: creation/acquisition, storage/recording, recovery/transfer, and application/reuse. These phases can be applied to both explicit and tacit knowledge. They usually require a technological support and the integration of the KM processes into the normal business processes, as well as an organizational culture that promotes the sharing of knowledge.

Thus, for each maturity level in the KMCA model, the evaluation process implies the evaluation of a set of general objectives and a set of KCA-specific objectives. Each objective is related to specific KM practices, which are represented by concrete questions in a maturity form.

The design of the proposed KBS is detailed in section 2 of this paper, section 3 shortly describes the system implementation, and section 4 sets out the conclusions.

2 Design of the proposed system

The quality of KBS design depends on the knowledge engineer's programming skills, and on his ability to devise, remember, and dynamically update a design specification. This is a difficult task for all but the smallest KBSs.

Difficulties like these can be alleviated by producing representations of the expert's knowledge and of the design specification in the shape of text or diagrams. The best known approach towards the production of such documents is the CommonKADS methodology [21, 22, 23, and 24]. It now is the European de facto standard for knowledge analysis and knowledge-intensive systems development, and it has been adopted as a whole or has been partly incorporated in existing methods by many major companies in Europe, as well as in the US and Japan [21]. We apply CommonKADS to elaborate a list of potential components of the model for the KBS, select the adequate template for the task, and construct the initial domain scheme. The last stage is a complete specification of the knowledge model. The following sections describe how each of these activities was carried out.

2.1 List of potential model components

When we start constructing a KBS we assume that a knowledge-intensive task has been selected, and we have to identify the main knowledge items involved in this task.

Thus, the goal of this identification stage is to survey the knowledge items and prepare them in such a way than they can be used in the next stages. This includes exploring and structuring the information sources for the task and constructing a knowledge items list (the list of potential model componets). Moreover, this activity has to try to pave the way for reusing model components that have already been developed and used elsewhere.

Firstly, we have to stablish the type of the domain: is it a domain with formal knowledge? Is the knowledge in the domain mainly heuristic?, and so on. Then, we have to look for standardized descriptions of the domain or of similar domains. These descriptions can take many forms (thesauri, ontology libraries, reference models, product model libraries, technical reports, etc.).

In this case, the task of the proposed KBS belongs to a highly specialized field, i.c. a concrete and classified theme within Quality Management. It is perhaps for this reason that we dispose of reliable information on how to carry out audits [25]. Consequently, the knowledge of the domain can be said to be formal.

On the one hand, there is evidence of the existence of a commonly accepted structure in the sphere of the KMCA model—shown in figure 1—that represents an initial candidate for the domain model. This structure reflects the existence of six maturity levels and four KCAs. Also, a maturity level requires a certain level of compliance with each KCA, and each KCA contains a series of questions in a maturity form; consequently, the compliance of all of these questions implies the compliance of the KCAs as a whole.

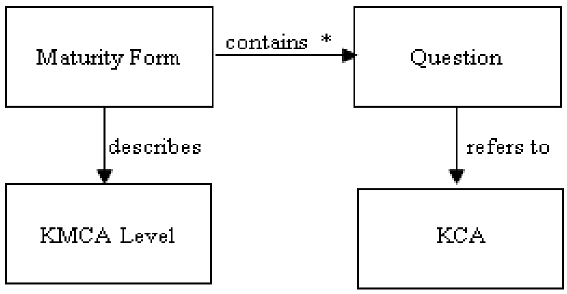

Fig. 1. Initial relationships structure.

On the other hand, it is fundamental to record the performed audits and their results in, for example, a database: when we consult the system with respect to the convenience of an audit of KM maturity level *n*, we must be able to check whether the organization was successfully audited in KM maturity level *n-1*. If the answer is negative, there is no possibility whatsoever to compete for the desired level, because neither in the system, nor in reality, it is good practice to "skip" KM maturity levels. For example, in order to reach level 3 we must previously have obtained level 2.

2.2 Selection of the task template

Several features of the application task can be important in choosing an appropriate task template [22]:

- The nature of the output (a fault category, a decision category, a plan, etc.).
- The nature of the inputs (kind of data available for solving the problem).
- The nature of the system the task is analyzing, modifying or constructing (a human-engineered artifact, a biological system o a phisical process).
- Constraints posed by the task environment (the required certainty of the solution, the costs of observations, etc.).

The final purpose of the proposed KBS is to provide an organization with the possibility to fill out a form for a given KM maturity level and consult the system regarding its viability: *Given the data contained in this form, is it possible for the organization to successfully pass an audit for KM maturity level x?*

In this context, and from the point of view of the task, this is an activity that fits into the category of assessment. These activities are provided with various templates, from which we have selected the one mentioned in [22] and shown in figure 2.

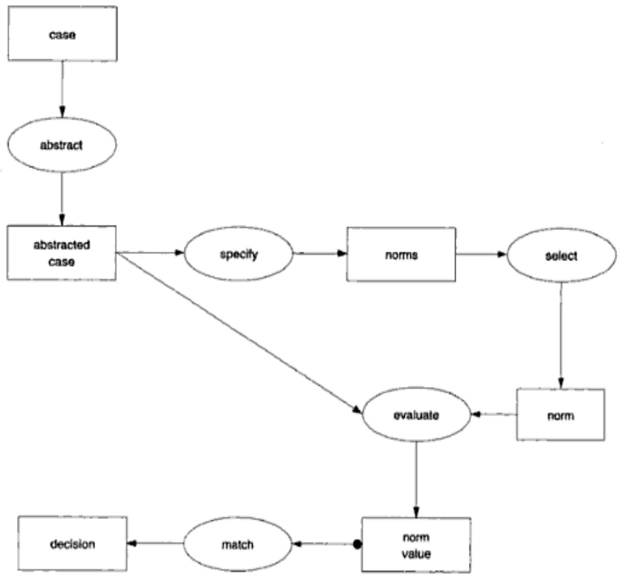

Fig. 2. Inference structure for the assessment method.

The main motive for this choice is that the associated inferential structure matches the purpose of the application. A good technique to establish this adequacy to the problem consists in building an annotated inferential structure in which the dynamic roles are annotated or made to correspond with specific elements of the domain. This inferential structure is shown in figure 3.

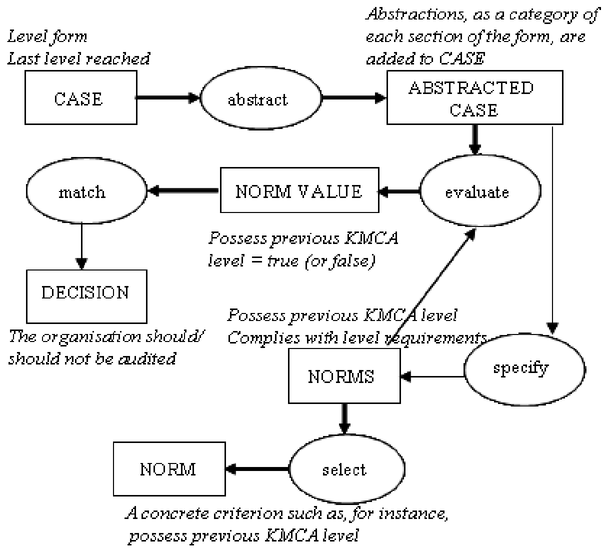

Fig. 3. Annotated inferential structure.

2.3 Construction of the initial domain scheme

As recommended in [22], this activity was carried out in parallel with the previous one. The goal of this activity is to construct an initial data model of the domain independent of the application problem being solved or the task method chosen. Typically, the domain schema of a knowledge-intensive application contains at least two parts:

- Domain-specific conceptualizations. These are the domain structures that we recognize directly in the domain, and that are likely to be present in any application independent of the way in which it is being used.
- Method-specific conceptualizations. A second set of domain constructs is introduced because these are needed to solve a certain problem in a certain way.

Figure 4 shows the domain-specific conceptualisations and figure 5 shows the method-specific conceptualisations.

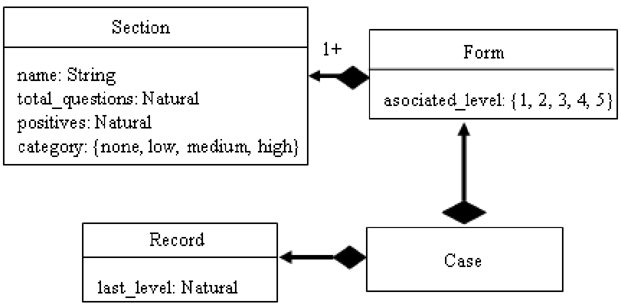

Fig. 4. Domain-specific conceptualisations.

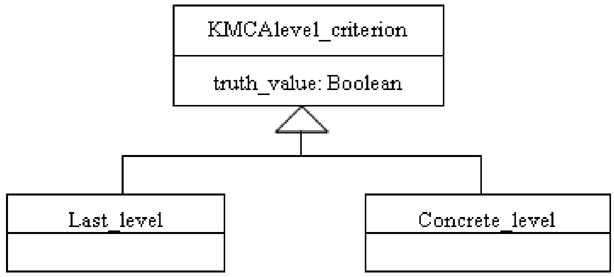

Fig. 5. Method-specific conceptualisations.

We detected two main concept types in the problem domain: Form and Section. We also need some historical information, such as the last KMCA level that was reached. To this effect, we model a concept Record with the attribute that represents this need. The concepts Form and Record constitute the initial reasoning case. A Form refers to a specific KMCA level and consists of a series of sections that each is related to a KCA. This fact is reflected by modeling an aggregation relationship between the concepts Form and Section. The concept Form has an "associated-level" attribute that indicates the KMCA level to which it corresponds. The concept Section presents four attributes: "name", "total-questions", "positives", and "category". The first refers to the name of the section—e.g. Lessons Learned—, the second indicates the total number of questions in the section, the third represents the total number of questions that were answered positively, and the last attribute refers to the organization's level of compliance in the section.

The previously mentioned level of compliance is obtained in function of the attributes "total-questions" and "positives" in the following way:

- If the positive answers represent less than 25% of the total, the level of compliance is considered "none". This means that the organization does not comply with the KCA represented by the section.
- If the positive answers represent between 25-50% of the total, the level of compliance is considered "low".
- If the positive answers represent between 50-75% of the total, the level of compliance is considered "medium".
- If the positive answers represent between 75-100% of the total, the level of compliance is considered "high".

Once it is determined how the domain concepts will be used, we must establish the criteria that will be applied to the data in order to determine the compliance with a given KMCA level. In this concrete case, we considered two different criteria, each with a truth-value attribute that determines whether or not the criterion was fulfilled:

- Last-level: Was the organization successfully audited in the level previous to that at which it aspires? In other words, does it meet the requirement of having been successfully audited at a KMCA level that precedes the desired level?
- Concrete-level: Does the organization meet the specific requirements of the level for which it wants to be audited? If the organization wishes to be successfully audited for a specific KMCA level, it must meet the KCAs at certain rates or levels (many possibilities are accepted).

Finally, we wish to emphasize that the system only offers a positive response if all the criteria present the value "true".

2.4 Complete specification of the knowledge model

Following [22], there are basically two routes for completing the knowledge model once the task template has been chosen and and initial domain schema has been constructed:

- Middle-out. Start with the inference knowledge, and complete the task knowledge and the domain knowledge, including the inference-domain role mapping. This approach is the prefered one, but requires that the task template chosen provide a task descomposition that is detailed enough to act as a good approximation of the inference structure.
- Middle-in. Start in parallel with decomposing the task through consecutive application methods, while at the same time refining the domain knowledge to cope with the domain-knowledge assumptions posed by the methods. The two ends (i.e., task and domain knowledge) meet through the inference-domain mappings. This means we have found the inferences (i.e., the lowest level of functional descomposition). This approach takes more time, but is needed if the task template is still too coarse-grained to act as an inference structure.

As explained before, the activity to be modelled is an instance of the task type assessment. Also, the selected template shows an adequate inferential structure for the purpose of this KBS, in which the inferences present sufficient detail. Therefore, we choose the middle-out route for completing the knowledge model.

The task that must be carried out is decomposed into two subtasks, which means that the "task method" structures the reasoning process in two steps:

- Abstraction: the purpose of this step is to obtain the level of compliance for each section (KCA). As explained above, this level of compliance can be "none", "low", "medium" or "high". The motive for this abstraction is the fact that what matters in a decision is not so much the number of positive answers by the user, but rather the meaning of this number. In other words, the reasoning of an expert auditor will be as follows: "The organization complies with all the sections at a medium level, but the section Knowledge Documents is indispensable (must have a high level of compliance) and I therefore consider that improvements must be made in that area".
- Matching: the abstractions are matched in order to take the final decision on whether or not there is compliance with the established criteria.

Figure 6 shows the template that was chosen for the modelling.

On the other hand, the knowledge scheme that was finally obtained is shown in figure 7. We can observe that the final domain scheme incorporates three rule types:

- "case-abstraction": the abstractions that are required for the application refer to the obtention of the Section compliance level by using the "total-questions" and "positives" attributes as previously mentioned. Even though the abstraction really refers to the Section concept, it is modeled in the Case concept as it possesses an aggregation relationship with Section.
- "form-requirement": this type of rule aims at offering truth values to the norms "Last-level" and "Concrete-level". Their instances therefore indicate the compliance with the previous KMCA level and the acceptable compliance levels of the maturity form's sections for a determined level.
- "level-decision-rule": we need some type of knowledge that refers to the final decision offered by the system to the user. This decision is represented by a "Level-decision" concept with a "value" attribute that indicates whether or not the organization has real possibilities of successfully passing an audit for the desired KMCA level. Also, this type of rule

expresses the relation between the different criteria and the final decision taken by the KBS.

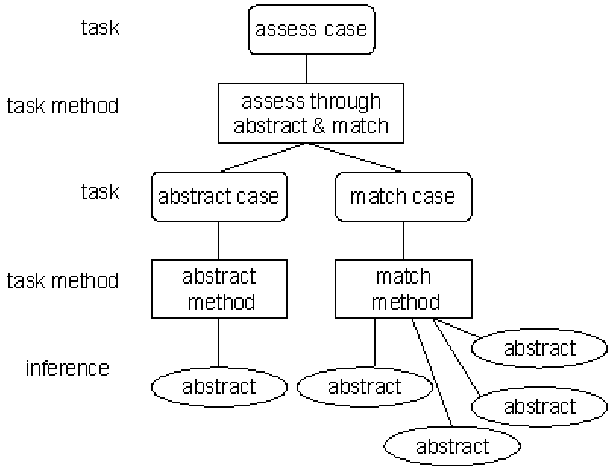

Fig. 6. Decomposition of the task.

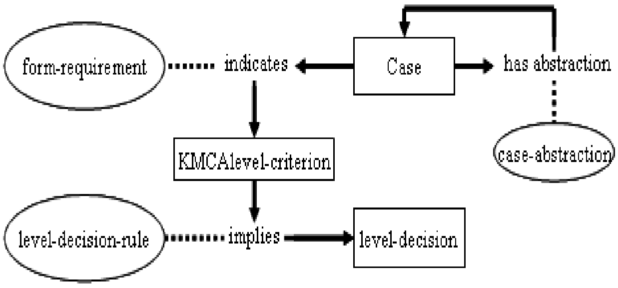

Fig. 7. Final knowledge scheme.

3 Implementation of the proposed system

The system was implemented according to the above design and by means of the Clips tool [26] because it is a development and delivery expert system tool which provides a complete environment for the construction of rule and/or object based expert systems.

In order to provide the application with modularity and simplify the development and depuration processes, we defined the following knowledge bases:

- General: This knowledge base contains all the definitions of classes, objects, and properties.
- Abstract: This knowledge base contains the abstraction rules needed to obtain the compliance level of each KCA. As explained in the knowledge model, each KCA entails a series of questions that the organization must answer. Relevant to the system are not the questions themselves but rather the number of affirmative answers with respect to their total amount. The abstraction of the incidence probability is then calculated.
- Level1, Level2, Level3, Level4, and Level5: This knowledge base contains the rules for the evaluation of the criteria "Last-level" and "Concrete-level". Since level 1 is not a maturity level (it has no KCA compliance), we begin with level 2, i.e. the knowledge base Level1 refers to KMCA level 2.
- Decision: The rules contained in this knowledge base refer to the final assessment decision according to the values of the criteria specified above.

These rules are based on the KMCA instrument proposed by Freeze and Kulkarni [13], which identifies a set of descriptor variables for each capability area. These variables are shown in tables 2, 3, 4 and 5.

Expertise Repository(ies)
Availability of repository(ies)
Accessibility of repository(ies)
Usefulness of repository content
Information about internal & external experts
Search capabilities
Ease of searching
Multiple search criteria
Expert Access/Consulting
Practice of looking for available expertise
Ease of finding experts
Embedded in normal work practices
Expert Profiling & Registration
Existence of a registering and profiling process
Ease to use
Allows self-updating
Managed for consistency
Expertise Taxonomy
Existence of taxonomy
Clarity and standardization
Comprehensiveness
Extensibility
Collaboration Tools
Routineness of use
Ease of use
Access to internal & external experts
Multiple tool set
Communities of Practice
Participation in SIGs
Encouragement for participation
Availability of relevant SIGs
Participation on company time
Financial support for participation

Table 1. Expertise descriptor variables.

Lessons Learned Repository(ies)
Accessibility of repository(ies)
Usefulness of repository content
Search & retrieval capabilities
Ease of searching
Multiple search criteria
Taxonomy
Existence of taxonomy
Clarity and standardization
Comprehensiveness
Capture
Practice of capture
Consolidation and management
Individual and group responsibilities
Existence of a systematic processes
Application/Use
Practice of application/use
Ease of finding relevant lessons
Embedded in normal work practices

Table 3. Lessons Learned descriptor variables.

Knowledge Documents Repository(ies)
Availability of repository(ies)
Accessibility of repository(ies)
Usefulness of repository content
Access to internal & external documents
Supports rich formats
Clarity of meta-data
Taxonomy
Existence of taxonomy
Clarity and standardization
Comprehensiveness
Search & Retrieval
Ease to use
Effectiveness of retrieval system
Multiple search criteria
Categorization
Existence of a categorization process
Ease to use
Embedded in normal work practices
Managed to ensure adherence
Reference & Use
Practice of reference/use
Ease of finding documents

Table 4. Knowledge Documents descriptor variables.

Data Repository(ies)
Availability of repository(ies)
Accessibility of repository(ies)
Currency of data
Level of detail/summarization
Clarity of meta-data
Data Relevance
Timeliness
Periodicity
Completeness
Usefulness of format
Accuracy
Decision Support Tools
Ease to use
Sufficiency

Table 5. Data descriptor variables.

The Clips inference engine is started and the corresponding knowledge bases are loaded. Once the graphic interface is initiated, the inferential process begins. Figure 8 shows an execution example in which a company wishes to be evaluated at KMCA level 4 after having been successfully audited at KMCA level 3, but the organization lacks an acceptable level of compliance with the KCAs. Figure 9 shows an execution example in which a company wishes to be evaluated at KMCA level 3 after having been successfully audited at KMCA level 2, and the organization has an acceptable level of compliance with the KCAs.

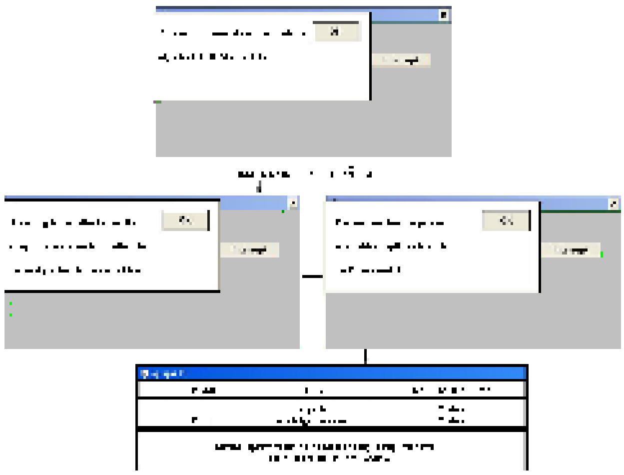

Fig. 8. An execution example

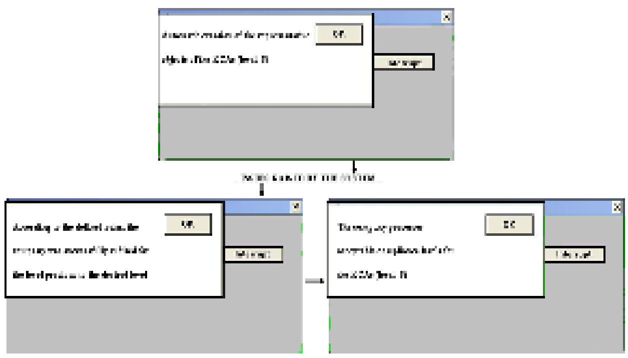

Fig. 9. An execution example (cont.)

4 Conclusion

In the last years, organisations have realised that success depends, more and more, on the knowledge they possess in detriment to other factors such as work or capital. This commonly accepted situation has caused organisations to dedicate numerous efforts to trying to retain and institutionalise the knowledge they possess. The answer to this growing need was the birth of the KM discipline and the development of a new type of systems or programmes: the KM systems or programmes.

With the aim of guarantee the quality of these KM programmes, there is a growing number of KM maturity models, which can be classified into two large groups: CMM-based models and models that are not CMM-related.

Thus, a high quality KM program undoubtedly implies prestige and competitive advantages for an organization. This is precisely the reason why organizations dedicate numerous resources to the adaptation of their processes to the requirements of the KM maturity models. One of these models is the KMCA, and it's the one we selected. However, until it reaches the desired KM maturity level, an organization usually needs to pass a series of audits. This paper proposes a KBS that considerably reduces the economic burden of such audits by limiting their number in function of the compliance with the desired KM level (KMCA level).

Finally, the developed KBS is currently being installed and tested in various companies at A Coruña, Spain, with which the authors have collaborated in previous occasions.

References:

[1] Scorta, L., A Knowledge Management Practice Investigation in Romanian Software Development Organizations, *WSEAS Transactions on Computers*, Vol.8, No.3, 2009, pp. 459-468.

[2] Metaxiotis, K., Ergazakis, K., Psarras, J., Exploring the world of knowledge management: agreements & disagreements in the research community, *WSEAS Transactions on Circuits and Systems*, Vol.3, No.3, 2004, pp. 617-626.

[3] Andrade, J., Ares, J., García, R., Rodríguez, S., Silva, A., Suárez, S, Knowledge Management Systems Development: a Roadmap, *Lecture Notes in Artificial Intelligence*, Vol.2773, 2003, pp. 1008-1015.

[4] Davenport, T.H., Prusak, L, *Working Knowledge: How Organizations Manage What They Know*, Boston, USA: Harvard Business School Press, 2000.

[5] Ares, J., Rodríguez-Patón, A., Suárez, S., *Practical Guidelines for Deploying a Knowledge Management System*, Idea Group Inc., 2008.

[6] Amescua, A., García, J., Sánchez-Segura, M.I., Medina-Domínguez, F., Software process improvement for practitioners based on knowledge management tools, *Proceedings of the WSEAS International Conference on Software Engineering, Parallel and Distributed Systems*, Madrid, Spain, 2006, pp. 24-29.

[7] Muntean, M.I., Târnaveanu, D., Some considerations about collaborative systems supporting knowledge management in organizations, *WSEAS Transactions on Computers*, Vol.8, No.8, 2009, pp. 1378-1387.

[8] Davenport, T., Prusak, L., *Working Knowledge: How Organizations Manage What They Know*, Boston, USA: Hardvard Business School Press, 1998.

[9] Markov, A., A Theory of Human Motivation, *Psychological Review*, Vol.50, 1943, pp. 370-396.

[10] McCollum, W.R., *Process Improvement in Quality Management Systems: A Case Study, Analyzing Carnegie Mellon's Capability Maturity Model (CMM)*, Cheshire, UK: Trafford Publishing, 2006.

[11] Paulk, M.C., A History of the Capability Maturity Model for Software, 2009, URL: http://home.comcast.net/~mark.paulk/papers/p2009c.pdf. Last access: 10/01/2010.

[12] Kulkarni, U., Freeze, R., Development and Validation of a Knowledge Management Capability Assessment Model, *Proceedings of the 25th International Conference on Information Systems*, Virginia, USA, 2004.

[13] Freeze, R., Kulkarni, U., Knowledge Management Capability Assessment: Validating a Knowledge Assets Measurement Instrument, *Proceedings of the 38th Hawaii Intl. Conference on System Sciences*, Hawaii, USA, 2005, URL: http://wpcarey.asu.edu/seid/cabit/upload/Freeze%20Kulkarni%20%20HICSS%202005.pdf. Last access: 10/01/2010.

[14] Kochikar, V.P., The Knowledge Management Maturity Model: A Staged Framework for Leveraging Knowledge, *Infosys Technologies Limited*, 2000. URL: http://www.infotoday.com/KMWorld2000/presentations/kochikar.ppt. Last access: 10/01/2010.

[15] Ehms, K., Langen, M., Holistic Development of KM with KMMM, *Siemens AG/ Corporate Technology Knowledge Management, and Business Transformation*, 2002.
[16] Ying-Hsun, H., Seng-Cho, T.C., On Constructing a management pyramid model, *Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration*, Las Vegas, USA, 2005.
[17] Gallagher, S., Hazlett, S.A., Using the Knowledge Management Maturity Model (KM3) as an Evaluation Tool, *Proceedings of the Conference on Knowledge Management Concepts and Controversies*, Conventry, UK, 2000.
[18] KPMG, KPMG Research Report, 2000. URL: http://www.providersedge.com/docs/km_articles/KPMG_KM_Research_Report_2000.pdf. Last access: 10/01/2010.
[19] Klimko, G., Knowledge Management and maturity models: building Common understanding, *Proceedings of the Second European Conference on Knowledge Management*, Bled, Slovenia, 2001, pp. 269-278.
[20] Santosh, K., Manish, C., 5iKM3 Knowledge Management Maturity Model, 2005. URL: http://www.tcs.com/resources/white_papers/Pages/5iKM3KnowledgeManagementMaturityModel.aspx. Last access: 10/01/2010.
[21] CommonKADS, CommonKADS home page, URL:http://www.commonkads.uva.nl/frameset-commonkads.html. Last access: 10/01/2010
[22] Schreiber, G., de Hoog, R., Akkermans, H., Anjewierden, A., Shadbolt, N., de Velde, W.V., *Knowledge Engineering and Management: The CommonKADS Methodology*, Cambridge, USA: The MIT Press, 2000.
[23] Kingston, J., Designing Knowledge Based Systems: The CommonKADS Design Model, *Knowledge-Based Systems*, Vol.11, No.5-6, 1998, pp. 311-319.
[24] Valente, A., Breuker, J., van de Velde, W., 1998. The CommonKADS Library in Perspective, *International Journal of Human-Computer Studies*, Vol.49, No.4, 1998, pp. 391-416.
[25] Software Engineering Institute home page, URL: http://www.sei.cmu.edu/. Last access: 10/01/2010.
[26] Riley, G., Clips. A Tool for Building Expert Systems, URL: http://clipsrules.sourceforge.net. Last access: 10/01/2010.