Assessing KBS Construction and Verification Techniques for Knowledge Management Systems

Robert Plant
Dept. of CIS
University of Miami
Coral Gables, FL 33124

Todd Gamble
Information Services
WorldCom, Inc.
Tulsa, OK 74117

Rose Gamble
Dept. Math and CS
University of Tulsa
Tulsa, OK 74104

Introduction

The aim of this research note is to identify some critical areas of knowledge management that may benefit from research results in the development and verification of knowledge-based systems (KBSs). While knowledge management efforts are usually much broader and more loosely defined than traditional KBSs, there are many activities that are analogous to those required for KBSs. In particular, we have found that meta-knowledge (knowledge about knowledge) is beneficial to understanding the dynamic aspects of KBSs (e.g., interaction, complexity, completeness, and use of changing heuristics). It is these dynamic aspects of knowledge that play an even larger role in knowledge management systems where organizational process knowledge is often chaotic in nature. The techniques developed for verification and validation of KBSs are applicable to knowledge management systems, although the existence of defined KBS anomalies may have more positive connotations for knowledge management systems.

While KBSs may represent only a portion of the diverse field of knowledge management, they are, arguably, one of the first instances where artificial intelligence techniques were applied to the representation and reuse of knowledge within a corporate setting. For example, during the recent era of business process reengineering (BPR) opportunities arose for organizations to question how and why they did what they did. This new reflective understanding of the organizational processes was accompanied by the awareness of a need to understand how 'experts' within the organization actually solve problems. Once the existence of 'expert' knowledge was identified, questions arose regarding how to capture and reuse that knowledge should the expert be unavailable, i.e., how could the knowledge be a managed as an asset of the corporation. The need to answer this question resulted in the implementation and commercialization of KBSs in the form of 'expert systems'. The resulting KBSs were narrow in their domain of knowledge and usually required strict controls on the quality and representation of the knowledge in order to function reliably and, thus, are only analogous to part of what knowledge management is attempting to address.

We divide the paper into two main sections. The first section discusses analogies between KBS and knowledge management system development. The second section examines the verification criteria resulting from KBS research and its application to knowledge management systems.

System Development

The Importance of Meta-Knowledge

Meta-knowledge (knowledge about knowledge) explains qualities and attributes (e.g., source, method of acquisition, results of previous use, history of changes, etc.) of the knowledge within a system. Most importantly, meta-knowledge provides a *context* for interpreting the knowledge within a system. In the KBS development framework described in the next section, meta-knowledge is used to help KBS designers understand the relationships between stages of KBS development. In order to understand the importance of context and (the user of meta-knowledge) to develop that context, it is useful to look at some of the literature on learning organizations. Many knowledge management efforts are part of a larger learning initiative.

> "We educate smart people for content, without regard for context. Yet in times of rapid change, context is everything: no fact, by itself, is as important as it is when applied and economic, social, environmental and political realities are factored in."[CR95]

If we take "smart people" to mean "experts", this quote helps explain the difference traditional KBSs focused on narrow domains of knowledge expressed in facts and rules and current knowledge management which seeks to include contextual information in the form of meta-knowledge.

A Methodological Development Framework

In order for organizations to create a knowledge management environment, they need to base their self understanding or meta-knowledge upon a solid methodological framework. We use the KBS development methodology previously defined by [PG97] and depicted in the figure below to suggest how the methodology could be amended to further assist the needs of the knowledge management community.

Technologies from Knowledge-Based Systems Development

As suggested earlier, many of the issues now facing the knowledge manager[1] of a corporation have been faced in a similar form by knowledge-based systems practitioners in the past. A large amount of research has been focused upon the development of KBS systems, however, without a connective framework around which the individual specialized aspects of KBS systems research could be placed, the research was difficult to link together in a useful manner. The framework defined by [PR97] proposes the use of a meta-knowledge model to link together the stages of system creation utilizing Newell's Knowledge Level construct [NA82].

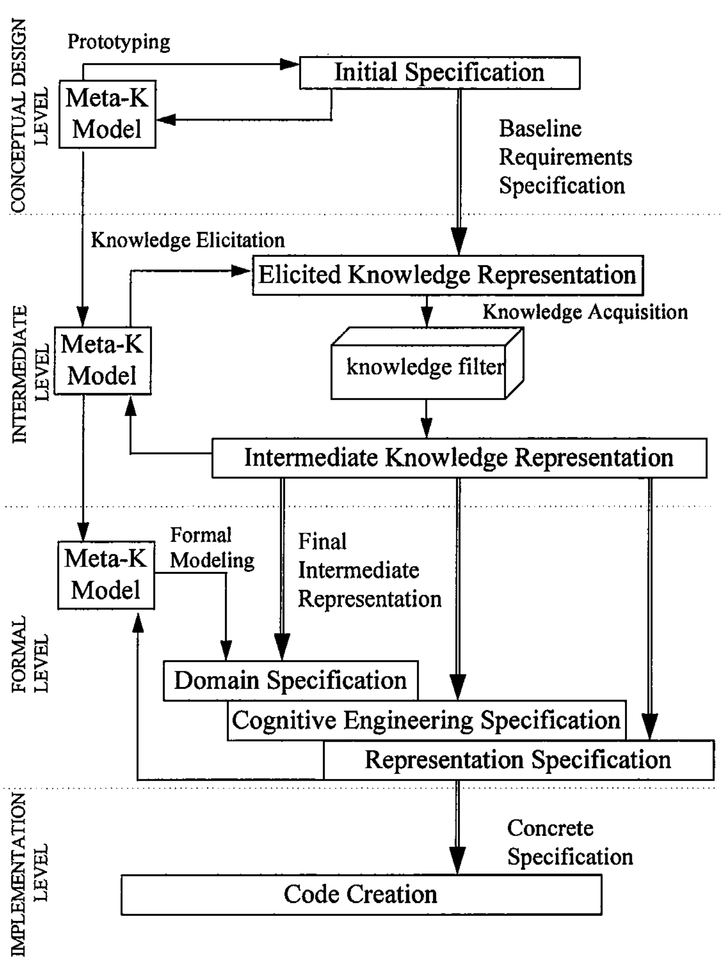
The methodology takes an initial specification of the system to be created and utilizes the meta-knowledge surrounding the project to create a more definitive baseline specification of the system. This specification is then acted upon in the knowledge elicitation phase again utilizing meta-knowledge to drive the process. The difficulty at the elicitation stage is that the knowledge may well be in many formats, making the validation and verification process extremely difficult. Hence the need to create an intermediate representation of the knowledge in a semi-formal notation. This initial representation allows for gross validation issues to be recognized and the elicitation cycle to be repeated until the issue has been clarified. As a critical mass of the system is defined, the information, data, knowledge and meta-knowledge is focused into a formal notation upon which rigorous verification can take place. These are the domain, cognitive, and representation specifications. These separate specification categories allow the knowledge, the interface, and representational issues to be focused upon individually but aligned through the meta-knowledge and the refinement process. Subsequent to this, the knowledge is refined into a coded system.

Application to Knowledge-Management System Concerns

The methodology for KBS design and development can also act as foundation for a methodology for knowledge management system design. The chief component of the methodology is based upon using meta-knowledge as a tracking agent, which is a primary information source in knowledge management [DP98]. Meta-knowledge can be decomposed into two forms *static* and *dynamic*. Static knowledge relates to the structure of the knowledge. Dynamic knowledge relates to the execution of the knowledge. The dynamic knowledge is defined at the object level and contains information about the task domain. The object level itself is described through using a *meta-level* of information (see [PG97] for a detailed discussion).

Knowledge management systems require a refinement of this knowledge model for applicability. The meta-knowledge in our descriptive language is used to represent information regarding the properties of the enterprise that are modeled at the object level. Thus, the object level will describe the knowledge and information that *identifies the knowledge assets a company possesses*, but the meta-level information will answer the questions, such as, where is the knowledge asset, what does it contain, what is its use, what form is it in, and how accessible is it? This meta-level information is ultimately placed in the formal domain specification at the Formal Level of the development methodology above. The representation of the knowledge in the knowledge management system is often in the form of data warehouses, databases and KBSs. Thus, the issue of representational selection is addressed via the representation specification at the Formal Level of the design process.

[1] Many organizations have officially designated an executive position the CKO or Chief Knowledge Officer.



A Development Methodology for Knowledge-based Systems [PG97}

Verification and Validation Concerns

Research in KBS verification and validation has established a set of anomalies whose existence in a KBS could cause problems. As a result, many theories, techniques, and systems have been develop to verify the absence of the anomalies, usually by first detecting the potential occurrences and then by investigating further when the occurrences can cause real problems, e.g., [OO93, PA90, SV96, GL95, PL94].

The anomalies are partitioned across categories of *redundancy, inconsistency,* and *incompleteness.* Redundancy implies that multiple knowledge structures use the same information and lead to the conclusions. Inconsistency implies that multiple knowledge structures use the same information and lead to conflicting conclusions. Incompleteness implies that there is insufficient knowledge to make a conclusion. There are smaller classes of anomalies that are present in each category.

In KBSs, these anomalies can be quite debilitating. While redundancy may not be a problem in a static KBS, imagine a KBS in which a piece of knowledge was found to be incorrect and is then deleted. However, it is not known that this knowledge exists redundantly in the KBS and, therefore, remains though it was believed to be deleted. Inconsistency can cause incorrect or non-deterministic answers depending on the reasoning path taken each time the KBS executes. Incompleteness may result in no available conclusions from the KBS. However, in some cases such anomalies are desired in the KBS. Thus, detection mechanisms usually find potential occurrences and then correct only those that have been investigated further.

For knowledge management systems, these anomalous criteria remain meaningful and generally detectable, though not necessarily by automated means. This is due to the multiple representation paradigms used to configure knowledge in a knowledge management system. However, the response to an anomalous occurrence is not generally analogous to that for a KBS. In this regard, we briefly look at each category of anomalies and discuss them in the context of knowledge management systems.

Redundancy

Within a single KBS, knowledge is represented in a consistent manner. In a knowledge management system, knowledge may be represented in multiple formats. For example, in the field of software architecture [SHG96], knowledge of the architectural style of a software system could be stored in multiple ways: diagrammatically, formal models, and natural language [SG97]. This knowledge is redundant though each format may be needed for a different context. With respect to providing a means of communicating knowledge (a very important knowledge management function), repeated representation and access may be crucial. Thus, redundancy can be a key feature in a knowledge management system in order to verify that there is a shared understanding among the stakeholders of the system.

Inconsistency

Inconsistency in a knowledge management may indicate a conflict or an incorrect input to the system as in KBSs. However, in knowledge management, inconsistency may also indicate innovation or a shift in business models. Thus, positive, not simply non-deterministic, outcomes may arise from inconsistent knowledge.

Incompleteness

While KBS developers strive to overcome incompleteness in the system, it is accepted, and even taken for granted, within a knowledge management system since most organizations are continually learning and changing. Multiple representations, multiple stakeholders, and multiple contributors all indicate that there will be gaps in the complex knowledge the system maintains.

Detection

Further research needs to be performed to determine if there exists a common abstract representation from which to detect potential anomalies in a knowledge management system and to define when there are harmful. Returning to the software architecture example, redundancy as detected is a plus to the system knowledge. Once it is detected, it must be shown that there is consistency across representations of the redundant knowledge. This is very difficult to perform without a stable medium of representation. In addition, detection for KBSs has mainly focused on production rules. However, knowledge for knowledge management systems may be very diverse and for each representational format, different detection mechanisms may be needed.

References

[CR95] Chawla, S., and Renesch, J. (1995), *Learning Organizations,* Productivity Press, 1995.

[DP98] Davenport, T.H., and Prusak, L. (1998), *Working Knowledge: How Organizations Manage What They Know,* Harvard Business School Press, 1998.

[GL95] Gamble, R. and Landauer, C. (1995), Validation and Verification of Knowledge-Based Systems. Workshop Notes IJCAI-95, Montreal, Quebec. Canada.

[NA82] Newell, A., (1982), The knowledge-level, *AI,* 18:87-127.

[OO93] O'Keefe, R. and O'Leary, D.E. (1993), Expert system verification and validation: A survey and tutorial, *Artificial Intelligence Review,* 7:3 - 42.

[PA90] Preece,A.D., Shinghal, R., and Batarek, A. (1992), Verifying expert systems: A logical framework and practical tool, *Expert Systems with Applications*, 5:421-436.

[PG97] Plant, R.T., and Gamble., R. (1997), Using Meta-Knowledge within a Multi-Level Framework for KBS Development, *International Journal of Human-Computer Studies*, 46:523-547.

[PA97] Plant, R.T., and Antoniou, G. (1997), Validation and Verification of Knowledge-Based Systems. Workshop Notes AAAI-97, Boston, MA.

[SHG96] Shaw, M. and Garlan, D. (1996), *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall.

[SG97] Stiger, T. and Gamble, R.F. (1997), Blackboard systems formalized within a software architectural style, with T. Stiger, *IEEE Int'l Conf. on Systems, Man, and Cybernetics,* October.

[SV96] Schmolze, J. and Vermesan, A. (1996), Validation and Verification of Knowledge-Based Systems. Workshop Notes AAAI-96, Portland, OR.