Using Ontologies for Knowledge Management: An Information Systems Perspective

Igor Jurisica, John Mylopoulos, Eric Yu
University of Toronto, Toronto, Ontario, Canada

Abstract

Knowledge management research focuses on the development of concepts, methods, and tools supporting the management of human knowledge. The main objective of this paper is to survey some of the basic concepts that have been used in computer science for the representation of knowledge and summarize some of their advantages and drawbacks. A secondary objective is to relate these techniques to information sciences theory and practice.

The survey classifies the concepts used for knowledge representation into four broad ontological categories. *Static ontology* describes static aspects of the world, i.e., what things exist, their attributes and relationships. A *dynamic ontology*, on the other hand, describes the changing aspects of the world in terms of states, state transitions and processes. *Intentional ontology* encompasses the world of things agents believe in, want, prove or disprove, and argue about. *Social ontology* covers social settings, agents, positions, roles, authority, permanent organizational structures or shifting networks of alliances and interdependencies.

INTRODUCTION

Knowledge management is concerned with the representation, organization, acquisition, creation, usage, and evolution of knowledge in its many forms. To build effective technologies for knowledge management, we need to further our understanding of how individuals, groups and organizations use knowledge. Given that more and more knowledge is becoming encoded in computer-readable forms, we also need to build tools which can effectively search through databases, files, web sites and the like, to extract information, capture its meaning, organize it and make it available. This paper focuses on the concepts used in computer-based information systems to exploit the meaning of information.

Information science as it exists today already provides many of the important foundations for supporting knowledge management. The documentation tradition has a long history of developing methods and practices for organizing the vast expanses of human knowledge for access by various kinds of users. The computational side of information science has developed powerful techniques for retrieving documents through different forms of computer-based processing and search (Buckland, 1999). Information science has also been building on the technologies of information systems to manage the vast amounts of information – initially for catalogues and bibliographic information, then full-text documents, and recently networked and multimedia information bases. Nevertheless, many significant challenges remain.

The information science field has historically focused on the “document” as the primary unit of information. Documents have traditionally been paper-based, and consisted primarily of published books and articles. Their contents are individually meaningful, at least at a literal, surface level. Deeper meanings however do require interpretation in relation to connected documents and cultural contexts. These connections are relatively sparse (e.g., a few dozen references in an academic article) and have little built-in semantics (e.g., a reference simply leads you to another document, much like the untyped hypertext links that predominate on today’s “World Wide Web”). Documents are fairly stable, and new ones take considerable time and effort to create. Document content is primarily read, interpreted and acted on by humans.

The electronic and digital media have changed all that. “Documents” can now be arbitrarily large, as they can be composites of volumes or even libraries of material. More importantly, they can be arbitrarily small – as paragraphs, text fragments, pieces of data, video or audio clips, etc. They are documents not so much in the commonsense usage of the term, but rather logically identifiable and locatable packages of information. This change in the granularity of information units increases the number of units that needs to be managed by many orders of magnitude. Just consider the number of email messages that are sent each day, or the number of post-it notes written. They tend to be much more densely connected, referring to each other in multiple ways. They also change rapidly. Data is constantly updated, documents are created and revised, post-it notes can be detached, re-attached in a different context, or discarded (analogous to the electronic cut-and-paste). Documents can even be active, with embedded software code (e.g., applets and software agents) that exhibit dynamic or even self-activating behavior. Today’s knowledge work relies heavily on electronic media. The move towards knowledge management therefore accelerates the need for information science to deal with this new, much more demanding notion of a “document”.

In contrast, the field of information systems has historically started off from the other end of the spectrum. Information comes in small chunks – e.g., bank account balances, ticket reservations, etc. Such information can change quickly and frequently, so the management of dynamic information has always been fundamental to information systems. Information items usually need to be interpreted in relation to other items, e.g., ‘London’ by itself on a ticket is quite meaningless unless you know that it is a departure or destination city, on what date, what airline, for which passenger, etc. These relationships need to be formally defined so that the network of connected information can be navigated and operated on by automated procedures, in order to produce a ticket within seconds. Now that information processing at electronic speeds has become commonplace, people have come to expect equally powerful technologies for managing much more complex knowledge structures. As in the case of information science, some foundations have been laid in the information systems area for managing knowledge, but there are considerable challenges too. A central issue lies in how meaning is exploited in information systems to produce computational results.

EXPLOITING MEANING IN INFORMATION SYSTEMS

Interestingly, within the field of computing science, there has in fact been a gradual movement towards what one might call “knowledge orientation”. This has been taking place over the past 15 to 20 years, long before the term or concept of “knowledge management” became fashionable. Although there is no consensus on a notion of knowledge or knowledge-based processing in computing science, the terms are used usually in contradistinction with data or data processing – to highlight the needs to clarify the relationship between symbols stored in computers and what they represent in the world outside, to dissociate the manipulation of such symbolic representations from internal computer processing, to explicitly and formally deal with the semantics of such representations and manipulations, and to make effective use of meta-descriptions in operating on these symbols and structures.

An assortment of techniques for representing and managing codified knowledge has emerged from a number of areas in computer science, notably artificial intelligence, databases, software engineering, and information systems. This movement towards knowledge orientation has not been organized as a coherent movement or even viewed as such, as it has come about for a variety of reasons. From a practical standpoint, the growing complexity of application domains, of software development, and the increasing intertwining of machine and human processes have all contributed to the recognition of such needs and the development of techniques to address them. However, the movement has also been motivated by the search for firmer foundations in the various computing disciplines (Bubenko, 1980; Newell, 1982; Ullman, 1988).

The artificial intelligence area has developed techniques for representing knowledge in forms that can be exploited by computational procedures and heuristics. Database management systems research produced techniques that support the representation and management of large amounts of relatively simple knowledge. Underlying vehicles include relational databases and related technologies. Software engineering have developed elaborate techniques for capturing knowledge that relates to the requirements, design decisions and rationale for a software system. The information systems area has benefited directly or indirectly from these developments.

In computer-based information systems, the meaning of information is usually captured in terms of conceptual information models which offer semantic terms for modeling applications and structuring information (Mylopoulos, 1998). These models build on primitive concepts such as `entity, activity, agent` and `goal`. In addition, the models support mechanisms for organizing information along generic abstraction dimensions, such as generalization, aggregation and classification (Mylopoulos, Jurisica & Yu, 1998). Defining terms and mechanisms for information modeling and organization in conceptual models requires assumptions about the applications to be modeled. For example, if we assume that our applications will consist of interrelated entities, it makes sense to build terms such as `entity` and `relationship` into our conceptual model, and to allow computation based on the semantics of those terms, e.g., to support navigation, search, retrieval, update, and inference based on the semantics of those relationships. The identification of the right concepts for modeling the world for which one would like to do computations (or knowledge management operations) on has come to be known as “ontology” within computer science.

ONTOLOGIES

Ontology is a branch of Philosophy concerned with the study of what exists. Formal ontologies have been proposed since the 18th century, including recent ones such as those by Carnap (1968) and Bunge (1977). From a computational perspective, a major benefit of such formalizations has been the development of algorithms which support the generation of inferences from a given set of facts about the world, or ones that check for consistency. Such computational aids are clearly useful for knowledge management, especially when one is dealing with large amounts of knowledge.

Various methods have been devised to support knowledge organization and interchange. Controlled vocabularies provide a standardized dictionary of terms for use during for example indexing or retrieval. Dictionaries can be organized according to specific relations to form taxonomies. Ontologies further specify the semantics of a domain in terms of conceptual relationships and logical theories.

For example, if one is interested in health care-related knowledge, then `patient`, `disease`, `symptom`, `diagnosis`, and `treatment` might be among the primitive concepts upon which one might want to describe the domain. These concepts and their meanings together define an ontology for health care. Such an ontology can be used as common knowledge that facilitates communication among health workers. It can also be used during development of hospital information systems or decision-support systems.

Earlier work in computational ontologies includes the Cyc project (Lenat & Guha, 1990) and the ARPA Knowledge Sharing effort (Neches et al., 1991). The Knowledge Interchange Format effort provides a declarative language for describing knowledge (Genesereth, 1991). The National Library of Medicine has assembled a large multidisciplinary, multi-site team to work on the Unified Medical Language System, aimed at reducing fundamental barriers to the application of computers to medicine (Humpheys, 1998).

Similarly, an ontology for manufacturing may consist of (industrial) `process`, `resource`, `schedule`, `product` and the like (Vernadat, 1996).

Ontologies may be constructed for different purposes, for example – to enable information sharing and to support specification. When we want to enable sharing and reuse, we define an ontology as a specification used for making ontological commitments (Gruber, 1993). Ontological commitment is an agreement to consistently use a vocabulary with respect to a theory specified by an ontology. In order to support a specification we define ontology as a conceptualization, i.e., ontology defines entities and relationships among them. Every information base is based on either implicit or explicit conceptualization.

Research within artificial intelligence has formalized many interesting ontologies and has developed techniques for analyzing knowledge that has been represented in terms of these. Along a very different path, Wand (1989; 1990) studied the adequacy of information systems to describe applications based on a general ontology, such as that proposed by Bunge (1977).

To characterize and classify current work on ontologies we propose four broad ontological categories, which respectively deal with *static*, *dynamic*, *intentional* and *social* aspects of the world. Our claim is that for a large class of applications, the representation of relevant knowledge can be based on primitive concepts from these four ontological categories. *Static ontology* describes things that exist, their attributes and relationships. *Dynamic ontology* describes the world in terms of states, state transitions and processes. *Intentional ontology* encompasses the world of agents, things agents believe in, want, prove or disprove, and argue about. *Social ontology* covers social settings, permanent organizational structures or shifting networks of alliances and interdependencies.

Static Ontology

Static ontology describes static aspects of the world, i.e., what things exist, their attributes and relationships. Most knowledge representation frameworks assume that the world is populated by `entities` which are endowed with a unique and immutable identity, a lifetime, a set of `attributes`, and `relationships` to other entities. Basic as this ontology may seem, it is by no means universal. For instance, Hayes (1985) offers an ontology for different classes of applications (modeling of material substances where entities (say, a liter of water and a pound of sugar) can be merged resulting in a different entity. Also note that some successful models, such as Statecharts (Harel, 1987), do not support this ontology, because they are intended for real-time systems). This ontology is not trivial. For certain applications it is useful to distinguish between different modes of existence for entities, including physical existence, such as that of the authors of this paper, abstract existence, such as that of the number 7, nonexistence, characteristic of Santa Claus or John's canceled trip to Japan, and impossible existence, such as that of the square root of -1 or the proverbial square circle (Hirst, 1989).

As an example, a partial static ontology for a hospital expressed in the KAOS modeling language (Dardenne, van Lamsweerde & Fickas, 1993) is presented in Figure 1. According to the example, an entity `Hospital` is defined with associated attributes `admitted`, `released`, `registered`, `available` and `specialty`. The first three attributes take as values sets of instances of `Patient`, `available` takes as values sets of instances of `Doctors`, and `specialty` takes as values sets of instances of `Subjects`. The definition includes one set-theoretic invariant constraint, which states that `admitted` is a subset of `registered` for every instance `hosp` of `Hospital`. In addition, `admitted` and `released` are mutually exclusive sets. Next we define a relationship class Treating, which relates the `Patient` and `Hospital` entity classes, has associated cardinality constraints and an invariant. The invariant states that if a patient is treated in the hospital and the patient is in the hospital, then the patient is eventually released.

As another example, an ontology for reproductive medicine would describe not only `patient, diagnosis, treatment`, but also `gametes`, their qualitative and quantitative characteristics, such as morphology (Jurisica et al., 1998). Morphological ontology further includes `shape, spatial-abnormality,` and `texture`. Spatial information is also important for applications which involve physical world, such as geographic information systems (GIS) (e.g., (Croner, Sperling & Broome, 1996)). Spatial information has been modeled in terms of 2D and 3D points or larger units, such as spheres, cubes, or pyramids. Formally defined spatial ontologies allow computational and reasoning operations such as rotation and occlusion to be provided.

Entity Hospital
Has admitted, registered, released: setOf[Patient]
specialty: setOf[Subject]
available: setOf[Doctor]
Invariant ($\forall$ hosp:Hospital)
(hosp.admitted $\subseteq$ hosp.registered
$\wedge$ hosp.admitted $\cap$ hosp.released = $\emptyset$)
...
end Hospital

Relationship Treating
Links Patient [**Role** isTreated, Card 0::1]
Hospital [**Role** treats, Card 0::N]
Invariant ($\forall$ hosp: Hospital, patient: Patient)
(Treating (hosp, patient) $\wedge$ patient $\in$ hosp.admitted $\Rightarrow$
$\Diamond$ patient $\in$ hosp.released
...
end Treating

Figure 1. Defining entities and relationships in KAOS

Dynamic Ontology

Dynamic ontology describes changing aspects of the world. Typical primitive concepts include `state`, `state transition` and `process`. Various flavors of finite state machines and Petri nets have been offered since the 1960's as appropriate modeling tools for dynamic discrete processes. Such models are well-understood and have been used extensively to describe real-time applications in telecommunications and other fields. Statecharts constitute a more recent proposal for specifying large finite state machines (Harel, 1987). A Statechart is also defined in terms of states and transitions, but more than one state may be 'on' at any one time, and states can be defined as AND or OR compositions of other statecharts. As a result, statecharts have been proven much more effective in defining and simulating large finite state machines than conventional methods. The Statecharts model is supported by a popular CASE tool called Statemate.

To take an example from the medical domain again, an *in vitro* fertilization procedure consists of patient selection by diagnosis of infertility, controlled ovarian stimulation for multiple oocyte recruitment and maturation, close monitoring of follicular development by ultrasound and hormonal assessment, oocyte retrieval, insemination of oocytes *in vitro*, determination of fertilization, assessment of embryo development and quality, assessment of endometrial quality, and intrauterine transfer of one or more cleaved embryos (Jurisica et al., 1998). During the treatment, decisions at a particular state depend on results of previous states. To describe such a process, we could use the ConGolog language (Levesque et al., 1997). ConGolog is a high level specification language for defining concurrent processes. Primitive actions can be defined in terms of pre- and post-conditions. Primitive actions can be composed into procedures using modeling constructs such as sequencing (‘`;`’), conditional (`if-then`), iteration (`while <condition> do`), concurrent activity (‘`||`’), non-deterministic choice (`choose`), etc. Although ConGolog offers programming language-like structures for describing processes, its distinctive feature is that the underlying logic is designed to support reasoning with respect to process specifications and simulations, even when the initial state for the process is only partially specified.

```
procedure determineIVFAction (patient)
 consultPatientFile (patient);
  % concurrently obtain patient cahracteristics and embryo morphology
  [request (IVF_patient_DB, doPatientAssessment (patient))]
  ||
  if PatientHasSuccessfulFertilization (patient) then
        [request (IVF_image_DB, doEmbryoMorphologyAnalysis (patient))];
  consultPatientAssessmentReport (patient);
  consultMorhologyAnalysisReport (patient);
  while (embryosAvailable) do
     [if highQuality (embryo) then freezeEmrbyo (patient);
      if lowerQuality (embryo) then transferEmbryo (patient);
      if lowQuality (embryo) then donateEmbryoToResearch (patient)];
  recordFinalReport (patient);
end procedure
```

Figure 2. ConGolog specification of a composite process

Figure 2 shows how one could use ConGolog to define a process for determining IVF action after successful oocyte fertilization. During the process, the physician has to consider the patient’s characteristics (her response to hormonal therapy, treatment history, age, etc.) and morphological properties of embryos. These two actions can be done in parallel. Since the quality of individual embryos vary, one has to consider them iteratively to decide on the action.

Temporal information is often needed when describing dynamic worlds. A temporal ontology can be based on time points and associated relations. An event can be represented as a single time point or two time points. Relations such as `before` or `after` can be used to relate individual points. Allen (1984) proposes a different ontology for time based on intervals, with thirteen associated relations such as `overlap`, `meet`, `before`, and `after`.

Causality is a concept that is closely related to time in ontologies. Causality imposes existence constraints on events: if event A causes event B and A has been observed, B can be expected as well, possibly with some time delay. For example, if a patient has an oocyte of lower quality, it is expected that it will develop into an embryo of a lower quality.

Intentional Ontology

Intentional ontologies encompass the world of motivations, intents, goals, beliefs, alternatives, choices, etc. Typical primitive concepts include `issue`, `goal`, `supports`, `denies`, `subgoalOf`, `agent`, etc. An intentional ontology allows alternate realities to be expressed and reasoned about. The subject of agents having beliefs and goals and being capable of carrying out actions has been studied extensively. For example, Maida (1982) addresses the problem of representing propositional attitudes, such as beliefs, desires and intentions for agents. The importance of the notion of goals and agents, especially for situations involving concurrent actions, has a long tradition in requirements modeling, beginning with Feather (1987) and continuing with recent proposals, such as Dardenne (1993) and Chung (1993).

Software nonfunctional requirements, such as software usability, security, reliability, user-friendlines, performance, etc., can be modeled using softgoals (Chung, 1993; Mylopoulos, Chung & Yu, 1999). Softgoals are goals whose criteria for satisfaction are not crisply defined a priori. The softgoal concept extends intentional ontologies for capturing design rationale (Potts & Bruns, 1988). Making available intentional information such as pro and con arguments and resulting decisions can be very useful during design and maintenance of information systems. It has been shown that softgoals can play an important role in many design tasks, by guiding the designer through alternative design choices. Jurisica and Nixon (1998) shows how one would use softgoals to build quality into complex medical decision support systems.

Consider an example of building an information system for an IVF clinic, which requires both clinical and research use of the system. System performance is an important factor for complex applications. Good performance includes fast response time and low space requirements. For the IVF system, a developer might state that one important goal is to have fast response time when accessing patient records, for reasoning as well as case updating. This requirement is represented as a softgoal: `Time[Patient Records and Reasoning]`, as shown at the top of Figure 3. `Time` is the *type* of the softgoal and `[Patient Records and Reasoning]` is the *topic*. This goal may be synergistic with or competing with the other main goal `Time[Research Reasoning]`, which is to have fast response time for reasoning operations done by researchers.

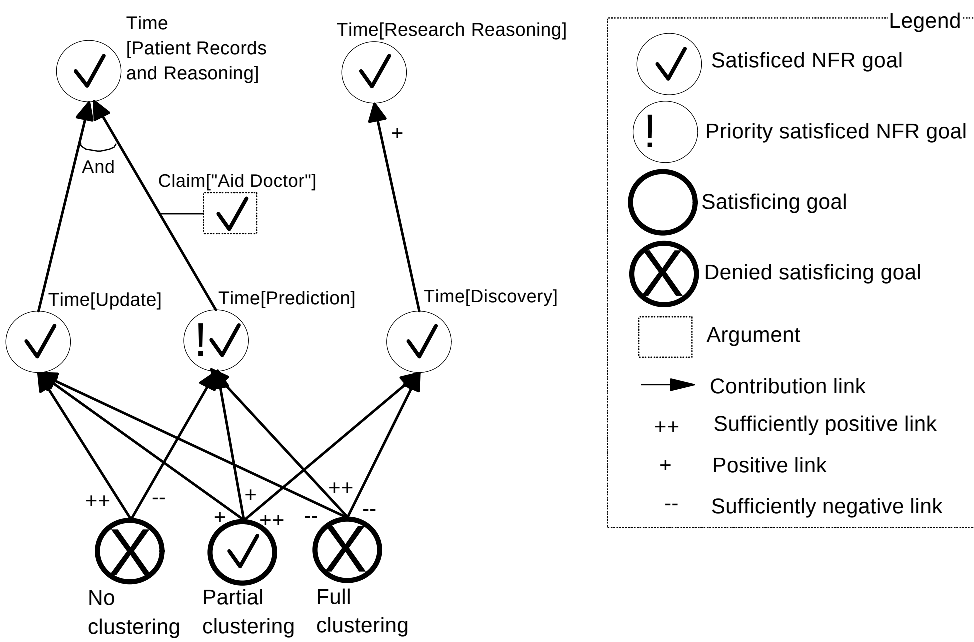

Figure 3. Dealing with performance requirements for reasoning

Using *methods* and *catalogues* of knowledge (for performance, case-based reasoning, IVF, etc.), goals can be refined into more specialized goals. Here, the developer used knowledge of the IVF domain to refine the time goal for patient information into two goals, one for good response time for updating patient records and the other for good response time for the retrieval and decision making process. These two offspring goals are connected by an And link to the parent goal. This means that if both the goal for fast updates and the goal for fast prediction are accomplished then we can say that the parent goal of fast access to patient records will be accomplished.

The figure also shows an example of recording design rationale - the reasons for decisions - using the NFR Framework's *arguments*. As part of the development graph, arguments are available when making further decisions and changes. It is important to note that the developers use their expertise to determine what to refine, how to refine it, to what extent to refine it, as well as when to refine it. The NFR Framework and its associated tool can help the developer do consistency checking and keep track of decisions, but it is the developer who is in control of development process (Chung et al., 1999).

Social Ontology

A social ontology covers social settings, organizational structures, or shifting networks of alliances and interdependencies (Galbraith, 1973; Mintzberg, 1979; Scott, 1987). Traditionally, social ontologies have been characterized in terms of concepts such as `actor`, `position`, `role`, `authority`, `commitment`, etc. Speech acts theory offers an ontology for modeling communication among actors (Medina & Mora, 1992). Social ontologies are also of interest in distributed artificial intelligence. Some of the concepts have been formalized using specialized logic (Castelfranchi, 1993).

Yu proposes a set of concepts which focus on strategic dependencies between actors (Yu, 1993; Yu, 1995). Such a dependency exists when an actor is committed to satisfying a goal or softgoal, carry out a task, or deliver a resource. Using these concepts, one can create organizational models that provide answers to questions such as “why does the technician need to enter detailed morphological information?”. Creating these models enables the analysis of an organizational setting, which is an important step in the re-design of business processes and the subsequent development of information systems (Yu, Mylopoulos & Lesperance, 1996). Reasoning about the inter-dependency relationships among strategic actors is also important for enterprise modelling and analysis (Yu, 1999).

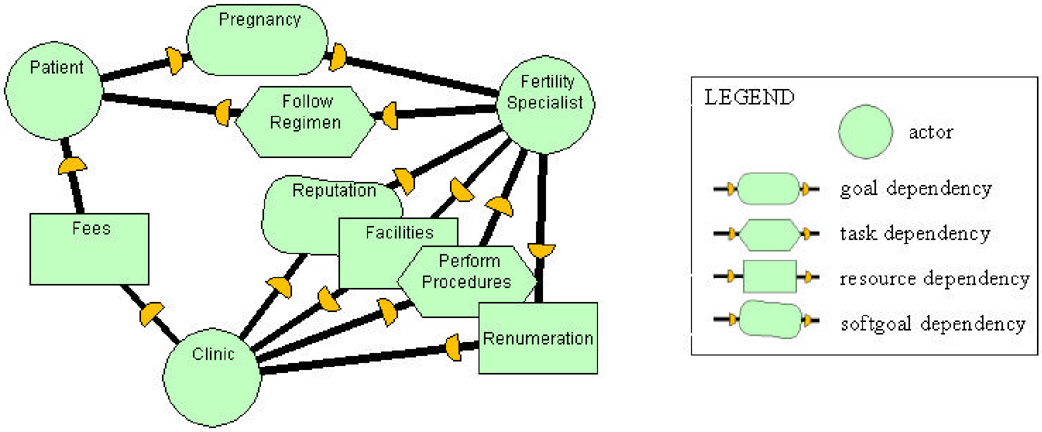

Figure 4. Strategic dependencies between actors

Health care involves some of the most complex social and organizational structures and processes in our society. In developing systems to support health care, it is important to understand the social context in order to identify and select appropriate technical solutions. Although the social issues can be very complex, adopting a suitable social ontology can provide some assistance to organizing and discerning the many issues, and to support analysis and argumentation.

Figure 4 shows a simplified example of a strategic dependency graph involving an IVF patient, the clinic, and the fertility specialist. The patient depends on the specialist to achieve the goal of pregnancy. The clinic depends on the specialist to perform procedures and also for good reputation. The specialist depends on renumeration from the clinic, which in turn depends on fees from the patient. One can use this kind of social ontology to model and explore alternative approaches to health care delivery.

APPLICATIONS OF ONTOLOGIES

The above categories of ontologies need to be used together in actual applications. For example, a major goal of reusable domain ontologies is to support the interchange of information. Sharable ontologies would allow different information systems to inter-work and cooperate with each other to accomplish goals. An agent in a medical diagnosis system uses an ontology of clinical concepts, both during structured data entry and decision support. A diagnostic agent needs to cooperate with a bibliographic agent that uses an ontology for bibliographies to associate literature references with particular diseases.

Developing ontologies that cover domain and application characteristics can be used to not only support system integration by using standardized vocabularies but also system development by reusing these ontologies. One can use various tools to help in the process, such as Ontolingua (Gruber, 1992). Ontolingua is an ontology development environment that provides tools for authoring ontologies. The tools support creating ontologies by assembling and enhancing ontologies obtained from a library of modular, reusable ontologies. Once we define ontologies for one or several domains, we may organize them to create a library of reusable ontologies (Heijst et al., 1995). Such libraries can be useful for building information systems and during knowledge acquisition (Tu et al., 1995). When the models get large, we need tools to help with their management. Analysis tools can help with model verification and validation. Verification checks if the model satisfies existing syntactic rules (e.g., checking cardinality constraints for entity-relationship-like models or checking semantic consistency of rules and constraints such as the patient cannot have more embryos than she had oocytes). Validation checks the consistency of an information base with respect to its application**.**

Most of the current efforts in medical ontologies are directed towards generation of controlled terminologies, or reference ontologies (Gennari, 1995; Musen, 1998; Oliver, 1998). Such vocabularies taxonomically organize terms in certain areas. This supports consistent usage of terms, enables information sharing and system cooperation. Kahn (1998) suggests to use an Internet-based ontology system, called NEON (Networked-based Editor for ONtologies), to standardize radiology appropriateness criteria. Individual concepts are represented in a semantic network and the system supports import and export of ontologies using SGML. Individual entities include concept name, abbreviation, synonym, and links such as `affectedBy, hasPart, partOf` and `imagedBy`. This approach can help not only to standardize terminology but also organize existing vocabularies.

Over the years, efforts to control medical terminology have resulted in various standard medical vocabularies, such as International Classification of Diseases (ICD-9-CM), Systematized Nomenclature of Human and Veterinary Medicine (SNOMED), Medical Subject Headings (MeSH), Read Codes, Current Procedural Terminology (CPT), Unified Medical Language System (UMLS), etc. However, none is sufficiently comprehensive and accepted to meet the full needs of the electronic health record (Shortliffe, 1998).

Despite standardization efforts, combining and synchronizing individual versions of existing medical terminology vocabularies is a problem (Oliver, 1998). For this reason, National Library of Medicine created a UMLS, which is a composite of about 40 vocabularies that contain approximately 500 thousand biomedical concepts and over 1 million terms to describe them (Humphreys, 1998). The Medical Ontology Group of Italian National Research Council has been working on integrating and reusing existing terminological ontologies in medicine (Steve, Gangemi & Pisanelli, 1997). Steve et al. have designed an ontology library ON9, which is written in Ontolingua (Gruber, 1992) and Loom (MacGregor, 1993). It includes thousands of medical concepts and organizes them into domain, generic and meta-level theories. They use a methodology called ONIONS to aid construction of ontologies starting from existing, contextually heterogeneous terminologies. This work led to a successful integration of five medical terminology systems: the UMLS-SN (about 170 semantic types and relations, and their 890 defined combinations), SNOMED-III (about 600 most general concepts), Gabrieli Medical Nomenclature (about 700 most general concepts), ICD10 (about 250 most general concepts), and the Galen Core Model- 5g (about 2000 items).

Another problem that must be addressed is complexity of controlled medical vocabularies. It is important to provide tools and techniques to help designing and organizing such vocabularies. Earlier models, such as ICD-9-CM, DSM, SNOMED, and Read Version 2 use the code not only to identify a concept uniquely, but also to indicate where a concept lies in the hierarchy. As a result, particular concept can be entered to only one place in the hierarchy. In addition, the number of levels in the hierarchy is usually limited, since existing codes have a fixed number of digits and each digit indicates a level. Alternatively, some system do not use code to indicate hierarchical location, e.g., Read Version 3, the MED (Medical Entities Dictionary) and SNOMED-RT. Gu et al. (1999) proposes a methodology to partition vocabularies into an `isA` hierarchy. Authors show how to partition an existing MED dictionary, which comprises 48,000 concepts, over 61,000 `isA` links and over 71,000 additional links (`categoryOf, roleOf`). Based on the partitioning into sets of concepts with the same sets of properties, MED was mapped into an object-oriented database ONTOS.

DISCUSSION

In the current literature on knowledge management, it is often observed that the main challenges are in the realm of human organizational culture and practices (Ruggles, 1998). However, the impact and potential of advanced information technologies, both positive and negative, should not be underestimated. Given today's vast, complex and dynamic information environments, the potential for using information technology to help arrive at and manage knowledge is enormous. However, the pitfalls are also plentiful. This is why the complementary use of concepts and techniques from information science and from information systems is crucial.

The ontology approach from information modeling described in this paper derives its power from the formalization of some domain of knowledge. However, many domains resist precise formalization. In each domain, there are points at which formalization becomes more of a straitjacket than a liberating force. The challenge is therefore not so much to decide which approach is better, but to develop techniques for the various approaches to work closely together in a seamless way.

This may be illustrated in a scenario of designing a form in which there are fields to be filled with content. If the content can be an arbitrary text string, then there is not much computational leverage that can be derived from it. But it is highly flexible and can accommodate any kind of input. If on the other hand, one restricts the content to a set of pre-defined values which obey given rules, and whose meanings are well-defined, one gains computational power, but loses flexibility. In an e-mail message, the address and date fields are strictly defined and can be operated on by automated procedures, such as those for routing and sorting. One can hardly imagine an e-mail system that requires human intervention to interpret addresses to manually sort and route the mail through the Internet. To gain the benefit of speedy communication, we have learned to live with the inflexibility of using precise addresses. The message body, however, is arbitrary text, and therefore requires human interpretation. When one is faced with thousands of messages week after week, some kinds of technology support becomes desirable.

There can be many shades in between full automation and no automation, as well as many forms of interactive semi-automated support. One can do string-based retrieval, filter out unwanted messages, or file them automatically into pre-defined folders. To do more powerful processing, one would need to attribute more meaning to the content. For example, one could define patterns which would be recognized as dates within a message body. One could define concepts related to meetings so as to recognize meeting announcements. One could then have reminders automatically inserted into an appointments calendar. In order to achieve this, one needs to define an ontology of appointment dates (the concept of dates and available time slots in the context of appointments), and perhaps also an ontology of meeting scheduling – what constitutes having two meetings being scheduled too close together; constraints such as meetings with overlapping time intervals cannot be in the same room, etc.

This example illustrates that ontologies are often not about an objective world, but are based on social conventions and agreements. Concepts, meanings, and interpretations are relative to some community and can change over time. Community boundaries and identities can also be dynamic. Here again, the experience and expertise in the information science area for dealing with much more open-ended kinds of human knowledge can be invaluable. Technical frameworks are increasingly paying attention to these factors, as exemplified in the intentional and social ontologies outlined above. However, technological support for dealing with these issues, such as contextual mechanisms for knowledge scoping and sharing, multiple perspectives and meanings, negotiation support, knowledge evolution, etc., can only be partial – again due to inherent limits to the formalization of human knowledge.

CONCLUSIONS

The technologies of information systems have been progressing at a rapid pace. Information systems are now being called upon to support knowledge management, and not just to process data or information. Many advances contribute to taking information systems beyond mere data into the realm of knowledge. These include: cooperative query processing (Chu et al., 1996; Jurisica, 1999), similarity-based retrieval and browsing (Jurisica, Glasgow & Mylopoulos, 1999), data mining and knowledge discovery (Jurisica et al., 1998), text understanding (Hahn, Romacker & Schulz, 1999; Riloff, 1996), data translation services (Gruber, 1993), and knowledge sharing (Orthner, Scherrer & Dahlen, 1994), to name a few.

However, the key to providing useful support for knowledge management lies in how meaning is embedded in information models as defined in ontologies. In this paper, we have surveyed some of the basic concepts under each of four ontological categories. We outlined the benefits and limitations of the ontology-based approach, and argued for the need for a combination of techniques from information science and information systems.

Acknowledgements. This research is supported in part by the Natural Sciences and Engineering Research Council of Canada, Communication and Information Technologies Ontario, and the Institute for Robotics and Intelligent Systems.

REFERENCES

Allen. (1984). Towards a general theory of action and time. *Artificial Intelligence*, **23**, 123-154.

Bubenko, J. A. (1980). Information Modeling in the Context of System Development, in *Proceedings IFIP Congress*, pp. 395-411.

Buckland, M. (1999). The Landscape of Information Science: The American Society for Information Science at 62. *J. Amer. Soc. Info. Sci.,* Special Issue.

Bunge. (1977) *Treatise on Basic Philosophy: Ontology I* **-** *The Furniture of the World.* Reidel.

Carnap, R. (1967) *The Logical Structure of the World: Pseudoproblems in Philosophy*. University of California Press.

Castelfranchi, C. & Mueller, J.-P., (Eds). (1993). From *Reaction to Cognition, 5th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW '93*, Neuchatel, Switzerland, August 25-27, 1993. Published as *Lecture Notes in Computer Science*, 957. Springer.

Chu, W.W., H. Yang, K. Chiang, M. Minock, G. Chow & C. Larson. (1996). CoBase: A scalable and extensible cooperative information system*. Journal of Intelligent Information Systems*, **6**.

Chung, L. (1993). *Representing and Using Non-Functional Requirements: A Process-Oriented Approach.* Ph.D. thesis, Department of Computer Science, University of Toronto.

Chung, L., Nixon, B.A., Yu, E. & Mylopoulos, J. (1999). *Non-Functional Requirements in Software Engineering.* (forthcoming monograph). http://www.cs.toronto.edu/km/nfr

Clamen, S. M. (1994). Schema evolution and integration. *Distributed and Parallel Databases.* **2**(1):101-126.

Coetzee, S. & Bishop, J. (1998). A new way to query GISs on the Web*. IEEE Software*, 1998. **15**(3): 31.

Croner, C.M., Sperling, J., & Broome, F. R. (1996). Geographic information systems (GIS): New perspectives in understanding human health and environmental relationships*. Stat Med*, **15**(17-18): 1961-77.

Curtis, B., Kellner, M., & Over, J. (1992). Process modeling. *Communications of the ACM*, **35**(9).

Dardenne, A., van Lamsweerde, A., & Fickas, S. (1993). Goal directed requirements acquisition. *Science of Computer Programming*, **20**, 3-50.

Falasconi, S. & Lanzola, G. (1997). Ontology and terminology servers in agent-based health-care information systems. *Methods Inf Med*, **36**(1): 30-43,.

Feather, M., (1987). Language support for the specification and derivation of concurrent systems. *ACM Transactions on Programming Languages,* **9**(2), 198-234.

Fensel, D. M. & Rousset, C. (1998). Workshop on comparing description and frame logic. *Data & Knowledge Engineering*, **25**(3): 347-352.

Galbraith, J. (1973). *Designing Complex Organizations*. Addison Wesley.

Genesereth, M. R. (1991). Knowledge Interchange Format. Principles of Knowledge Representation and Reasoning, *Proceedings of the Second International Conference*, Cambridge, MA, pp. 599-600, Morgan Kaufmann.

Gennari, J. H. & Oliver, D. E. (1995). A web-based architecture for a medical vocabulary server. *19th*

Annual Symposium on Computer Applications in Medical Care, New Orleans, LA..

Gruber, T. R. (1992). Ontolingua: A mechanism to support portable ontologies. *Technical Report KSL-91-66,* Stanford University, Knowledge Systems Laboratory.

Gruber, T. R. (1993). A translation approach to portable ontology specification. *Knowledge Acquisition*, **5**(2):199-220.

Gu, H., Perl, Y., Geller, J., Halper, M. & Singh, M. (1999). A methodology for partitioning a vocabulary hierarchy into trees. *Artificial Intelligence in Medicine*, **15**, pp. 77-98.

Hahn, U., Romacker, M. & Schulz, S. (1999). How knowledge drives understanding - matching medical ontologies with the needs of medical language processing, *Artificial Intelligence in Medicine,* **15**, pp. 25-51.

Harel, D. (1987). State charts: A visual formalism for complex systems. *Science of Computer Programming,* 8.

Hayes, P. (1985). The second naive physics manifesto. In Hobbs, J. & Moore, R. (Eds.). *Formal Theories of the Commonsense World,* Ablex Publishing Corp., Norwood, N. J., 1-36.

Heijst, G. Van, Falasconi, S., Abu-Hanna, A., Schreiber, G. & Stefanelli, M. (1995). A cese study in ontology library construction. *Artificial Intelligence in Medicine*, **7**, pp. 227-255.

Hirst, G. (1989). Ontological assumptions in knowledge representation. *Proceedings First International Conference on Principles of Knowledge Representation and Reasoning,* Toronto, 157-169.

Humphreys, B. L. & Lindberg, D. A. (1998). The Unified Medical Language System: an informatics research collaboration. *J Am Med Inform Assoc,* **5**(1): 1-11.

Jurisica, I. (1999). Supporting Evidence-Based Medicine by Cooperative Information Systems. in *Digital Knowledge III*. Toronto, ON: CPI.

Jurisica, I., DeTitta, G., Luft, J., Glasgow, J. & Fortier, S. (1999). Knowledge Management in Scientific Domains. in *AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning*. Orlando, FL: AAAI Press.

Jurisica, I., Glasgow, J. & Mylopoulos, J. (1999). Building Efficient Conversational CBR Systems. Incremental Iterative Retrieval and Browsing. Submitted.

Jurisica, I. & Nixon, B. (1998). Building Quality into Case-Based Reasoning Systems. in *Lecture Notes in Computer Science, Conference on Advance Information Systems Engineering, CAiSE'98*. Pisa, Italy: Springer-Verlag, pp. 363-380.

Jurisica, I., Mylopoulos, J., Glasgow, J., Shapiro, H & Casper, R. (1998). Cased-Based Reasoning in IVF: Prediction and Knowledge Mining, *Artificial Intelligence in Medicine,* **12**(1):1-24.

Kahn, C. (1998). An Internet-based ontology editor for medical appropriateness criteria. *Computer Methods and Programs in Biomedicine,* **56**, pp. 31-36.

Lenat, D. B., & Guha, R. V. (1990). *Building Large Knowledge Bases*. Addison-Wesley, Reading, MA.

Levesque, H., Reiter, R., Lesperance, Y. & Lin, F. (1997). GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, pp. 59-84.

MacGregor, R. M. (1993). Representing reified relations in Loom. *Journal of Experimental and Theoretical Artificial Intelligence.* **5**:179-183.

Maida, A. & Shapiro, S. (1982). Intensional Concepts in Propositional Semantic Networks. *Cognitive Science*, **6**, 291-330.

Medina-Mora, R., Winograd, T., Flores, R. & Flores, F. (1992). The Action Workflow Approach to Workflow Management Technology. *Proc. Conf. on Comp. Supported Cooperative Work*, pp. 281-288.

Mintzberg, H. (1979). *The Structuring of Organizations*. Prentice Hall.

Musen, M. A. (1998). Modern architectures for intelligent systems: Reusable ontologies and problem-solving methods. *AMIA Fall Symposium*, Orlando, FL.

Mylopoulos, J. (1998) Information modeling in the time of the revolution. *Information Systems*, **23**(3/4): 127-155.

Mylopoulos, J., Chung, L. & Yu, E. (1999). From Object-Oriented to Goal-Oriented Requirements Analysis. *Communications of the ACM*, **42**(1): 31-37.

Mylopoulos, J., Jurisica, I. & Yu, E. (1998). Computational mechanisms for knowledge organization. *Proceedings of the 5th International Conference of the International Society on Knowledge Organization (ISKO'98),* Lille, France, Ergon Verlag.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI Magazine*, **12**(3):16-36.

Newell, A. (1982). The Knowledge Level, *Artificial Intelligence*, **18**(1): 87-127.

Oliver, D. E. (1998). Synchronization of diverging versions of a controlled medical terminology. Stanford, CA, Stanford University School of Medicine, Technical Report SMI 98-0741.

Oliver, D. E., Shahar, Y., Shortliffe, E. H. & Musen, M. A. (1999). Representation of change in controlled medical terminologies, *Artificial Intelligence in Medicine*, **15**, pp. 53-76.

Orthner, H.F., Scherrer, J. R. & Dahlen, R. (1994). Sharing and communicating health care information: summary and recommendations. *Int J Biomed Comput*. **34**(1-4): 303-318.

Potts, C. & Bruns, G. (1988). Recording the reasons for design decisions. *Proceedings of the 10th International Conference on Software Engineering*, Singapore.

Rector, A. L. & Bechhofer, S. B. (1997). The GRAIL concept modeling language for medical terminology. *Artificial Intelligence in Medicine*, **9**, pp. 139-171.

Rillof, E. (1996). An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence in Medicine,* **85**, pp. 101-134.

Ruggles, R. (1998). The State of the Notion: Knowledge Management in Practice. *California Management Review*, **40**, 3.

Scott, W. (1987). *Organizations: Rational, Natural or Open Systems*. Prentice Hall, second edition.

Shortliffe, E. H. (1998). The evolution of health-care records in the era of the Internet. *MEDINFO'98*.

Steve, G., Gangemi, A. & Pisanelli, D. M. (1997) Integrating medical terminologies with ONIONS methodology. http://www.saussure.irmkant.rm.cnr.it/onto/publ/onions97/onions97.pdf. To be published by IOS-Press.

Studer, R. & Benjamins, V. R. (1998). Knowledge Engineering: Principles and methods. *Data & Knowledge Engineering*, **25**(1-2): 161-197.

Tu, S. W., Eriksson, H., Gennari, J. H., Shahar, Y. & Musen, M. A. (1995). Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: Application of PROTEGE-II to protocol-based decision support. *Artificial Intelligence in Medicine,* **7**, pp. 257-289.

Ullman, J. D. (1988). *Principles of database and knowledge-base systems*, vol. 1, Computer Science Press.

Vickery, B. C. (1997), Ontologies, *Journal of Information Science*, **23**(4): 277-286.

Vernadat, F. (1996). *Enterprise Modeling and Integration*, Chapman & Hall.

Wand, Y. (1989). A proposal for a formal model of objects. In Kim, W. & Lochovsky, F. (Eds.). *Object-Oriented Concepts, Databases and Applications*. Addison-Wesley.

Wand, Y. & Weber, R. (1990). An ontological model of an information system. *IEEE Transactions on Software Engineering,* **16**, 1282-1292.

Yu, E. (1993). Modeling organizations for information systems requirements engineering. *Proceedings IEEE International Symposium on Requirements Engineering*, San Diego, IEEE Computer Society Press, pp. 34-41.

Yu, E. (1995). *Modelling Strategic Relationships for Process Reengineering.* Ph.D. thesis, Department of Computer Science, University of Toronto.

Yu, E. (1999). Strategic Modelling for Enterprise Integration. *Proceedings 14th World Congress of the International Federation of Automatic Control*, July 5-9, 1999, Beijing, China.

Yu, E., Mylopoulos, J. & Lesperance, Y. (1996). AI Models for Business Process Reengineering. *IEEE Expert: Intelligent Systems and Their Applications*, **11**(4), pp. 16-23.