

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 concepts, methods, and tools supporting the management of human knowledge. The main objective of this paper is to survey 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 science theory and practice.

The survey classifies the concepts used for knowledge representation into four broad ontological categories. *Static ontologies* describe 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 ontologies* encompass the world of things agents believe in, want, prove or disprove, and argue about. Finally, *social ontologies* cover social settings – agents, positions, roles, authority, permanent organizational structures or shifting networks of alliances and interdependencies.

Keywords: Ontologies; Knowledge management; Knowledge representation; Information science; Information systems

1. Introduction

Knowledge management is concerned with the representation, organization, acquisition, creation, use 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 represented in computer-readable forms, we also need to build tools

* This is an extended and revised version of: Jurisica, I., Mylopoulos, J., Yu, E., “Using ontologies for knowledge management: An information systems perspective,” Proceedings of the Annual Conference of the American Society for Information Sciences (ASIS’99), Washington, DC, November 1999.

Received 17 September 2002

Revised 2 December 2002

Accepted 8 August 2003

Published online 8 March 2004

that can effectively search databases, files, web sites and the like to extract information, capture its meaning, organize and analyze it, and make it useful. This paper focuses on the concepts used in computer-based information systems to capture and exploit the meaning of information.

Information science, as it exists today, already provides many of the foundations for knowledge management. After all, the documentation tradition has a long history of developing methods and practices for organizing the vast expanses of human knowledge so that it is accessible by diverse 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 for full-text documents, and most recently for distributed, heterogeneous information sources. Nevertheless, many significant challenges remain.

Historically, information science has focused on the document as the primary unit of information. Documents have traditionally been paper based, consisting primarily of published books and articles. The contents of such documents have been individually meaningful, at least at a literal, surface level. Deeper meanings, however, do require interpretation in relation to interconnected documents as well as social and cultural contexts. These connections are usually relatively sparse (e.g., a few dozen references in an academic article) and have little built-in semantics. For instance, a reference simply leads to another document, much like untyped hypertext links that dominate today's World Wide Web (WWW). Moreover, documents have traditionally been fairly stable, and new ones took considerable time and effort to create. Finally, documents were primarily used by humans, who read, interpreted and acted on them.

The 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 – paragraphs, text fragments, pieces of data, video or audio clips, etc. They are documents not so much in the common sense usage of the term, but rather logically identifiable and locatable packages of information. This change in the granularity of information units has increased the number of units that need to be managed by many orders of magnitude. In addition, today's documents tend to be much more densely connected, referring or relating to each other in multiple ways. Moreover, documents are continuously created and revised, post-it notes are attached, detached, reattached in a different context, or discarded. 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 digital media. This means that research on knowledge management must 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 information unit spectrum. Information in information systems 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 fast and reliable information processing 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 information systems for managing knowledge, but there are considerable challenges too. Perhaps the most important among these is the development of techniques for exploiting meaning to produce computationally useful results.

Section 2 of the paper discusses basic approaches to capture the meaning of information, while Sect. 3 introduces ontologies and discusses four categories of ontological elements. In Sect. 4, we discuss applications of ontologies in medicine. Section 5 includes discussion and summarization of the main points of the paper.

2. Exploiting Meaning in Information Systems

Interestingly, within the field of computer science, there has been a gradual movement toward what one might call knowledge orientation or semantic processing. This has been taking place over the past 20–30 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 computer science, the terms are used usually in contradistinction with data or data processing to highlight the need to clarify the relationship between symbols stored in computers and what they represent in the world outside. The terms also imply technologies that deal explicitly and formally with the semantics of such representations and their manipulations.

An assortment of techniques for representing and managing codified knowledge has emerged from several areas in computer science, notably artificial intelligence, databases, software engineering, and information systems. This movement toward knowledge orientation has not been organized into a coherent research methodology, as it has come about for a variety of reasons. From a practical standpoint, the growing complexity of application domains and the increasing intertwining of machine and human processes have all contributed to the recognition of knowledge orientation as a promising research direction that can address open knowledge-management problems. However, the movement has also been motivated by the search for solid foundations in various computing disciplines (Bubenko 1980; Newell 1982; Ullman 1988).

Artificial Intelligence has developed techniques for representing knowledge in forms that can be exploited by computational procedures and heuristics. Research on databases has produced techniques that support the representation and management of large amounts of relatively simple information. Supporting mechanisms include relational databases and associated technologies. Software engineering has developed elaborate techniques for capturing knowledge that relates to the requirements, design decisions and rationale for a software system. The area of information systems 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 that 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 et al. 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, i.e., to support navigation, search, retrieval, update and inference that are consistent with the meaning of those terms. The identification of the right concepts for modeling a world for which one would like to do computations (or knowledge management operations) has come to be known as an *ontology* within computer science.

3. 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 (1967) and Bunge (1977). In computer science, ontologies have come in a variety of forms, ranging from lexicons, to dictionaries and thesauri or even first order logical theories. Lexicons provide a standardized dictionary of terms for use during, for example, indexing or retrieval. Dictionaries can be organized according to specific relations to form hierarchies (taxonomies, meronomies, etc.). Thesauri add related terms to any given term. In any of these forms, ontologies are useful because they encourage standardization of the terms used to represent knowledge about a domain. When ontologies are formalized in first-order logic or a subset thereof, they can also support inference mechanisms. For a given collection of facts, these mechanisms can be used to derive new facts or check for consistency. Such computational aids are clearly useful for knowledge management, especially when one is dealing with large amounts of knowledge.

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 application 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. Similarly, an ontology for manufacturing may include the concepts of (industrial) process, resource, schedule, product and the like (Vernadat 1996). Brazma et al. (2001) proposes a flexible and open specification language called MIAME (Minimum Information About Microarray Experiments) to represent microarray experiments. Although details for particular experiments may be different, MIAME aims to define the core that is common to most experiments. MIAME is not a formal specification language, but a set of guidelines, which encourages users to provide their own qualifiers and values identifying the source of their terminology. It promotes the use of controlled vocabularies and external ontologies.

Early work in computational ontologies includes the Cyc project (Lenat and 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). Research within artificial intelligence has formalized interesting specialized ontologies and has developed techniques for using them to represent and analyze knowledge. Well-designed ontologies need to be addressed as properly deep and long-standing problems in philosophy, such as the problems of identity and unity. The designer of ontologies needs to provide answers to questions such as:

- Do entities have any *essential properties*?
- Does a change of parts affect identity?
- When does an entity count as *one*?

Guarino and Welty (2000) offer an elegant account of how such issues can be taken into consideration in the design of taxonomies of concepts. Along a very differ-

ent path, Wand (1989, 1990) studied the adequacy of conceptual models 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 derived from these four ontological categories. For example, if we want to model a university environment, we may choose entities and relations to model static aspects of the domain and processes to model dynamic aspects. For a multiagent system, on the other hand, we may want to use concepts such as agent, team, goal and social dependency to model social and intentional aspects of the application. Of course, agents and goals could be represented simply as entities. Such representations, however, miss important properties of agency (e.g., an agent's intentionality and autonomy) and lead to incomplete forms of inference. For example, representations that use elements of dynamic ontologies are amenable to simulation, a special form of inference not supported by static ontologies. Likewise, formal goal models support their own special forms of reasoning – e.g., Giorgini et al. (2002) – that go beyond inference mechanisms for entities and relationships. Our classification of ontological concepts into four categories has been derived from a broad survey of modeling techniques in computer science (Mylopoulos 1998).

The rest of the section briefly reviews the four categories. A *static ontology* describes things that exist, their attributes and relationships. A *dynamic ontology* describes the world in terms of states, state transitions and processes. An *intentional ontology* encompasses the world of agents, things agents believe in, want, prove or disprove, and argue about. Finally, a *social ontology* covers social settings, permanent organizational structures or shifting networks of alliances and interdependencies.

3.1. Static Ontologies

A 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* that 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 elements from a static ontology because they are intended for real-time systems. Static ontologies are 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, characteristics 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 et al. 1993) is presented in Fig. 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 *doctor*, and *specialty* takes as values sets of instances of *subject*. The definition includes one set-theoretic

```

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

```

Fig. 1. An example of a partial static ontology for a hospital in KAOS modeling language (Dardenne et al. 1993).

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 at one instance of time. Next we define the relationship class *treating*, which relates a patient to a hospital entity, 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.

Many domains require support for description and analysis of image data. For example, an ontology for an *in vitro* fertilization (IVF) clinic needs concepts for patient, diagnosis, treatment and morphological characteristics of sperms, oocytes and embryos (Jurisica et al. 1998).

Spatial information is also important for applications that involve the physical world, such as geographic information systems, e.g., Croner et al. (1996). Spatial information has been modeled in terms of two-dimensional (2D) and three-dimension (3D) points or larger units, such as spheres, cubes or pyramids. Formal spatial ontologies enable computational and reasoning operations such as rotation and occlusion to be provided. One such class of applications deals with representing 2D and 3D formations of proteins. Proteins are large, complex molecules composed of long chains of molecules, called amino acids. They provide the structural components of cells and enzymes for essential biochemical reactions. The subcellular localization of proteins specifies where they are and determines their ability to interact with other proteins and small metabolites in their local environment. One can then define an ontology for biological function using molecular interactions (Karp 2000), which then enables computation with biological function. Subcellular localization of proteins is another key functional characteristic. Gene ontology addresses this issue at the cellular level by providing a hierarchy of cellular component concepts such as *organelles*, *membranes* and *protein complexes* (Ashburner et al. 2000). These concepts are used to annotate the location of gene products (Xie et al. 2002).

3.2. Dynamic Ontologies

Dynamic ontologies describe changing aspects of the world. Typical primitive concepts include *state*, *state transition* and *process*. Various flavors of finite-state ma-

```

procedure determineIVFAction (patient)
  consultPatientFile (patient);
  % concurrently obtain patient characteristics and embryo morphology
  [request (IVF_patient_DB, doPatientAssessment (patient))]
  ||
  if PatientHasSuccessfulFertilization (patient) then
    [request (IVF_image_DB, doEmbryoMorphologyAnalysis (patient))];
  consultPatientAssessmentReport (patient);
  consultMorphologyAnalysisReport (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

```

Fig. 2. ConGolog (Levesque et al. 1997) specification of the process of IVF action after successful oocyte fertilization. The physician has to, in Parallel, consider patient's characteristics and morphological properties of embryos.

chines and Petri nets have been offered since the 1960s 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 compared with conventional modeling methods. The statecharts model is supported by a popular CASE tool called Statemate.

To take another example from the reproductive-medicine domain, an IVF 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 *postconditions*. Primitive actions can be composed into procedures using modeling constructs such as sequencing (';'), conditional (if-then), iteration (while <condition> do), concurrent activity ('||'), nondeterministic 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.

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 are

to be carried out in parallel. Because the quality of individual embryos varies, 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 multiple 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 13 associated relations such as overlap, meet, before and after.

Causality is a concept that is closely related to time. 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. In this context, one may need to explore properties such as probability and transitivity.

3.3. Intentional Ontologies

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 enables 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.

Modeling the issues that arise during complex decision making is discussed in Conklin and Begeman (1988). The application of such a framework to software design, intended to capture the arguments pro and con, and the decisions they result in, has been a fruitful research direction since it was first proposed in Potts and Bruns (1988), with notable refinements described in MacLean et al. (1991), Lee and Lai (1991). For example, MacLean et al. (1991) models design rationale in terms of *questions* (Q), *options* (O) and *criteria* (C). Figure 3 shows the structure of a decision space concerning the design of an automated teller machine (ATM). The four questions raised have associated options. Choice among them will be done by using an associated list of criteria. For example, for the question of what range of services will be offered (by the ATM under design), there are two options, full range and cash only, and two criteria for choosing among them. The cash-only option raises an auxiliary question, whether services can be restricted by having switchable machines, where services can be masked out, or by having machines that are inherently limited in the services they offer. On a complementary front, Gotel and Finkelstein (1995) study the types of contributions a stakeholder can make to an argumentation structure such as the one shown in Fig. 3.

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 influential proposals, such as Dardenne et al. (1993) and Chung (1993).

Software nonfunctional requirements (NFRs), such as software usability, security, reliability, user friendliness, performance, etc., can be modeled using softgoals (Chung 1993; Mylopoulos et al. 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 and Bruns 1988). Making available

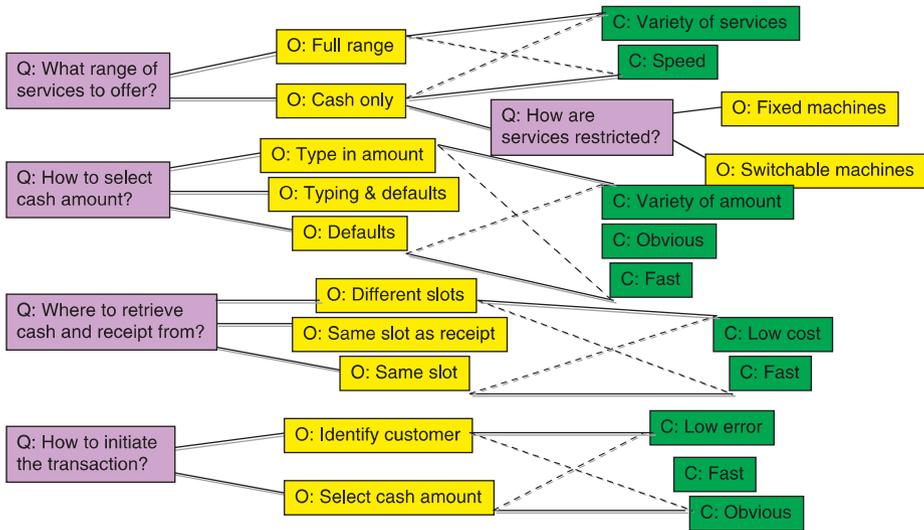


Fig. 3. Modeling design rationale in terms of questions (Q), options (O) and criteria (C).

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) show how one would use softgoals to build quality into complex medical decision-support systems.

Consider an example of an information system for an IVF clinic, with the aim to fulfill both clinical and research purposes. Suppose potential users rate performance as an important factor for the system. Further assume that good performance is characterized as fast response time when accessing patient records, for reasoning as well as patient record updates. This requirement is represented as a softgoal: *Time[Patient Records and Reasoning]*, as shown at the top of Fig. 4. *Time* is the type of the softgoal and *[Patient Records and Reasoning]* is the topic. This goal may be synergistic or competing with other goals, such as *Time[Research Reasoning]*, which requires fast response time for inference operations done by researchers. Using *methods* and *catalogues* of knowledge (for performance, case-based reasoning, IVF, etc.), goals can be refined into more specialized goals. Here the developer uses knowledge of the IVF domain to refine the time goal for patient information into two subgoals, 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 subgoals are connected by an *And* relationship to the parent goal. This means that if both subgoals are satisfied, then the parent goal is satisfied as well. Because, by nature, softgoals are not formalizable, we say that they are satisfied rather than satisfied. Satisficing implies good enough rather than complete solutions to a given problem.

The figure also shows an example of recording design rationale – the reasons behind design decisions – using NFR framework *claims* (Chung et al. 2000). As part of the development graph, recorded claims 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

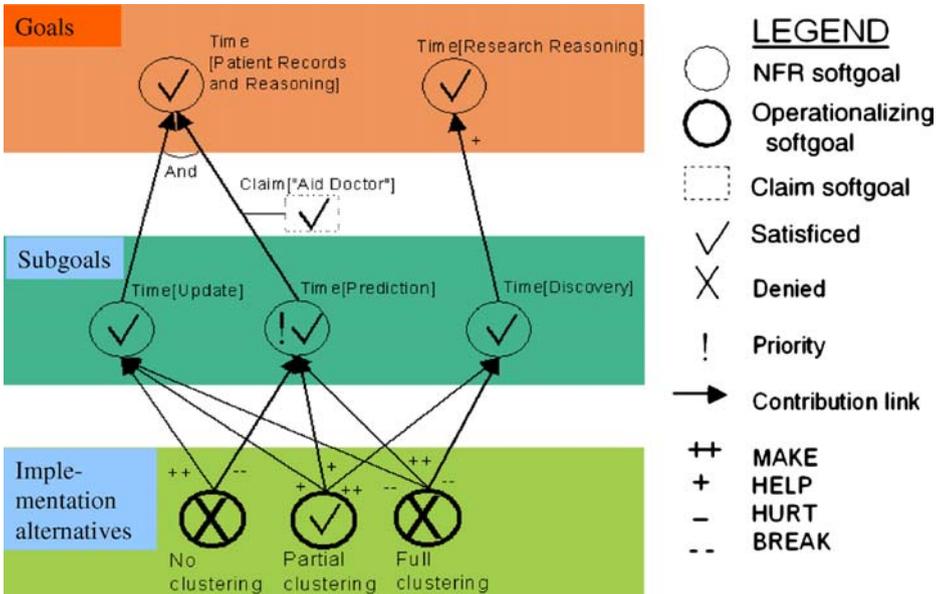


Fig. 4. Representing performance requirements for an IVF decision-support system using softgoals from the NFR framework (Chung et al. 2000). Nonfunctional requirements (initially represented as NFR softgoals) are refined until implementable solutions are identified (represented as operationalizing softgoals). Subgoals may contribute positively or negatively, to different extents, toward higher goals – MAKE means sufficiently positive, BREAK means sufficiently negative, HELP and HURT denote partial contributions. The contribution types are used to propagate the evaluation of solution alternatives toward the higher goals.

do consistency checking and keep track of decisions, but it is the developer who controls the development process (Chung et al. 2000).

3.4. Social Ontologies

A social ontology covers social settings, organizational structures or shifting networks of alliances and interdependencies (Galbraith 1973; Mintzberg 1979; Scott 2002). 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 et al. 1992). Social ontologies are also of interest in distributed artificial intelligence. Some of the concepts have been formalized using a specialized logic (Castelfranchi and Mueller 1993).

A set of concepts that focus on strategic dependencies between actors is proposed by Yu in the *i** framework for organization modeling (Yu 1993, 1995a). Such a dependency exists when an actor is committed to satisfying a goal or softgoal, carrying out a task, or delivering a resource. Using these concepts, one can create organizational models that provide answers to questions such as “Why does a fertility specialist work within a clinic rather than as an independent operator?” Creating these models enables the analysis of an organizational setting, which is an important step in the redesign of business processes and the subsequent development of information systems (Yu, Mylopoulos, and Lesperance 1996). Reasoning about the

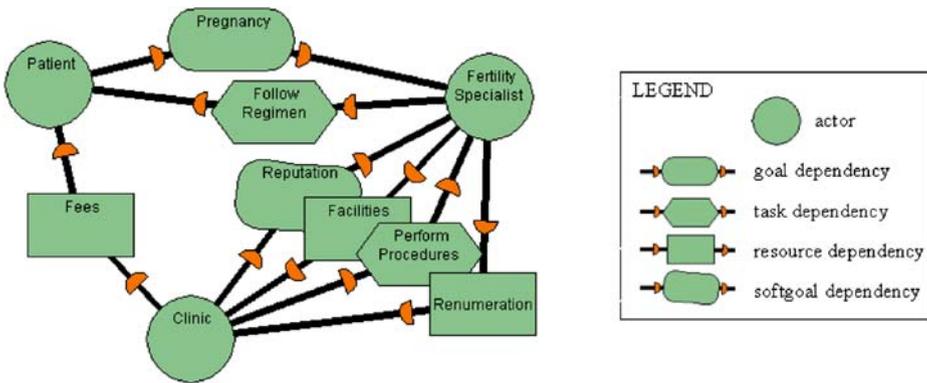


Fig. 5. Strategic dependencies between actors.

interdependency relationships among strategic actors is also important for enterprise modeling and analysis (Yu 1999).

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 assistance in organizing and discerning the many issues, and to support analysis and argumentation.

Figure 5 shows a simple example of a strategic dependency graph involving three actors: an *IVF patient*, a *clinic*, and a *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 to uphold and enhance its good reputation. The specialist depends on the clinic for facilities and remuneration, while the latter depends on the patient for the payment of fees.

One can use this kind of social ontology to model and explore alternative approaches to health care delivery. Figure 6 shows alternative arrangements by which health care costs may be covered. The patient's goal Health Costs Be Covered may be met by paying the clinic directly, by buying private insurance or by having the government pay for it via public health insurance. The patient ultimately pays for health costs because every patient is also a taxpayer. The three methods of payment are modeled as tasks in the i^* framework. While each task can meet the functional goal that Health Costs Be Covered, they contribute differently to the patient's non-functional goals. A prominent goal in this context is that the clinic's services be affordable at the time the need arises. Both insurance approaches provide timely affordability, whereas this is a main drawback of the pay direct approach. The public insurance approach contributes negatively to the tax burden, while the other alternatives are positive. This type of modeling can help generate the space of alternatives to be considered. Goals can be refined into subgoals by elaborating on the network of means-ends relationships, as illustrated in Fig. 4. The patient may look for other ways to achieve timely affordability (e.g., by maintaining personal savings) and to reduce tax burden (e.g., take advantage of tax incentive schemes). In elaborating on the reasoning structure of intentional relationships, the nature of the contribution may not be known initially. In the example of Fig. 6, the contributions of the three payment methods toward reducing overall health costs for the patient are initially

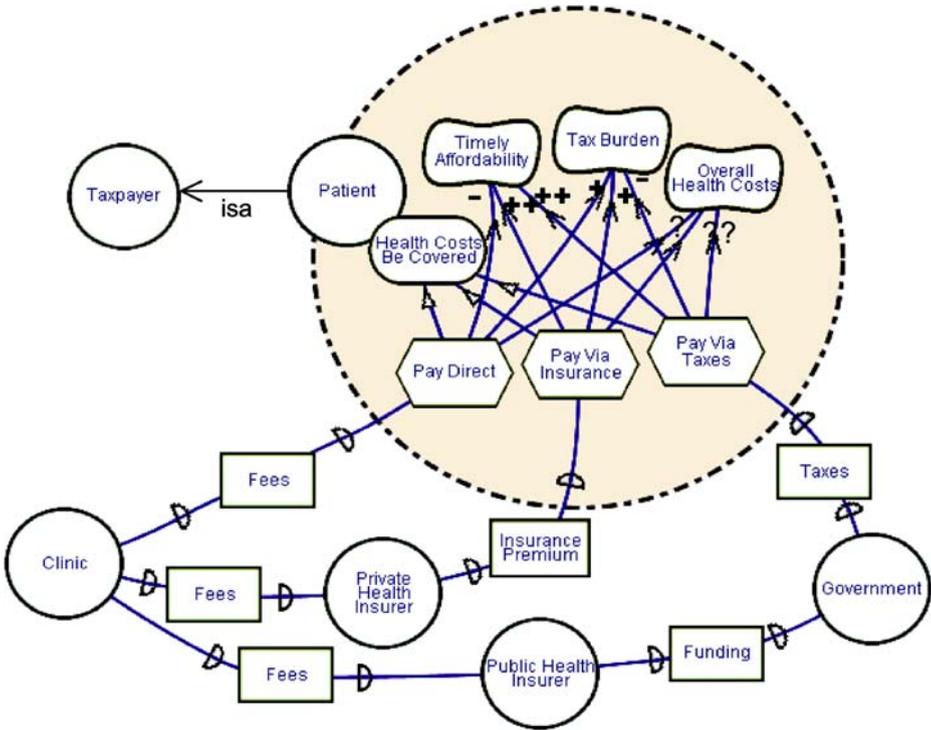


Fig. 6. Strategic rationale for patient payment alternatives.

unknown. This is because the model has not been sufficiently refined. In this case, the refinement can also be done by considering specialization on the classes of social actors – e.g., patients as belonging to high or low health care usage groups and taxpayers according to tax brackets. The model needs to be elaborated sufficiently to differentiate among alternatives, thus guiding what areas to further explore, until satisfactory solutions are found. In a social context, the exploration and consideration of alternatives need to be done from the viewpoint of each of the stakeholders. Their choices and reasoning may influence each other. For example, the government may have to balance many competing goals such as assuring a healthy citizenry, equitable access and cost containment. Again, classification and specialization may be used to differentiate how different governments and societies make different tradeoffs among goals, and thus focus on different branches when exploring alternatives. More detailed evaluation (e.g., quantitative metrics) may be introduced once the range of alternatives has been narrowed down to a manageable number. The use of a social ontology aids in identifying the right set of issues to address. The social ontology from *i** modeling concepts has been applied to requirements engineering (Yu 1997), business process analysis (Yu et al. 1996), manufacturing (Petit 1999), software processes (Briand et al. 1998), data warehouse management processes (Vassiliadis et al. 2001), analyzing trust (Yu and Liu 2001; Gans et al. 2003) and intellectual property management (Yu et al. 2001).

As an example for another social ontology intended for a different level of analysis, we may consider the action workflow approach to business process modeling

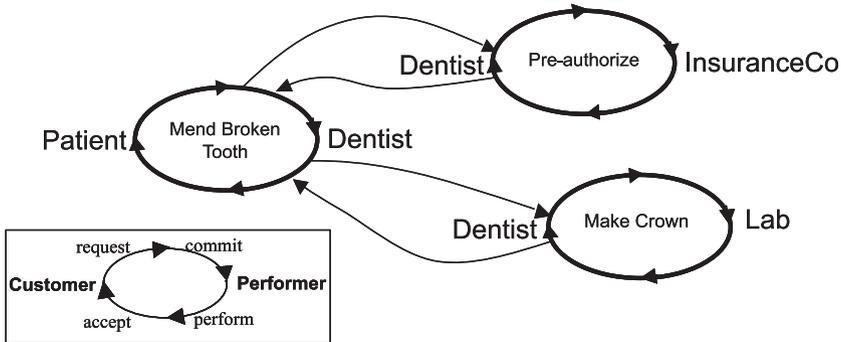


Fig. 7. An action workflow model.

(Flores et al. 1988; Medina-Mora et al. 1992; Agostini et al. 1993; Denning and Medina-Mora 1995). The focus here is on the coordination among social actors interacting in a collaborative work process. The primary unit for modeling is the cycle of events that take place between someone who wants to get something done (the customer) and the person who provides that service or product (the performer). The cycle is taken to consist of four phases: requesting, committing, performing and acceptance. Diagrammatically, a four-phased loop is drawn between the names of the customer and the performer, with the work to be done identified in the middle. Delegation and other auxiliary processes can occur at any of the four phases, depicted as branching and returning arrows from the loop. Figure 7 shows an example of an action workflow model for a dental treatment that involves a preauthorization from the insurance company during the agreement phase in the main loop and a delegation to the lab for part of the work during the performance phase.

The analysis focuses on identifying who are the relevant actors involved in a workflow, who is the customer and who is the performer for each relationship, whether the four phases are properly attended to in each customer-performer relationship and what breakdowns might occur and where. This approach differs from more conventional workflow modeling techniques that provide basically dynamic ontologies, e.g., activity sequences, synchronization, and information flow. The action workflow approach focuses instead on social mechanisms such as negotiation, commitment, coordination, and evaluation and satisfactory achievement of results. On the other hand, constructs from intentional ontologies such as goals or decision criteria, means-ends relationships and alternatives are not explicitly incorporated. A more detailed comparison with the *i** approach may be found in Yu (1995b).

4. Development and Application of Ontologies in Medicine

For a given domain, the representation of relevant knowledge can be based on primitive concepts derived from any one or all of the proposed four ontological categories. Regardless of which category is used to cover a particular application domain, information system development benefits from using tools that support ontology design and integration processes. For example, DAML and OIL define a semantic markup language for Web resources (Connolly et al. 2001; Horrocks et al. 2002). DAML (the DARPA agent markup language) has been designed as an extension of XML

(extended markup language) and RDF (resource description framework). It offers a language and tools developed to facilitate the concept of the semantic web. OIL (an Ontology Interface Layer) provides classification using constructs from frame-based AI, combined with the expressiveness and reasoning power of description logics. DAML+OIL provides modeling primitives similar to ones defined in frame-based languages and a set of constructs for creating ontologies (Fensel and Rousset 1998). Ontolingua supports authoring ontologies by providing tools for assembling and enhancing a library of modular, reusable ontologies (Gruber 1992). Once ontologies are defined for one or several domains, they may be organized into libraries, thereby enhancing their reusability (Van Heijst et al. 1995). Such libraries can be used to build information systems by supporting requirements acquisition and design (Tu et al. 1995).

When ontologies get large, tools must be used for their management. Analysis tools can help with ontology verification and validation. Verification checks whether an ontology (or any formal model, for that matter) satisfies particular constraints. For example, verification tools can check cardinality constraints for entity-relationship models or semantic consistency of rules and constraints such as “patient cannot have more embryos than she had oocytes.” Validation checks the consistency of a model with respect to its application. Because the application is informal, validation has to be done manually or, at best, semiautomatically. Other supporting tools include ontology editors, viewers and servers to support creation, maintenance and use of ontologies (Cimino 2000; Kahn 1998). For example, NEON (networked-based editor for ontologies) has been designed to standardize radiology appropriateness criteria (Kahn 1998). 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 to not only standardize terminology but also organize existing vocabularies.

On the application side, the most prevalent ontology-based activity is developing static ontologies, such as taxonomies or controlled vocabularies (Ashburner et al. 2000; Gennari and Oliver 1995; Godfray 2002; Musen 1998, 2002; Oliver 1998). The goal is to standardize terminology and taxonomically organize concepts in specific domains to enable information sharing and system cooperation. For example, an agent in a system for medical diagnosis may use an ontology of clinical concepts, both during structured data entry and decision support (Falasconi et al. 1997). A diagnostic agent may need to cooperate with a bibliographic agent that uses an ontology for bibliographies to associate literature references with particular diseases.

Over the years, many of these static ontologies turned into standard medical vocabularies, such as the International Classification of Diseases (ICD-9-CM), Systematized Nomenclature of Human and Veterinary Medicine (SNOMED), Medical Subject Headings (MeSH), Read Codes of clinical terms, Current Procedural Terminology (CPT), Unified Medical Language System (UMLS), Generalized Architecture for Languages, Encyclopedia and Nomenclatures in medicine (GALEN), etc. However, none of these standards is sufficiently comprehensive and accepted to meet the full needs of the electronic health record (Shortliffe 1998; Langlotz and Caldwell 2002; Strang et al. 2002).

Despite standardization efforts, combining and synchronizing individual versions of existing medical vocabularies remains an open problem (Oliver 1998). For this reason, the National Library of Medicine has created a UMLS (Humphreys and Lindberg 1998), which is a composite of about 60 vocabularies that contain 870,853 concepts and 2.27 million concept names in its source vocabularies (UMLS 2002 AC).

UMLS is available via a UMLS knowledge source server (UMLSKS) that provides a set of web-based interaction tools as well as an API to access the UMLS biomedical terminologies. The current UMLSKS release comprises three knowledge sources: UMLS Metathesaurus (UMLS-MT), Semantic Network (UMLS-SN) and Specialist Lexicon (UMLS-SL). UMLS-MT comprises information about biomedical concepts from diverse controlled vocabularies, as well as classifications used in patient records, administrative health data, bibliographic and full-text databases and decision-support systems. A consistent categorization of all concepts from UMLS-MT is supported by semantic types defined in UMLS-SN. The links between the semantic types represent relationships in the biomedical domain and thus provide the structure for the Network. UMLS-SL is an English language lexicon containing syntactic, morphological and orthographic information for biomedical concepts.

The Medical Ontology Group of the Italian National Research Council has been working on integrating and reusing existing terminological ontologies in medicine (Steve et al. 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 2,000 items).

Another problem that must be addressed is complexity of controlled medical vocabularies. It is important to provide tools and techniques to facilitate the design and organization of such vocabularies. Earlier models, such as ICD-9-CM, DSM, SNOMED and Read codes Version 2 (CTV2 – Clinical Terms Version 2 of the Read Codes) use the code not only to identify a concept uniquely but also to indicate where a concept lies in the hierarchy, which thus has a tree structure. As a result, a particular concept can be associated only with one node of the hierarchy. In addition, the number of levels in the hierarchy is usually limited because existing codes have a fixed number of alphanumeric characters and each character indicates a level. For example, A specifies infectious and parasitic diseases, A1 specifies tuberculosis, A130 specifies tuberculosis meningitis, while F00, in a different part of the hierarchy, specifies bacterial meningitis, and F004 meningitis-tuberculous. This results in duplication where the same concepts are represented by two different codes. Alternatively, some systems do not use codes to indicate hierarchical location, e.g., Read codes Version 3 (CTV3), the MED (Medical Entities Dictionary) and SNOMED-RT. The alphanumeric codes are a label for the concept and no longer represent the hierarchical relationship. Li et al. (2002) describes a DAG structure enrichment of the tree-structured semantic network of the UMLS. Gu et al. (1999) proposes a methodology to partition vocabularies into contexts where each context contains an isA tree hierarchy. Liu et al. (1999, 2002) shows how to partition an existing MED dictionary, which comprises 48,000 concepts, over 61,000 isA links and over 71,000 additional links (e.g., *categoryOf*, *roleOf*). Based on the partitioning into sets of concepts with the same sets of properties, MED has been implemented using a commercial object-oriented database management system (ONTOS).

Ontologies are also becoming an integral part of bioinformatics because they encourage common terminology for describing complex and evolving biological knowledge (Schulze-Kremer 1998; Stevens et al. 2002). Ontologies can be used to sup-

port a common access to diverse information repositories. One such example from biology is called TAMBIS (transparent access to multiple bioinformatics information sources; Baker et al. 1998). In this work, a uniform global schema and query interface is achieved by using an ontology (called TaO) that describes diverse bioinformatics tasks and resources (Baker et al. 1999). The TaO schema does not have materialized instances. Instead, instances are extracted from several distributed databases. Along similar lines, Boulos et al. (2002) describes the use of Dublin Core for creating a semantic medical web, to link medical web resources in a novel way to support enhanced retrieval and navigation. A step further in this direction is PharmaGKB, a frame-based ontology for pharmacogenomics (Hewett et al. 2002; Oliver et al. 2002), which was designed to support heterogeneous data integration and data acquisition. Automating data acquisition and ontology creation further requires text mining and natural language processing (Hahn et al. 2002). The trend is slowly moving into automating the process of ontology creation and using the existing ontologies during reasoning and simulation. This trend will result in development of new ontologies, especially dynamic and intentional. Application of reasoning systems in medicine will require introduction of social ontologies.

5. Discussion

Current literature on knowledge management generally agrees that the main challenges ahead lie in the realm of 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 discover, deliver and manage knowledge is enormous (Jurisica et al. 2001). Unfortunately, the pitfalls are also plentiful. This is why the complementary use of concepts and techniques from information science and information systems is crucial.

The ontological approach with an information modeling bias described in this paper derives its power from formal models of domain knowledge. Such models can be formally analyzed and processed for useful purposes. However, many domains resist precise formalization. In such domains, formalization can become a straitjacket. For this reason, formal modeling techniques need to be integrated with informal ones. For instance, the Unified Modeling Language (UML) has gained a foothold in software engineering practice, even though parts of it are informal. Informal models can help understanding a domain and bring consensus among collaborating software engineers. The challenge is therefore to blend the two approaches so that they can be used together in a seamless way.

For example, consider the design of a form. If the content of the form fields can be arbitrary text strings, then there is little to be formalized. Consequently, little computational leverage can be derived from the formalization. At the same time, such a format is highly flexible and can accommodate a broad range of inputs. However, if a field content is restricted to a finite set of predefined values, which obey specific rules, these rules can facilitate automated consistency checking, albeit at a loss in flexibility. In an e-mail message, the format of the address and date fields is formally 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 formally defined e-mail addresses. The message body, how-

ever, is arbitrary text that requires human interpretation. When one is faced with thousands of messages week after week, some kind of technology support becomes desirable.

There can be many shades in between full formalization (and the computational support it entails) and no formalization, 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 predefined 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 a scheduling conflict, when is a meeting room available, who can approve or cancel a scheduled meeting and the like.

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 information science 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.

6. 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), similarity-based retrieval and browsing (Jurisica et al. 2000), data mining and knowledge discovery (Jurisica et al. 2001), text understanding (Hahn et al. 2002; Riloff 1996), data translation services (Gruber 1993), and knowledge sharing (Orthner et al. 1994), to name a few.

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

Acknowledgements. Greg McArthur offered valuable feedback to an earlier draft of this paper.

This research has been supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Communication and Information Technologies Ontario (CITO), and the Institute for Robotics and Intelligent Systems (IRIS).

References

- Agostini A, De Michelis G, Grasso MA, Patriarca S (1993) Reengineering a business process with an innovative workflow management system: a case study. Proceedings of the Conference on Cooperative Office Computing Systems (COOCS'93), Hayward, CA, November 1993, pp 154–165
- Allen J (1984) Towards a general theory of action and time. *Artif Intell* 23:123–154
- Ashburner M, Ball CA et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* 25(1):25–29
- Baker PG, Brass A, Bechhofer S, Goble C, Paton N, Stevens R (1998) TAMBIS: transparent access to multiple bioinformatics information sources. An overview. In: Proceedings of the 6th international conference on intelligent systems for molecular biology (ISMB'98). AAAI Press, Menlow Park, CA, pp 25–34
- Baker PG, Goble CA, Bechhofer S, Paton NW, Stevens R, Brass A (1999) An ontology for bioinformatics applications. *Bioinformatics* 15(6):510–520
- Boulos M, Roudsari A et al (2002) Towards a semantic medical Web: HealthCyberMap's tool for building an RDF metadata base of health information resources based on the Qualified Dublin Core Metadata. *Set Med Sci Monit* 8(7):MT124–136
- Brazma A, Hingamp P et al (2001) Minimum information about a microarray experiment (MIAME) – toward standards for microarray data. *Nat Genet* 29(4):365–371
- Briand L, Kim Y-M, Melo W, Seaman C, Basili V (1998) Q-MOPP: Qualitative evaluation of Maintenance Organizations, Processes, and Products. *J Softw Maintenance: Res Pract* 10:249–278
- Bubenko JA (1980) Information modeling in the context of system development. Proceedings IFIP congress, pp 395–411
- Buckland M (1999) The landscape of information science: the American Society for Information Science at 62. *J Am Soc Info Sci* 50(11):970–974
- Bunge (1977) Treatise on basic philosophy: ontology I – the furniture of the world. Reidel, Dordrecht
- Carnap R (1967) The logical structure of the world: pseudoproblems in philosophy. University of California Press, Berkeley
- Castelfranchi C, Mueller J-P (eds) (1993) From reaction to cognition. 5th European workshop on modeling autonomous agents in a multi-agent world, MAAMAW '93, Neuchatel, Switzerland, 25–27 August 1993. Lecture Notes in Computer Science. Springer, Berlin Heidelberg New York, p 957
- Chu WW, Yang H, Chiang K, Minock M, Chow G, Larson C (1996) CoBase: a scalable and extensible cooperative information system. *J Intell Inf Syst* 6(2/3):223–259
- Chung L (1993) Representing and using non-functional requirements: a process-oriented approach. PhD thesis, Department of Computer Science, University of Toronto
- Chung L, Nixon BA, Yu E, Mylopoulos J (2000) Non-functional requirements in software engineering. Kluwer
- Cimino JJ (2000) From data to knowledge through concept-oriented terminologies: experience with the Medical Entities Dictionary. *J Am Med Inf Assoc* 7(3):288–297
- Conklin J, Begeman M (1988) gIBIS: a hypertext tool for exploratory policy discussion. *Trans Office Inf Syst* 6(4):281–318
- Connolly D, van Harmelen F, Horrocks I, McGuinness DL, Patel-Schneider PF, Stein LA (2001) DAML+OIL (March 2001) Reference Description
- Croner CM, Sperling J, Broome FR (1996) Geographic information systems (GIS): new perspectives in understanding human health and environmental relationships. *Stat Med* 15(17–18):1961–1977
- Curtis B, Kellner M, Over J (1992) Process modeling. *Commun ACM* 35(9):75–90
- Dardenne A, van Lamswerde A, Fickas S (1993) Goal directed requirements acquisition. *Sci Compu Program* 20:3–50
- Denning P, Medina-Mora R (1995) Completing the loops. *Interfaces* 25(3):42–57
- Falasconi S, Lanzola G (1997) Ontology and terminology servers in agent-based health-care information systems. *Meth Inf Med* 36(1):30–43
- Feather M (1987) Language support for the specification and derivation of concurrent systems. *ACM Trans Program Lang* 9(2):198–234
- Fensel DM, Rousset C (1998) Workshop on comparing description and frame logics. *Data Knowl Eng* 25(3):347–352
- Flores F, Graves M, Hartfield B, Winograd T (1988) Computer systems and the design of organizational interactions. *ACM Trans Office Inf Syst* 6(2):153–172
- Galbraith J (1973) Designing complex organizations. Addison Wesley, Reading, MA
- Gans G, Jarke M, Kethers S, Lakemeyer G (2003) Continuous requirements management for organisation networks: a (dis)trust-based approach. *Requirements Engineering* 8(1):4–22

- Genesereth MR (1991) Knowledge interchange format. Principles of knowledge representation and reasoning. Proceedings of the 2nd international conference, Cambridge, MA. Morgan Kaufmann, San Mateo, CA, pp 599–600
- Gennari JH, Oliver DE (1995) A web-based architecture for a medical vocabulary server. 19th annual symposium on computer applications in medical care, New Orleans, LA
- Giorgini P, Mylopoulos J, Nicchiarelli E, Sebastiani R (2002) Reasoning with goals. Proceedings 21st international conference on conceptual modeling (ER'02), Tampere, October 2002
- Godfray HC (2002) Challenges for taxonomy. *Nature* 417(6884):17–19
- Gotel O, Finkelstein A (1995) Contribution structures. Proceedings 2nd IEEE international symposium on requirements engineering, York, England
- Gruber TR (1992) Ontolingua: a mechanism to support portable ontologies. Technical Report KSL-91-66, Stanford University, Knowledge Systems Laboratory
- Gruber TR (1993) A translation approach to portable ontology specification. *Knowl Acquis* 5(2):199–220
- Gu H, Perl Y, Geller J, Halper M, Singh M (1999) A methodology for partitioning a vocabulary hierarchy into trees. *Artif Intell Med* 15:77–98
- Guarino N, Welty C (2000) Ontological analysis of taxonomic relations. Länder A, Storey V (eds), Proceedings of ER-2000: the international conference on conceptual modeling. Lecture notes in Computer Science, vol 1920. Springer, Berlin Heidelberg New York
- Hahn U, Romacker M et al (2002) Creating knowledge repositories from biomedical reports: the MEDSYNDIKATE text mining system. *Pac Symp Biocomput* 7:338–349
- Harel D (1987) State charts: a visual formalism for complex systems. *Sci Comput Program* 8:231–274
- Hayes P (1985) The second naive physics manifesto. Hobbs J, Moore R (eds) Formal theories of the commonsense world. Ablex, Norwood, NJ, pp 1–36
- Hewett M, Oliver DE et al (2002) PharmGKB: the pharmacogenetics knowledge base. *Nucleic Acids Res* 30(1):163–165
- Hirst G (1989) Ontological assumptions in knowledge representation. Proceedings 1st international conference on principles of knowledge representation and reasoning, Toronto, pp 157–169
- Horrocks I, Patel-Schneider PF, van Harmelen F (2002) Reviewing the design of {DAML+OIL}: an ontology language for the semantic web. Presented at 18th national conference on artificial intelligence (AAAI'2002)
- Humphreys BL, Lindberg DA (1998) The Unified Medical Language System: an informatics research collaboration. *J Am Med Informat Assoc* 5(1):1–11
- Jurisica I, Glasgow J, Mylopoulos J (2000) Building efficient conversational CBR systems. Incremental iterative retrieval and browsing. *Int J Appl Intell* 12:251–268
- Jurisica I, Mylopoulos J, Glasgow J, Shapiro H, Casper R (1998) Cased-based reasoning in IVF: prediction and knowledge mining. *Artif Intell Med* 12(1):1–24
- Jurisica I, Nixon B (1998) Building quality into case-based reasoning systems. Conference on advance information systems engineering, CAiSE'98, Pisa, Italy. Lecture notes in computer science. Springer, Berlin Heidelberg New York, pp 363–380
- Jurisica I, Rogers P, Glasgow J, Collins R, Wolfley J, Luft J, DeTitta GT (2001) Improving objectivity and scalability in protein crystallization: integrating image analysis with knowledge discovery. *IEEE Intell Syst J*, Special issue on intelligent systems in biology, November/December, pp 26–34
- Jurisica I, Rogers P, Glasgow J, Fortier S, Luft J, Wolfley J, Bianca M, Weeks D, DeTitta GT (2001) Intelligent decision support for protein crystal growth. *IBM Syst J* 40(2):394–409
- Kahn C (1998) An Internet-based ontology editor for medical appropriateness criteria. *Compu Meth Programs Biomed* 56:31–36
- Karp PD (2000) An ontology for biological function based on molecular interactions. *Bioinformatics* 16(3):269–285
- Langlotz CP, Caldwell SA (2002) The completeness of existing lexicons for representing radiology report information. *J Digit Imag* 15(1):201–205
- Lee J, Lai K-Y (1991) What is design rationale? *Hum Comput Interaction* 6(3–4)
- Lenat DB, Guha RV (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. *J Logic Program* 59–84
- Li Z, Perl Y, Geller J, Halper M, Cimino JJ (2002) Enriching the structure of the UMLS semantic network. Proceedings of the 2002 AMIA annual symposium, Nov 2002, pp 939–943
- Liu L, Halper M, Geller J, Perl Y (1999) Controlled vocabularies in OODBs: modeling issues and implementation. *Parallel Distib Databases* 7(1):37–65
- Liu L, Halper M, Geller J, Perl Y (2002) Using OOB modeling to partition a vocabulary into structurally and semantically uniform concept groups. *IEEE Trans on Knowl Data Eng* 14(4):850–866

- MacGregor RM (1993) Representing reified relations in loom. *J Exp Theoret Artif Intell* 5:179–183
- MacLean A, Young R, Bellotti V, Moran T (1991) Questions, options, criteria: elements of design space analysis. *Hum Compu Interaction* 6(3/4):201–250
- Maida A, Shapiro S (1982) Intentional concepts in propositional semantic networks. *Cognit Sci* 6:291–330
- Medina-Mora R, Winograd T, Flores R, Flores F (1992) The action workflow approach to workflow management technology. *Proceedings of the Conference on Computer Supported Cooperative Work*, pp 281–288
- Mintzberg H (1979) *The structuring of organizations*. Prentice Hall, Englewood Cliffs, NJ
- Musen MA (1998) Modern architectures for intelligent systems: reusable ontologies and problem-solving methods. *AMIA Fall Symposium*, Orlando, FL
- Musen MA (2000) Design and use of clinical ontologies: curricular goals for the education of health-telematics professionals. *Stud Health Technol Inform* 72:40–47
- Musen MA (2002) Medical informatics: searching for underlying components. *Methods Inf Med* 41(1):12–19
- Mylopoulos J (1998) Information modeling in the time of the revolution. *Inf Syst* 23(3/4):127–155
- Mylopoulos J, Chung L, Yu E (1999) From object-oriented to goal-oriented requirements analysis. *Commun 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, Würzburg
- Neches R, Fikes R, Finin T, Gruber T, Patil R, Senator T, Swartout WR (1991) Enabling technology for knowledge sharing. *AI Mag* 12(3):16–36
- Newell A (1982) The knowledge level. *Artif Intell* 18(1):87–127
- Oliver DE (1998) Synchronization of diverging versions of a controlled medical terminology. Stanford, CA, Stanford University School of Medicine, Technical Report SMI 98-0741
- Oliver DE, Rubin DL et al (2002) Ontology development for a pharmacogenetics knowledge base. *Pac Symp Biocomput* 7:65–76
- Orthner HF, Scherrer JR, Dahlen R (1994) Sharing and communicating health care information: summary and recommendations. *Int J Biomed Comput* 34(1–4):303–318
- Petit M (1999) *Formal Requirements Engineering of Manufacturing Systems: A Multi-Formalism and Component-Based Approach*. PhD Thesis, Facultés Universitaires Notre-Dame de la Paix, Dept of Computer Science, Namur, Belgium
- Potts C, Bruns G (1988) Recording the reasons for design decisions. *Proceedings of the 10th international conference on software engineering*, Singapore
- Rector AL, Bechhofer SB (1997) The GRAIL concept modeling language for medical terminology. *Artif Intell Med* 9:139–171
- Rillof E (1996) An empirical study of automated dictionary construction for information extraction in three domains. *Artif Intell Med* 85:101–134
- Rubin DL, Hewett M, Oliver DE, Klein TE, Altman RB (2002) Automating data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML. *Pac Symp Biocomput* 7:88–99
- Ruggles R (1998) The state of the notion: knowledge management in practice. *Calif Manage Rev* 40:3
- Schulze-Kremer S (1998) Ontologies for molecular biology. In: *Proceedings of the 3rd Pacific symposium on biocomputing*. AAAI Press, pp 693–704
- Scott RW (2002) *Organizations: rational, natural or open systems*, 5th ed. Prentice Hall, Upper Saddle River, NJ
- Shortliffe EH (1998) The evolution of health-care records in the era of the Internet. *MEDINFO'98*
- Steve G, Gangemi A, Pisanelli DM (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
- Stevens R, Goble C et al (2002) Building a bioinformatics ontology using OIL. *IEEE Trans Inf Technol Biomed* 6(2):135–141
- Strang N, Cucherat M et al (2002) Which coding system for therapeutic information in evidence-based medicine. *Comput Meth Programs Biomed* 68(1):73–85
- Tu SW, Eriksson H, Gennari JH, Shahar Y, Musen MA (1995) Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTEGE-II to protocol-based decision support. *Artif Intell Med* 7:257–289
- Ullman JD (1988) *Principles of database and knowledge-base systems*, vol. 1. Computer Science Press
- Van Heijst G, Falasconi S, Abu-Hanna A, Schreiber G, Stefanelli M (1995) A case study in ontology library construction. *Artif Intell Med* 7:227–255
- Vassiliadis P, Quix C, Vassiliou Y, Jarke M (2001) Data Warehouse Process Management. *Inf Syst* 26(3):205–236

- Vernadat F (1996) Enterprise modeling and integration. Chapman & Hall
- Wand Y (1989) A proposal for a formal model of objects. Kim W, Lochovsky F (eds) Object-oriented concepts, databases and applications. Addison-Wesley, Reading, MA
- Wand Y, Weber R (1990) An ontological model of an information system. *IEEE Trans Softw Eng* 16:1282–1292
- Xie H, Wasserman A et al (2002) Large-scale protein annotation through gene ontology. *Genome Res* 12(5):785–794
- Yu E (1993) Modeling organizations for information systems requirements engineering. Proceedings IEEE international symposium on requirements engineering. IEEE Computer Society Press, San Diego, pp 34–41
- Yu E (1995a) Modeling strategic relationships for process reengineering. PhD thesis, Department of Computer Science, University of Toronto
- Yu E (1995b) Models for supporting the redesign of organizational work. Proceedings of the conference on organizational computing systems (COOCS'95), 13–16 August 1995, Milpitas, California, pp 225–236
- Yu E (1997) Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. Proceedings of the 3rd IEEE Int Symp on Requirements Engineering (RE'97), 6–8 January 1997, Washington, DC, pp 226–235
- Yu E (1999) Strategic modeling for enterprise integration. Proceedings of the 14th world congress of the international federation of automatic control, 5–9 July 1999, Beijing, China
- Yu E, Liu L (2001) Modelling Trust for System Design Using the i* Strategic Actors Framework. In: Falcone R, Singh M, Tan YH (eds) Trust in Cyber-Societies – Integrating the Human and Artificial Perspectives. LNAI-2246, Springer, pp 175–194
- Yu E, Liu L, Li Y (2001) Modelling Strategic Actor Relationships to Support Intellectual Property Management. 20th International Conference on Conceptual Modeling (ER-2001). Yokohama, Japan, 27–30 November 2001. LNCS 2224, Springer, pp 164–178
- Yu E, Mylopoulos J, Lesperance Y (1996) AI models for business process reengineering. *IEEE Expert: Intell Syst Appl* 11(4):16–23

Author Biographies



Igor Jurisica is an Assistant Professor in the Departments of Computer Science and Medical Biophysics, University of Toronto, and in the Department of School of Computing, Queen's University. In addition to his position as a scientist at the Ontario Cancer Institute/Princess Margaret Hospital, Division of Cancer Informatics, Igor holds a Visiting Scientist position at the IBM Canada Toronto Laboratory. He is recognized for the work in computational biology, including representation, analysis and visualization of high-dimensional data generated by high-throughput biology experiments.

Igor is particularly interested in the use of comparative analysis for mining different dataset types, including protein-protein interaction, gene expression profiling and high-throughput screens for protein crystallization. He is currently a member of the Research Management Committee of Precarn. Igor has published over 70 articles in journals, books, conferences, workshops and

symposia, and jointly holds a US patent related to data mining of large repositories.



John Mylopoulos is Professor of Computer Science and Bell Laboratories chair of Information Systems at the University of Toronto. John's research interests include conceptual modeling, requirements engineering, data semantics and knowledge-based systems. John was a corecipient of the most influential paper award at the International Conference on Software Engineering (1994), is the president of the VLDB Endowment, is also a fellow of the AAAI. He has served on the editorial board of the *ACM Transactions on Software Engineering and Methodology* (TOSEM), the *ACM Transactions on Information Systems* (TOIS). He is currently program cochair of the 2nd Semantic Web Conference (2003), program cochair of the 5th Web Information Systems Engineering (2003), and general chair of the Very Large Databases Conference (2004).



Eric Yu is Associate Professor at the Faculty of Information Studies, University of Toronto. He received his Ph.D. in Computer Science from the University of Toronto in 1995. His interests are in the areas of information systems design, software engineering, and knowledge management. His research emphasizes concepts and techniques for modeling and analyzing strategic relationships among social actors. His publications include more than 50 articles in journals, books, and conference and workshop proceedings, and the book *Non-Functional Requirements in Software Engineering* (2000). He has served on many committees in the areas of information systems, requirement engineering, and agent-based systems, and is a founding cochair of the International Workshop series on Agent-Oriented Information Systems. Earlier, he held positions in hardware, software and services development at the technology division of Nortel Networks in Ottawa.

Correspondence and offprint requests to: Eric Yu, Faculty of Information Studies, University of Toronto, St. George Street 140, Toronto ON M5S 3G6, Canada. Email: eric@cs.toronto.edu