School of Informatics, University of Edinburgh

Centre for Intelligent Systems and their Applications

Ontology, Knowledge Management, Knowledge Engineering and the ACM Classification Scheme

by

John Kingston

Informatics Research Report EDI-INF-RR-0169

School of Informatics **December 2002**
http://www.informatics.ed.ac.uk/

Ontology, Knowledge Management, Knowledge Engineering and the ACM Classification Scheme

John Kingston

Informatics Research Report EDI-INF-RR-0169

SCHOOL *of* INFORMATICS
Centre for Intelligent Systems and their Applications

December 2002

in Proceedings of ES02, the 22nd Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence, Peterhouse College, Cambridge, 10-12 December 2002.

Abstract :

The purpose of this paper is to test the theory of multiple perspectives being necessary for completeness in ontologies by applying it to the task of placing knowledge management and knowledge engineering within the ACM classification scheme. The thesis of this paper is that a multi-perspective analysis of the ACM classification scheme, along with a published extension for AI subjects, should demonstrate some of the principles on which the classification is based, and therefore help in deciding where knowledge management and knowledge engineering (and knowledge acquisition) should appear in the classification. Some implications for ontology building are discussed.

Keywords : Ontology, knowledge management, knowledge engineering, ACM classification

Copyright © 2003 by The University of Edinburgh. All Rights Reserved

The authors and the University of Edinburgh retain the right to reproduce and publish this paper for non-commercial purposes.

Permission is granted for this report to be reproduced by others for non-commercial purposes as long as this copyright notice is reprinted in full in any reproduction. Applications to make other use of the material should be addressed in the first instance to Copyright Permissions, School of Informatics, The University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, Scotland.

Ontology, Knowledge Management, Knowledge Engineering and the ACM Classification Scheme

John Kingston
AIAI, School Of Informatics, University of Edinburgh
Edinburgh, Scotland
J.Kingston@ed.ac.uk

Abstract

The purpose of this paper is to test the theory of multiple perspectives being necessary for completeness in ontologies by applying it to the task of placing "knowledge management" and "knowledge engineering" within the ACM classification scheme. The thesis of this paper is that a multi-perspective analysis of the ACM classification scheme, along with a published extension for AI subjects, should demonstrate some of the principles on which the classification is based, and therefore help in deciding where knowledge management and knowledge engineering (and knowledge acquisition) should appear in the classification. Some implications for ontology building are discussed.

1 Introduction

Much work is being carried out these days on the classification of objects or concepts in a standardised manner; such a classification is often referred to as an *ontology*. Various researchers are promoting different ontologies, approaches to building ontologies, standards for ontologies, and so on. Such work is valuable and worthy of respect, but often a single ontology cannot describe an object or concept fully. It is proposed in [1] (with a case study in [2]) that representing an object or concept completely may require up to six ontologies, covering *who, what, how, where, when and why* perspectives, and furthermore that these perspectives may recur at different levels of abstraction, from an "organisational" level right down to a "system implementation" level. This is referred to as a *multi-perspective modelling* approach. The contents of the "what" perspective on knowledge are typically resources of some kind; the "how" perspective contains methods or techniques; the "who" perspective will typically contain agents; the "where" perspective will demonstrate external connections; the "when" perspective will include control and constraints; and the "why" perspective will include justifications and goals.

The purpose of this paper is to test the theory of multiple perspectives being necessary for completeness in ontologies by applying it to the task of placing "knowledge management" and "knowledge engineering" within the ACM classification scheme. This task arose from a request by the librarian of the Artificial Intelligence library at the University of Edinburgh. For several years, the AI library has been classifying its collection according to the ACM classification scheme, along with an extension to the Artificial Intelligence section of the scheme that was published in the AI magazine in 1985 [3]. However, recent interest in knowledge management from commercial and research organisations, along with a grant from EPSRC to develop a Master's Training Package in Knowledge Management and Knowledge Engineering, has led to an influx of books and other materials on these topics. There is no entry in the current ACM scheme for knowledge management, and although there is an existing category for knowledge engineering in the extended version of the scheme (as a subclass of Learning), the librarian had noticed that books on knowledge engineering were being classified in four different places, which suggests that there may be a problem with the current classification scheme.

The thesis of this paper is that a multi-perspective analysis of the ACM classification scheme and the AI extension should demonstrate some of the principles on which the classification is based, and therefore help in deciding where knowledge management and knowledge engineering should appear in the classification.

2 The ACM Classification Scheme and the Scientific Datalink extension

The ACM classification scheme [4] was first published in 1964, with seven top level topics. In its third revision, produced in 1998, the number of top level categories had increased to 11 (see Table 1), along with major extensions of lower level categories.[1]

A	General literature
B	Hardware
C	Computer systems organisation
D	Software
E	Data
F	Theory of computation
G	Mathematics of computing
H	Information systems
I	Computing methodologies
J	Computer applications
K	Computer milieux (philosophy, legislation, administration)

Table 1: Top level of the 1998 ACM classification scheme

[1] The report [4] accompanying the 1998 classification suggests that another major revision is needed, but because deletion of categories would render historical indexes inaccurate, it was decided that a major revision would be delayed; and in addition, categories that were considered redundant would be "retired" rather than being deleted from the hierarchy.

Artificial Intelligence appears in the ACM classification scheme as a subcategory of one of the newer top level categories, “Computing Methodologies”, alongside “symbolic and algebraic manipulation”, “computer graphics”, “simulation and modelling”, “document and text processing”, and others. The subcategories of AI (apart from General and Miscellaneous, which appear in every list of subcategories) are Applications and Expert Systems; Automatic Programming; Deduction and Theorem Proving; Knowledge Representation Formalisms and Methods; Programming Languages and Software; Learning; Natural Language Processing; Problem Solving, Control methods and Search; Robotics; Vision and Scene Understanding; and Distributed Artificial Intelligence. Each of these has some suggested interest areas (i.e. a partial list of possible subcategories); for Applications and Expert Systems, for example, the current list of interests includes (among others) cartography, games, industrial automation, law, medicine and science, natural language interfaces, mathematical aids and prosthetics. It’s immediately clear to readers familiar with the Artificial Intelligence field that, however valid this classification was when it was developed, it does not reflect the current levels of interest in the field very well: an obvious example is cartography, which is listed as a fourth level classification here, but nowadays would probably not even make it to the fifth level – it might be regarded as a subclass of “Geographical applications” which in turn would be a subclass of “Medicine and Science”. Similarly, it’s hard to believe that a new classification would grant “Distributed Artificial Intelligence” the same level of prominence as “Applications and Expert Systems”. The original classification may have been based on what was known at the time, on the political preferences of the ACM committee[2], or on some other basis. However, this highlights the need to understand the principles on which ontological decisions are based to be noted.

In 1985, David Waltz was invited by Scientific Datalink, a division of Comtex Scientific Corporation to extend the AI classification to account for some of the subdivisions of AI, to aid Comtex in indexing the series of AI memos and reports that they had been gathering. The resulting classification, which has been published by Waltz in AI Magazine [3], retains all of the above top level categories except for “Distributed Artificial Intelligence”, which is replaced by “Specialized AI Architectures”. Two new categories are also added: “Cognitive Modelling and Psychological Studies of Intelligence”, and “Social and Philosophical issues”. The contents of most categories have been significantly expanded: continuing the earlier example, “Applications and Expert Systems” now has 19 subcategories, including the 7 proposed as “interests” by the ACM, and these 19 subcategories have up to 11 sub-sub-categories or even sub-sub-sub-categories. Space prevents the replication of the entire classification here, but four of the nineteen categories are described in detail in Table 2.

---

[2] To illustrate “political preferences”, AIAI helped to carry out a project to merge four ontologies of “scientific knowledge management” (i.e. academics and their publications) prepared by different universities into one “reference ontology” [5]. When the four original ontologies were compared, it was noted that there were many similarities, but if a research group’s own special interest area appeared in the ontology, it was classified at a higher level in its own ontology than in the others’ ontologies.

I.2.1. Applications and Expert Systems	**Subcategories**
1.0 Cartography	
1.1 Games	Chess, Checkers, Backgammon, Bidding Games, Wagering Games, War Games, Other
1.2 Industrial Applications	Automatic Assembly, Parts Handling, Inspection, Welding, Planning for Production, Inventory
1.3 Law	
1.4 Medicine and Science	Medical Applications, Chemical Applications, Biological Applications, Geological Applications
1.5 Natural Language Interfaces	
1.6 Office Automation	
1.7 Military Applications	Autonomous Vehicles, Integration of Information, Decisions Aids, Target Tracking, Communication
1.8 Business and Financial	Tax experts, Investment, Financial Planning, Information Storage and Retrieval
1.9 Natural Language Processing Applications	
1.10 Mathematical Aids	
1.11 Education	Tutoring systems, Intelligent Computer-aided Instruction, Aids to learning Programming, Curriculum Design
1.12 Library Applications	
1.13 Engineering Automation	Computer System Design, VLSI Design Aids, CAD/CAM, Programming Aids
1.14 System Troubleshooting	
1.15 Expert Systems	Expert System Languages and Aids for Building Expert Systems, Acquisition of Expert Knowledge, Plausible Reasoning, Representation of Expert Knowledge, Generation of Explanations, Expert Systems based on Simulation and Deep Models, User Interfaces for Expert Systems, Validation of Expert Systems
1.16 Prosthetics	
1.17 Aviation Applications	
1.18 Applications, Other	
I.2.4 Knowledge Representation	
4.0 Frames and Scripts	Defaults, Stereotypes and Prototypes, Generation of Expectations, Frame Languages, Frame-Driven Systems, Inheritance Hierarchy
4.1 Predicate Logic	First Order Predicate Calculus, Skolem Functions, Second Order Logic, Modal Logics, Fuzzy Logic
4.2 Relational Systems	Relational Data Bases, Associative Memory
4.3 Representation Languages	
4.4 Representations (Procedural and Rule-Based)	Production Rule Systems, Knowledge Bases
4.5 Semantic Networks	
4.6 Connectionist Systems	
4.7 Multiple Agent/Actor Systems	
4.8 Constraints	

4.9 Discrimination Trees and Networks	
4.10 Belief Models	
4.11 Representation of the Physical World	
4.12 Representation of Natural Language Semantics	
I.2.6 Learning	
6.0 Analogies	Geometric Analogies, Natural Language Analogies, Structural Analogies, Functional Analogies
6.1 Concept learning	Near-Miss Analysis, Version Spaces, Schema Acquisition and Generalisation, Learning of Heuristics, Credit and Blame Assignment, Conceptual Clustering
6.2 Induction	Statistical Methods, Inductive Inference
6.3 Knowledge Acquisition	Advice Taking and Learning by Being Told, Learning from Examples, Learning by Observation, Learning from Experience, Learning by Discovery
6.4 Knowledge Engineering	Dialogues with Experts, Knowledge Base Stability, Knowledge Base Consistency
6.5 Language Acquisition	Acquisition of Grammar, Learning of Concepts through Language
6.6 Parameter Learning	
6.7 Associative Learning	
6.8 Learning of Skills	
6.9 Developmental and Incremental Learning	
6.10 Evolutionary Models for Learning	
I.2.8 Problem Solving, Control methods and Search	
8.0 Backtracking	
8.1 Dynamic Programming	
8.2 Graph and Tree Search Strategies	Depth first, Breadth first, Best first, Branch & Bound, Hill Climbing, Minimax, Alpha-Beta, A*, Beam, Dependency-Directed Backtracking, Constraint Propagation, Relaxation Methods, Marker Passing, Bidirectional, Data-Driven/Top-Down
8.3 Heuristic Methods	Nature of Heuristics, Heuristic Control of Search, Strategies, Default Reasoning, Closed World Heuristics, Induction and Evaluation of Heuristics, Qualitative Reasoning and Envisionment
8.4 Plan Execution, Formation, Generation	Means-End Analysis, Forward Chaining, Backward Chaining, Weak methods, Generate and Test, Hierarchical Planning, Metaplanning and Multiple Goals, Plan Verification, Plan Modification
8.5 Matching	

Table 2: Part of the Scientific Datalink AI classification scheme

3 Dimensions of classification: classes, subclasses and multi-perspective modelling

The ACM classification scheme is considered to be a four-level, hierarchical taxonomy. A "taxonomy" is defined in Merriam-Webster's dictionary as "a classification, especially an orderly classification of plants and animals according to their presumed natural relationships". Taxonomies are typically used to represent one class of objects or concepts and its sub-types; that is, objects/concepts that possess all the defining features[3] of the higher level object/concept plus a couple of extra features. A 'true' taxonomy therefore includes only one relationship between objects or concepts; one object/concept is a subclass (or "a kind of") the other.[4]

However, when ontologies are built to represent the relationships between tasks, activities, philosophies, or other conceptual entities, it's often difficult to connect them all using only subclass relationships; maybe there are no obvious taxonomic groupings, or maybe there is a more obvious grouping according to function, form, role or relevance. An example of a "more obvious" grouping can be found in vegetable classification; while it might possibly be helpful to know that the Linnaean classification of (most) tomatoes places them alongside aubergines and potatoes in the Nightshade genus of the Potato family, many gardeners would probably prefer to see tomatoes classified alongside other vegetables that grow on vines, vegetables that grow in greenhouses, or even vegetables that are served in salads. An example of "no obvious groupings" can be found by looking at cars. Possible classifications include "saloon", "hatchback", "sports car", etc (based largely on form, but also on role) or "petrol engine cars", "diesel engine cars" and "alternative fuel engine cars" (based on function), but such subdivisions seem less "natural" than the higher level classes – and yet taxonomies are supposed to be based on "presumed natural relationships".

In fact, the whole issue of "natural" versus "artificial" classification has been a major subject of academic debate. A good summary is produced by Wilkins [6] who argues that "all classifications are artificial, but some have a degree of naturalness about them" and quotes R.G. Millikan who proposes that a "natural" concept can be determined by making a historical investigation of how an object and its name came about, and then determining what the name refers to today in most cases.[5] The

---

[3] There is much debate in psychological circles about what constitutes a "defining feature". Interested readers might look at the work of Rosch on "typicality" [7].

[4] There is also a variant of 'subclass' – 'instance-of' – that allows for individual members of classes; so an object can be an instance of a class. Strictly speaking, therefore, a taxonomy allows two types of relationship between objects and concepts.

[5] This is a highly simplified summary; there is an entire journal devoted to classification. Wilkins' complete summary quotation is: "All classifications are artificial, but some have a degree of naturalness about them. Natural classifications are the result of a refinement of the intension of terms based on a very broad and generally culture-neutral set of observations. Species names, indeed all taxa names, are terms with a proper function assigned by the history of their use, and which may change as new evidence is arrived at."

practical result of these "artificial" distinctions is that taxonomies are sometimes based on relationships other than 'subclass'. Common ones are 'part of', 'causes/produces', and 'has property'[6]. In the next section, an analysis of the ACM classification will be carried out to determine what relationships are actually used.

4 The ACM Classification scheme: analysis

The ACM classification covers several of the multiple perspectives. The perspectives covered include "what" is needed for a computer system (hardware and software), "how" to build a computer system (techniques), and "why" systems are built (computing milieux). The categories also cover different levels of abstraction: some categories consider the contents of the computer itself (hardware, software, computer systems organisation, data, information systems) while other categories consider the computer as a single concept in the context of applications (computing methodologies, computer applications, computing milieux). There's also a third level of detail to be found in the two theoretical categories (*Theory of Computation* and *Mathematics of Computing*) which provide the foundational techniques for computer systems organisation, data and information systems. See Table 3 for a summary.

	What	**How**	**Why**	**When**	**Where**	**Who**
Computer applic-ations	Computer Applic-ations	Computing Method-ologies	Computer milieux			
What goes inside a computer	Hardware, Software	Computer Systems Organisation, Data, Information Systems				
Theoret-ical level		Theory of Computation, Mathematics of Computing				

Table 3: Top level categories from the ACM scheme, classified according to multi-perspective modelling

This organisation is broadly mirrored in the organisation of some of the second level categories in the ACM classification scheme. For example, the subclasses of *Computer Systems Organisation* are *Processor Architectures* and *Computer-Communication Networks* (two disjoint components that are necessary for a functioning hardware system, aka *Hardware* and *Software* at the top level); while *Special Purpose and Application Based Systems* and *Computer systems*

[6] Each of these relationships can be broken down into a number of distinct relationships, but this level of detail is beyond the scope of this paper. For an example, see [8] on the breakdown of 'part of'.

implementation look at the "what" and "how" perspectives on hardware construction "applications". There's also a subcategory for *Performance of systems*, which probably falls under the "when" perspective.

The subclasses of Information Systems, Data and Software all use a similar multi-perspective classification scheme. Not all of the second level categories and their decompositions fit neatly into this multi-perspective framework, however. The subdivisions of *Computer Applications* appear to be closer to a taxonomy, in that their second level breakdown consists of different areas of study or different disciplines which reads like a list of university faculties (*Administrative data processing, Physical sciences and engineering, Life and medical sciences, Social and behavioural sciences, Arts and Humanities).* While disciplines are not strictly speaking subclasses of "computer applications", they do (or should) form a single coherent subclass of a (hypothetical) taxonomy of knowledge.[7] The two top-level categories with a theoretical leaning also have sub-categories that reflect different areas of study in the disciplines of (applied) mathematics and (applied) logic.

A third approach is found in the *Hardware* category; its subcategories name different areas of hardware design (*Control structures, Arithmetic and logic structures, Memory structures, Input/Output and Data Communications, Register-transfer-level implementation*, *Logic Design* and *Integrated Circuits*), each of which includes the same small set of sub-sub-categories: *Design Styles*, *Design Aids*, and (until it became a separate category in the 1998 classification), *Performance and reliability*. It seems, therefore, that the Hardware category is decomposed into its second level using the 'part of' relation instead of the 'subclass' relation (i.e. each subcategory is a "part of" the hardware of a computer system rather than a subclass) while a multi-perspective approach is used at the third level, which explains the recurrence of the same subcategories at this level.

5 The Scientific Datalink AI extension: analysis

As with the ACM classification, each of the four categories of the Scientific Datalink AI classification (as reproduced in Table 2) can be broken down into subgroups.

- *Applications and Expert Systems* has nineteen subcategories, seven of which are drawn from the "interests" in the ACM classification scheme. Most of these are concerned with different domains in which expert systems have been applied (similar to the ACM's taxonomic breakdown of Computer Systems Applications into different disciplines), but I.2.1.15 ("Expert Systems") and I.2.1.5 ("Natural Language Interfaces") are more concerned with techniques for expert system construction, and I.2.1.14 ("System Troubleshooting") focuses on a particular task rather than on a domain. The distinction between tasks and domains, which

[7] If the subcategories were relabelled "Applications in <Discipline>" rather than just <Discipline>", the taxonomic connection would be much clearer.

is a key tenet of the CommonKADS methodology for knowledge engineering [9], corresponds to the distinction between “how” and “what” in multi-perspective modelling.

- Most of the subcategories of *Knowledge Representation* are concerned with different knowledge representation formalisms – the “what” of knowledge representation. Frames and Scripts, Predicate Logic, Procedural & Rule-based Representations, Semantic Networks, Constraints and Connectionist Systems all fall into this category. The odd ones out are Representation of the Physical World and Representation of Natural Language Semantics; while these have some correlation with representation formalisms (e.g. simulation models with Representation of the Physical World), these two categories are primarily concerned with knowledge representation as a task rather than a formalism -- i.e. with “how” rather than “what”.
- Several subcategories of *Learning* deal with different methods of learning (by analogy; induction; associative learning), others deal with subjects to be learned (Concept learning; Language Acquisition; Learning of Skills). So here there is a multi-perspective decomposition; some subclasses represent “what” subcategories while others represent “how”. And then there’s Knowledge Acquisition and Knowledge Engineering. Knowledge Acquisition is apparently categorised under “learning” because its subcategories include learning from examples (i.e. induction), learning by observation, learning from experience and learning by discovery. Yet several popular knowledge acquisition techniques are not covered here at all – and while there is a category named “Acquisition of Expert Knowledge” (I.2.1.15.1) two levels down from “Applications and Expert Systems”, the popular techniques are classified in various different places rather than being collected together in I.2.1.15.1. Protocol analysis, for example, is categorised under I.2.11 Cognitive Modelling and Psychological Studies of Intelligence, while the analysis of interview transcripts is most closely covered under Dialogues with Experts, which is considered to be one of only three subcategories of Knowledge Engineering. The reader is left with a strong feeling that Knowledge Acquisition and Knowledge Engineering are underspecified, incomplete, and (possibly as a result) misclassified.
- The final category considered here, *Problem Solving, Control Methods and Search* seems to be something of a catch-all category for methods of controlling inference in AI programs. It has six subcategories, two of which are (unsurprisingly) *Heuristic Methods* and *Graph and Tree Search Strategies*. It also has categories for *Backtracking, Dynamic Programming, and Matching*, which are concerned with the implementation of rule-based systems, and finally a category for *Plan Execution, Formation and Generation*. Control knowledge is slightly difficult to categorise within a multi-perspective framework. In theory, it should be “meta-how” knowledge (i.e. knowledge about the process of controlling processes); in practice, it often includes information about the ordering or processes and the timing of key inputs and outputs to a process, and thus consists of “when” knowledge. This is particularly true of knowledge about planning.

To summarise: Scientific Datalink's AI extension to the ACM classification seems to stick with a formula where formalisms/resources ("what" knowledge) are mixed with methods/techniques ("how" knowledge) to generate subcategories. A taxonomic breakdown is also used (for Applications).

6 Correct classification of Knowledge Management, Knowledge Engineering and Knowledge Acquisition

Having carried out this detailed analysis, it is time to use the principles identified to meet the original goal of this paper: to determine where Knowledge Management and Knowledge Engineering should be classified. Knowledge Acquisition will be considered too.

6.1 Correct classification of Knowledge Engineering

Knowledge Engineering has been variously classified as "the design and development of knowledge based systems"; "application of logic and ontology to the task of building computable models of some domain for some purpose"; "[the study of] the development of information systems in which knowledge and reasoning play pivotal roles"; and "[a] scientific methodology to analyze and engineer knowledge". Using the classifications identified earlier, it's clear that knowledge engineering is primarily application-focused (as opposed to concerned with the internal function of knowledge based systems or theoretical principles of knowledge); and that it focuses on the task of system development (i.e., "how" knowledge). From this analysis, the following classifications of *Knowledge Engineering* are possible:

- Knowledge Engineering could be a subclass of *I.2.1 Applications and Expert Systems*. Unfortunately, *Applications and Expert Systems* uses a largely taxonomic breakdown; but there are two subcategories of *Applications and Expert Systems* that are concerned with techniques for expert system construction. These do not fit well with in the taxonomic breakdown of I.2.1, but would be appropriate siblings for Knowledge Engineering.
- Knowledge Engineering could sit alongside Software Engineering as a subcategory of *D. Software* in the ACM classification. The primary objections to this are the "political" ones – there's much more interest and activity in Software Engineering than in Knowledge Engineering, which makes it difficult to place them at the same level.
- Knowledge Engineering could be a subcategory of *D.2 Software Engineering*. This is probably the most "principled" place to put it, since knowledge engineering is indeed a subcategory of software engineering – it is software engineering for a specialised type of software system. However, this conflicts with the current basis of decomposition of Software Engineering, which is by subtasks rather than a "taxonomy" of types of software.

- Knowledge Engineering could appear alongside *Representation of the Physical World* and *Representation of Natural Language Semantics* as a "how" category under *I.2.4 Knowledge Representation* in the AI extension. The difficulty with this is that the focus of Knowledge Representation is very much on the internals of a knowledge based system, whereas the focus of Knowledge Engineering is on applications, so there is a clash in levels of abstraction.
- Finally, Knowledge Engineering could be left in its current location as a subcategory of *I.2.6 Learning*. This is probably the worst option of all, since knowledge engineering techniques (with accompanying knowledge models) are only appropriate for software that *doesn't* rely on learning as its primary input method, since it's hard to analyse knowledge that has not yet been learned.

In summary, there is no ideal location for Knowledge Engineering in the ACM or Scientific Datalink hierarchies. Since a proposal is needed, a "tie-breaker" can be found in the current subcategory *I.2.1.15 Expert Systems* of *I.2.1.Applications and Expert Systems*. This subcategory actually has a number of knowledge engineering subtasks as its subcategories already. For the sake of backward compatibility, therefore, I.2.1.15 should be left in its current position in the hierarchy, but be renamed to "Expert Systems and Knowledge Engineering".

6.2 Correct classification of Knowledge Acquisition

Once the classification of Knowledge Engineering has been decided, the correct classification of Knowledge Acquisition is fairly easy to determine, for Knowledge Acquisition is a subtask of Knowledge Engineering. Indeed, there is already a category I.2.15.1 named "Acquisition of Expert Knowledge". The only difficulty lies in determining where to classify those topics that are currently subclasses of *I.2.6.3 Learning: Knowledge Acquisition*. Since the Learning section needs to revised anyway to take account of (a) the removal of Knowledge Engineering and (b) the presence of Induction but the absence of two related technologies, Case Based Reasoning and Neural Networks[8], it is proposed that the subcategories of I.2.6.3 are either transferred to other categories under Learning (for example, *I.2.6.3.1, Learning from Examples*, would be appropriate for this) or moved to *I.2.1.15.1, Acquisition of Expert Knowledge.*

6.3 Correct Classification of Knowledge Management

Deciding where to classify knowledge management is difficult because there is considerable disagreement about the best approach to knowledge management. A good working definition of knowledge management would be "the deliberate design of artifacts with the intent to improve the use of knowledge within an organisation",

[8] There are existing Scientific Datalink categories for Connectionist systems under Knowledge Representation, and Connectionist Architectures under *I.2.12 Specialised AI Architectures*, but there is no explicit category for "how" to build neural networks. There is so much work on neural networks these days that it probably deserves its own separate category.

but a range of artifacts have been suggested, from knowledge based systems (thus considering knowledge management as an early stage in knowledge engineering) through to communication forums (considering knowledge management as a process of community interaction in which knowledge-based technology has no part to play). A good survey is given by Binney [10] in which he identifies a “KM spectrum” where knowledge management activities are classified according to their overall goal. Applications that embed knowledge in organisational transactions lie at the “technology-focused” end of the spectrum whereas applications that support innovation and creation of new knowledge lie at the “community-focused” end of the spectrum. Between these two extremes can be found “analytical KM” (the use of knowledge to interpret vast amounts of material); “asset management” KM; “process-based” KM (the codification and improvement of organisational processes); and “developmental” KM (increasing the competencies or capabilities of an organisation’s knowledge workers).

KM is therefore generally application-focused; it can be focused on “what”, “how”, “who” or even “why” depending on the KM approach that is taken; and Binney’s decomposition of KM is focused on “how” a particular goal should be achieved. From this analysis, options for classification of Knowledge Management would be:

- As a subclass of *I.2.1.15 Applications and Expert Systems*, alongside Knowledge Engineering;
- As a subclass of *I.2.4 Knowledge Representation*; however, the arguments against this are the same ones that applied to Knowledge Engineering;
- As a subclass of *I.2.13 Social and Philosophical Issues [in Artificial Intelligence]*. This, however, is more of a theoretical perspective while Knowledge Management is more focused on applications;
- As a subclass of *H.4 Information Systems* in the ACM classification scheme. This removes the commitment that a KM system must be knowledge-based in some fashion, and thus encompasses more of the various KM approaches than would otherwise be the case, but it’s debatable whether or not Knowledge Management should appear at the same level as Database Management – for despite the similarity in terminology, these are really quite different tasks;
- As a subclass of *H.4.1 Office Automation* underneath *H.4 Information Systems*. H.4.1 already contains a category for Workflow management, which is a key enabling technology for process-based KM, and a category for Groupware;
- As a subclass of *H.4.2 Types of Systems* underneath *H.4 Information Systems*. This category currently includes “Decision support systems (e.g. MIS)” and “Logistics”, both of which are reasonably application-focused and also focus on “how” tasks are done.

It seems that there are advantages in taking “Knowledge Management” outside the Artificial Intelligence classification and using the Information Systems classification instead, since some knowledge management approaches are based on software that is not knowledge based. The final recommendation is that Knowledge Management should be a subclass of *H.4.2 Types of [Information] Systems*, since it fits better alongside other types of systems (decision support systems and logistics) than alongside its own enabling technologies (workflow systems and groupware). A new category is therefore proposed, to be labelled *H.4.2.3 Knowledge Management.*

7 Discussion

It has been shown that the ACM classification, and Scientific Datalink's extension, are based on two or three different structuring principles: sometimes taxonomic, sometimes based on "what" knowledge, (which implies that the subcategory is something that is used for, or produced by the top level category; it is a resource in the most general sense of the word), and sometimes based on "how" knowledge – i.e. techniques for, or methods to achieve the top level category. In addition, the Hardware category has a 'part of' decomposition, and some political considerations come into play as well.

What does this tell us about the ACM classification, about multi-perspective modelling, and about ontologies in general? It tells us that if an ontology tries to use "natural" categories, then it will almost certainly be developed using multiple perspectives; so the original thesis of this paper, that multiple ontologies from different perspectives are needed for completeness, is borne out. However, the "what" and "how" perspectives are much more common than the "who", "when", "where" and "why", so it seems that while six ontologies from different perspectives may be necessary, two – with appropriate attention to whether the ontology is focused on theoretical principles, system internals, or applications -- will often be sufficient.

It also tells us that "political" considerations – the level of interest in a subject – have considerable weight when determining the level of various categories in the ontology. The underlying message of this is that there is no canonical way of determining when a set of subcategories is complete – or at least, no way that is sufficiently widely accepted to override political concerns. Some guidance on category completeness may be available from other research; to give an example, "System Troubleshooting" has been identified as the only subcategory of *I.2.1.15 Applications and Expert Systems* that represents an application-focused task. However, a set of "knowledge based tasks" has been proposed by the CommonKADS methodology [11], and one of them (diagnosis) can be instantiated to "troubleshooting". This implies that all the other knowledge based tasks should be eligible, or even expected to make an appearance in I.2.1.15; examples might be "artifact design", "system monitoring", and "selection/ assessment". But this set of tasks is not theoretically proven to be complete; in fact, the original author of this set of tasks has since revised his opinions and proposed that the tasks above are actually composed from a smaller set of five or six "primitive" tasks [12]. So while published sets of categories such as this can be pragmatically useful to ontology developers, they rarely actually solve the problem of canonically determining all possible members of a category.

The ACM classification scheme itself, along with its AI extension, is detailed, widely accepted, and reasonably principled, and so should continue to be used. Some revisions are needed, though (especially under I.2.6 Learning in the AI extension), and it is worth questioning why *Hardware* uses a different decomposition principle from the rest of the scheme: is this an artifact of political lobbying, or is there a "natural" principle here that could be extended to other areas of the classification?

Finally, the new classifications proposed by this paper have classified Knowledge Engineering and Knowledge Management very differently. This raises the issue of the purpose of a classification: should it be carried out according to ontological principles for robustness, or should it be organised to place relevant subjects close to others, to facilitate serendipitous browsing? The case of knowledge management is a difficult one because there are different opinions about it – some books on knowledge management will draw heavily on techniques from knowledge engineering and will serve as useful precursors to knowledge engineering projects, while other books will have little or no relevance to knowledge engineering. An intriguing alternative to the ontological approach would be to use learning techniques to create an entirely new classification scheme based on cluster analysis (using references, keywords, or other criteria); an examination of this approach is suggested for future research.

References

1. Kingston J.K.C., *Multi-Perspective Modelling: A Framework for Re-usable Knowledge*. Forthcoming.
2. Kingston J.K.C. and Macintosh A.L., Knowledge Management through Multi-Perspective Modelling: Representing and Distributing Organizational Memory. Proceedings of ES '99, Cambridge, 1999.
3. Waltz D.L., Scientific Datalink's Artificial Intelligence Classification Scheme. The AI Magazine, Spring 1985, pp.58-63.
4. N. Coulter et al, Report of the CCS Update Committee. http://www.acm.org/class/1998/ccsup.pdf, 1998.
5. Kingston J.K.C., Developing a Reference Ontology for Scientific Knowledge Management. Proceedings of AAAI-02 Workshop on Ontologies and the Semantic Web, AAAI-02, Edmonton, Canada, 29 July 2002.
6. Wilkins J., Natural and Artificial Classification, http://www.users.bigpond.com/thewilkins/papers/artifnat.html, 1997.
7. Rosch E., Natural categories. Cognitive Psychology 4: 328 – 350, 1973.
8. Artale A., Franconi E., Guarino N. & Pazzi, L. Part-Whole Relations in Object-Centered Systems: an Overview. Data and Knowledge Engineering, 20(3): 347-383. 1996.
9. Schreiber A.Th., Akkermans J.M. et al, Engineering and Managing Knowledge: The CommonKADS Methodology. University of Amsterdam, Amsterdam, 1998.
10. Binney D.,The knowledge management spectrum – understanding the KM landscape. Journal of Knowledge Management, 2001, 5(1):33-42.
11. Breuker, J.A., Model Driven Knowledge Acquisition. ESPRIT project 1098 KADS, Deliverable, A1, 1987.
12. Breuker J.A., Problems in Indexing Problem-Solving Methods, Proceedings of the Workshop on Problem Solving Methods, IJCAI-97, Nagoya, Japan, August 1997.