Handbook of Software Engineering and Knowledge Engineering

EXPLORING ONTOLOGIES

YANNIS KALFOGLOU
School of Artificial Intelligence, University of Edinburgh, 80 South Bridge
Edinburgh EH1 1HN, Scotland
Email: yannisk@dai.ed.ac.uk

Received June 1, 2000
Revised September 27, 2000
Accepted (accepted date)

Abstract

Ontologies are studied by many scholars with diverse backgrounds and are applied in a variety of contexts and application areas. Despite the numerous reviews published there are many issues which still remain unclear with respect to their cost-effective deployment, identification of tradeoffs, maintenance strategies and ways of integration. Furthermore, there is no report which refers to all those issues along with the basic information in order to be used as a road-map. This survey article aims to provide such information in a manner which will help the interested software practitioner to comprehend the basic principles in ontology design, understand their strengths and weaknesses, be aware of a variety of areas where ontologies have been successfully applied, and identify tradeoffs and potential solutions.

Keywords: Ontologies, knowledge sharing and reuse, knowledge management, software design.

1. Introduction

In this survey article we are exploring ontologies with emphasis on their deployment in a wide variety of areas ranging from software design to knowledge management and information retrieval. We are not interested to provide an in-depth analysis of the field of ontologies in an isolated manner nor to provide a methodological approach for guiding the software design processes. Rather, we scrutinise the ontologies field as practised, mainly, in the knowledge engineering community over the last decade, and report on their impact in software design through example application cases, worked projects, and emerging experimental results.

Before we proceed with our survey, however, we look closer at software design. Software design is still a young field, and we are far from having a clear articulation of the relevant principles. Winograd offers a parallelism of the phrase with software engineering(hereafter, SE): "...is often used to characterise the discipline that is also called software engineering - the discipline concerned with the construction of software that is efficient, reliable, robust, and easy to maintain."[108]. Although work has begun in engineering software design with the emergence of methodological approaches[78], guidelines, and bringing in rationale[68], there is still an area that remains unexplored: bringing into the design process explicit knowledge regarding the domain on which the system to be developed will operate.

The study and modelling of that knowledge is a core theme in artificial intelligence(hereafter, AI) research. Having their roots in knowledge representation, knowledge engineering methods and techniques gave AI researchers a powerful tool for transforming contextual knowledge into machine-readable form to enable mechanised reasoning about a domain of interest. Ontologies are such a form of domain knowledge. We should also note the similarity of this topic with such areas as domain analysis[4] and engineering(this volume, J.L.Diaz-Herrera, chapter ??), object oriented patterns[30], etc. In this chapter, however, we are interested in ontologies as practiced, mainly, in the AI community.

Nowadays, they are studied by many scholars who belong to different communities. Hence, a plethora of articles and field reviews are available for the interested practitioner. However, each of them is focussed on a specific area: for example, the Uschold and Gruninger review - one of the first comprehensive reviews published - is concerned with design principles and methodological ways of construction with selective references to exemplar applications[93]; the Fridman-Noy and Hafner review further explores and compares design methods[29], the Chandrasekaran and colleagues article provides a general overview of the field[13]; and Gomez-Perez and Benjamins devote half of their report to give a catalogue-style information on ontologies research[34]. Despite the bulk of resources available it is still difficult for practitioners, especially those with a SE background, to locate and elicit the right information with respect to engineering, application, and cost-effective issues. Most of the times this information is found in different resources.

This survey aims to fill-in this gap. In order to do this effectively, we explore the field from the following angles: design, deployment, and tradeoffs. We explore design issues in sections 2 to 6 where we describe what an ontology stands for, ways of design, the role of ontological commitment, methodologies to follow when building ontologies, and explain the various types reported in the literature. Deployment issues are described in sections 7 to 8 with emphasis on uses of ontologies, ways of deployment, and references to applications and influential projects from both industry and academia. Lastly, we conclude our survey by discussing potential problems, tradeoffs and solutions proposed and used, in section 9, followed by list of pointers to resources for further reading, in section 10.

2. Definitions

We start our review by explaining what an ontology stands for. Although a single definition will usually suffice, ontologies have a peculiar characteristic: there are a number of different definitions proposed and used. Even nowadays there are people who argue about the actual meaning of the term. A reason for this is, probably, the fact that ontologies are studied, developed and applied by people with diverse backgrounds and interests. We do not subscribe to this pointless debate over the meaning of the term in this article nor we will introduce yet another definition. Rather, we briefly review the most commonly used definitions found in the literature in order to explain what an ontology stands for.

One of the early definitions appeared in [72]. The authors define an ontology as: "the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary". This definition introduced the idea that ontologies can be viewed linguistically, as extensible vocabularies regarding a topic area. In the context of knowledge sharing, Gruber offered a short definition which became the most widely cited in the literature: "an ontology is an explicit specification of a conceptualisation"[37]. This definition was further enriched by Borst and his colleagues in [10], where they argued that the specification is actually formal and the conceptualisation is shared. Studer and colleagues analysed the terms used in the definition and provide the following explanation: "Conceptualisation refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared refers to the notion that an ontology captures consensual knowledge, that is, it is not primitive to some individual, but accepted by a group"[85]. Uschold offers a working definition which hints at the purpose of having ontologies: "An ontology is virtually always the manifestation of a shared understanding of a domain that is agreed between a number of agents. Such agreement facilitates accurate and effective communication of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing"[90]. Others, consider ontologies as domain theories[25], as vocabularies[83], as standards[58], etc. In [43] the authors offer a clarification of terminological issues regarding the various definitions founded in the literature. In [88], the authors relate an ontology to a knowledge base in their definition: "An ontology provides the basic structure or armature around which a knowledge base can be built".

Based on the definitions quoted above we summarise what an ontology stands for: an explicit representation of a shared understanding of the important concepts in some domain of interest. The role of an ontology is to support knowledge sharing and reuse within and among groups of agents(people, software programs, or both). In their computational form, ontologies are often comprised by definitions of terms organised in an hierarchy lattice along with a set of relationships that hold among these definitions. These constructs collectively impose a structure on the domain being represented and constrain the possible interpretations of terms.

3. Design principles

A number of design criteria have been proposed, originally analysed in [38]. For a thorough analysis of these criteria we point the interested reader to the aforementioned citation whereas here we briefly recapitulate them. The criteria proposed are: *clarity*, *coherence*, *extendibility*, *minimal encoding bias*, and *minimal ontological commitment*. *Clarity* means that the intended meaning should be communicated effectively. This means that ambiguity should be minimised, distinctions should be motivated, and examples should be given to help the reader understand definitions that lack necessary and sufficient conditions. When a definition can be stated in logical axioms, it should be. Where possible, a complete definition(a predicate defined by necessary and sufficient conditions) is preferred over a partial definition(defined by only necessary or sufficient conditions). All definitions should be documented with natural language. *Coherence* means that the ontology should be internally consistent. At the least, the defining axioms should be logically consistent. Coherence should also apply to the concepts that are defined informally, such as those described in natural language documentation and examples. *Extendibility* means that one should be able to extend the existing terms in a way that does not require the revision of existing definitions. The next two criteria help to achieve that. The *encoding bias* and *ontological commitment* should be minimal. An *encoding bias* results when representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimised, because knowledge-sharing agents may be implemented in different representation systems and styles of representation. *Minimal ontological commitment* means that the ontology should make as few claims as possible about the world being modelled, allowing the parties committed to the ontology freedom to specialise and instantiate the ontology as needed. While making too many ontological commitments can limit extensibility, making too few can result in the ontology being consistent with incorrect or unintended worlds(i.e., models). For this reason, it is beneficial to make ontological commitments with respect to aspects intrinsic to a domain.

We should note that the above criteria are not always possible to meet by ontology designers. A number of tradeoffs have been identified[38], and ways of compromising between well designed ontologies and applicability have been investigated [10]. We will not expand on this issue in this article because it is peripheral to our topic: uses of ontologies. To support this we shift our attention to the notion of ontological commitment which plays an important role in using ontologies in software systems.

4. Ontological commitment

Ontological commitment refers to agreement on the use of the shared vocabulary by the agents commited to the ontology in question. When we say that an agent commits to an ontology we mean that its observable actions are consistent with the definitions in the ontology[38]. It has been said that commitment to a common ontology is a guarantee for consistency but not for completeness, with respect to queries and assertions using the vocabulary defined in the ontology[38].

Guarino describes the role of ontological commitment in software: "ontological commitment should be made explicit when applying the ontology in order to facilitate its accessibility, maintainability, and integrity. This will lead to an increase of transparency for the application software which based on that ontology"[40]. These commitments are often encoded as axioms that enforce the syntactic consistency of the definitions used. Practically, an ontological commitment is an agreement to use a vocabulary(i.e., ask queries and make assertions) in a way that is consistent with respect to the theory that specifies the ontology. We build agents that commit to ontologies and we design ontologies so we can share knowledge with and among these agents. With a declarative specification, we can explicitly reason about different ontological commitments. For example, we can compare two different proposals for an ontology with respect to the classes of objects that they require and the properties and relations among these objects that they postulate[93].

Guarino and colleagues argue for a greater role of ontological commitment. In [42], the authors continue, an ontological commitment should capture and constrain a set of conceptualisations. They propose a formalisation of ontological commitments which: "offers a way to specify the intended meaning of [a logical language] vocabulary by constraining the set of its models, giving explicit information about the intended nature of the modelling primitives used and their a priori relationships". The work of Guarino and colleagues is focussed on the design phases of ontology. Other scholars' work aim on the deployment of ontological commitments in applications. We will describe this work in section 8 where we summarise uses of ontologies in software design.

5. Methodologies

The construction of an ontology is a time-consuming and complex task. Although there are no standards to obey when building an ontology, various design guidelines and methodological approaches have been proposed and used. In particular, in a comprehensive review of the field[93] the authors report on two methodologies used in the context of the *Enterprise* ontology[97] and the TOVE project[20]. In the former, a skeletal methodology has been proposed[98] which identifies five main steps: (a)identify purpose and scope, (b)build the ontology, (c)evaluation, (d)documentation, and (e)guidelines for each phase. Step (b) is further divided into ontology capture, coding, and integration of existing ontologies. This skeletal methodology was used in the construction of the *Enterprise* ontology but does not explicitly deploy a formal evaluation procedure. This was the main focus of the methodology used in the context of the TOVE project[20]. In particular, Gruninger and Fox used a formal methodology that supported evaluation of the ontology using the notion of *competency questions*[39]. The underlying philosophy is to define a set of queries that the ontology can answer. These queries help to assess the ontology's competence. They evaluate the expressiveness of the ontology which is required to represent these questions and characterise their solutions. These queries are drawn from a number of motivating scenarios which are story problems or examples which are not adequately addressed by existing ontologies.

Apart from the work on evaluation and construction methodologies by Uschold, Gruninger and colleagues, others have focussed on the preliminary phases of construction. In [24] the authors presented a system, called *METHONTOLOGY*, which provides support for the entire life-cycle of ontology development. A distinguishing characteristic of the METHONTOLOGY framework is that it is tailored to support the early phases of development by employing the notion of *intermediate represen-*

tations. These are representations independent of the implementation language in which the ontology will be developed. The system that support the use of these representations is the Ontology Development Environment(ODE)[9]. An overview of methodologies used in AI projects along with a comparison with standards from SE literature is given in [23].

6. Types

The development methodologies reported above were used in some of the ontologies which will be described in sections 7 and 8. Before we proceed to survey actual implementations of ontologies we describe various types of them as reported in the literature. Ontologies can be classified in terms of genericity. For example, broad ontologies like CYC[57], model generic notions that forms the foundations for knowledge representation across various domains. These are also called top-level ontologies[13], like Sowa's ontology[84]. On the other hand, small-scale, domain-specific ontologies are carefully tailored to the domain at question. Examples of this type are the *PhysSys* ontology[10] which captures knowledge regarding physical system processes, the *AIRCRAFT* ontology [100] used to represent air-campaign planning knowledge, the *PIF* ontology[55] used for business process modelling, etc.

Another classification of ontologies is concerned with their purpose. There exist *task* ontologies[67] that capture task-related knowledge independently of the domain that the task is defined. Complementary to these are the *method* ontologies[12] which provide definitions of the relevant concepts and relations used to specify a reasoning process to achieve a particular task. A specific type of ontologies is the *knowledge representation* ontologies. The most representative example is the Frame ontology[37] which captures the representation primitives used in frame-based languages. It allows other ontologies to be specified using frame-based conventions, as implemented by the Knowledge Interchange Format(KIF)[33].

Most ontologies, however, are placed under the tag *domain* ontology. These are designed to support a specific domain and applications defined within that domain. For example, the *PIF* ontology is concerned with the business process modelling domain and supports the exchange of information among a variety of business process modelling applications.

There is another type of ontology, the *linguistic* ontologies. The most illustrative examples are the Generalised Upper Model(GUM)[7], WordNet [66], and SENSUS[54]. However, these usually have the form of a vast collection of terms which led to another classification with regard to the level of formality. These sort of ontologies are often called "terminological" ontologies whereas ontologies like TOVE are called "axiomatised" ontologies.

In their overview of the field, Uschold and Gruninger identified the following types with respect to the degree of formality: *highly informal*, *semi-informal*, *semi-formal*, *rigorously formal*[93]. In the informal cluster we see definitions in natural language or at most in a structured form of natural language. In the formal cluster we have ontologies implemented in an artificial formal language(i.e., *Ontolingua*), or in first order theories with formal semantics, theorems and proofs of such properties as soundness and completeness(i.e., TOVE).

7. Engineering

Although many argue that engineering of ontologies is still in its infancy the first comprehensive reports covering all aspects of ontology construction and deployment began to emerge few years ago. We selectively report here some of these efforts by highlighting their contributions to the field. In an experiment of ontology reuse[92], researchers working at Boeing were investigating the potential of using an existing ontology for the purpose of specifying and formally developing software for aircraft design. The application problem addressed was to enhance the functionality of a software component used to design the layout of an aircraft stiffened panel. They describe a start-to-finish process that used an existing ontology, residing on the Ontolingua [21] server, the *EngMath*[36] ontology, which was then translated to the target specification language and integrated to an engineering software component. They then executed that component and demonstrated the benefits of reusing an existing knowledge component in the development process. The lessons learned from that experience is that ontology reuse can be pursued on a large scale and, under certain circumstances, it can be a cost-effective approach. We will revisit the tradeoffs identified by Uschold and colleagues in their experiment in section 9 while we continue here by reporting two studies that were focussed on the whole spectrum of engineering ontologies: the *AIRCRAFT* project, and the *PhysSys* project.

In [100] the authors describe how they achieved reuse among ontologies themselves. The resulted ontology, *AIRCRAFT*[a] , contains knowledge about types of US military aircraft, including data about the engines, PODs, and fuel tanks that these aircraft can carry. The distinguishable feature of this ontology is how it has been developed in the first place. The process, which is described in [87], was based on the use of a large-scale, linguistic ontology, the *SENSUS*[54]. A characteristic of *SENSUS* is that it is actually constructed from extracting and merging information from existing electronic resources(like the *WordNet*, dictionaries, GUM ontology). The authors, used this broad coverage ontology and then devised a semi-automatic method which made it possible to identify terms in the original ontology that were relevant to their particular domain, and then pruned the ontology so that it included only those terms. In addition, they enhanced the newly emerged ontology with terms tailored to the domain of air campaign planning. These were military terms. The resulting ontology, *AIRCRAFT*, is accessible through an ontology development environment, the *ontosaurus* browser which supports the idea of "ontology developed collaboratively by the system developers themselves"([87]).

In [10], a general and formal ontology, called *PhysSys* is presented. It covers the domain of dynamic physical systems and it is composed of seven different ontologies. This work explored a new idea in ontology engineering, that is *ontology projections*:

[a] A demonstration version is electronically available from the URL: http://www.isi.edu/isd/ontosaurus.html

"a flexible mechanism to link and configure ontologies into larger ones." Three kinds of projections demonstrated in the paper, *include-and-extend*, *include-and-specialise*, and *include-and-project*. The latter was used to link an ontology developed by the group of *PhysSys* authors to an outsourced ontology, the *EngMath*. The *PhysSys* ontology was used as the foundation for the conceptual database schema of a library of reusable engineering model components, the *OLMECO* library. The library was evaluated by modelling and numerically simulating the existing heating system of a general hospital in Schiedan, the Netherlands[10].

In the context of the *Plinious* project[101], the bottom-up method in ontology development is discussed[102]. In contrast with the majority of approaches in ontology construction which fall into two categories, top-down and middle-out(analysed in [93]), the bottom-up way "proposes to lay down the meaning of complex concepts by means of primitive meaning constituents." It has been applied to the domain of ceramic materials and covers their properties and the processes to make them. It was found that this approach was suitable for such a domain because, the authors continue, it is impossible to exhaustively predict in advance which concepts will be needed to express the knowledge found in the texts. As this domain covers chemical substances, it was argued that listing all these substances is an open-ended task. As such, keeping track of the regular updates in a top-down designed ontology was impractical since it requires substantial effort and is error-prone. Consequently, the approach used supports reasoning along two orthogonal hierarchies: "the partonomy formed by substances and their constituents and the taxonomy formed by concepts and superconcepts"[102].

Other projects which provide an insight in the engineering process are the re-engineering effort of implemented ontologies, described in [35], and the collaborative effort in developing a common ontology for the knowledge acquisition community[8]. In particular, Gomez-Perez and Amaya describe a re-engineering process of retrieving and transforming a conceptual model of an existing ontology into a new one. The work of Benjamins and Fensel describes the *Knowledge Annotation Initiative of the Knowledge Acquisition Community* ontology(in short, KA^2), which models the knowledge acquisition community and forms the basis to annotate its documents on the Web[b] in order to enable intelligent access.

8. Applications and projects

A complete listing of applications of ontologies is impossible. The literature references are huge and citing lengthy lists is not practical. However, we provide pointers to various resources in section 10 whereas here we selectively report the most representative ones. To do this effectively we cluster them according to their application domain.

We start with the area of *enterprise modelling*. In this area we found the *Enterprise* ontology[97], which captures the organisational structure of an enterprise with

[b]The ontology is accessible online from the following URL: http://www.aifb.uni-karlsruhe.de/WBS/broker/KA2.html

emphasis to activities and processes. The ontology is developed in a structured text form and a translation in *Ontolingua* is also available. In the same line is the TOVE ontologies set[20] which shares the same aims with *Enterprise*, but has been developed in a formal computational form and uses different underlying principles[28]. The differences between these two representative ontologies for enterprise modelling are highlighted in [93].

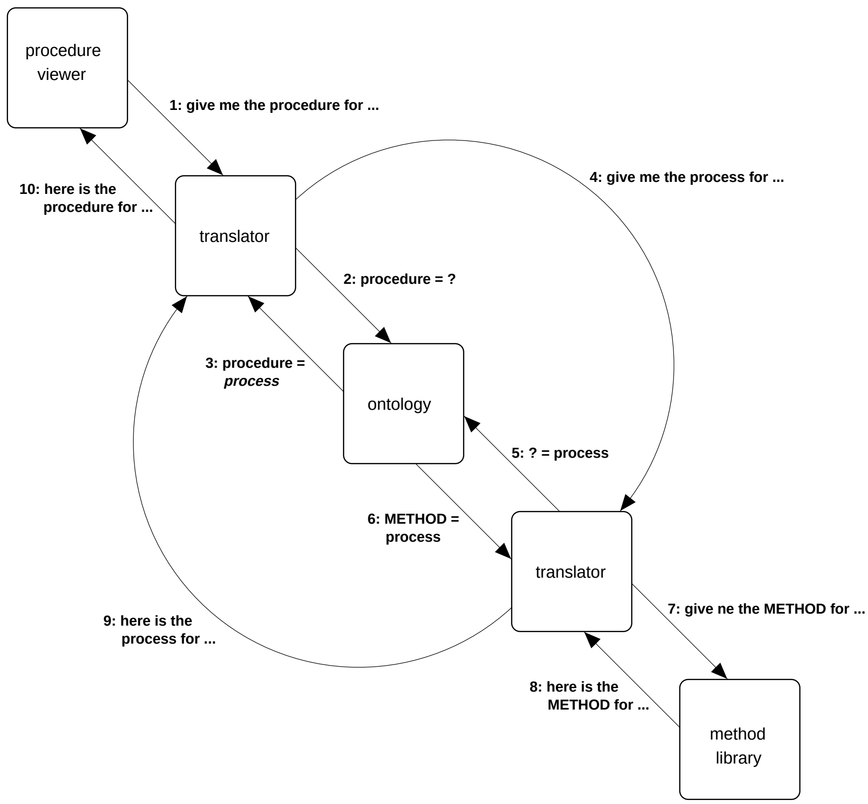

Figure 1: Ontology as inter-lingua: Example taken directly from [93]. This illustrates the use of an ontology as an inter-lingua to integrate different software tools. The term procedure, used by one tool is translated into the term, *method* used by the other via the ontology, whose term for the same underlying concept is *process*.

A relevant application area is that of *business process modelling*. The Process Interchange Format(PIF)[55] is among the best known in this area. The aim of PIF is to develop an interchange format to help automatically exchange process descriptions among a variety of business modelling and support systems such as workflow software, flow charting tools, planners, process simulation systems and process repositories. The core of PIF consists of the minimal sets of constructs necessary to translate simple but non-trivial process descriptions. In addition, PIF can be extended to represent local needs of individual groups with the use of Partially Shared Views(PSV) described in [56]. The PIF framework has been applied in a *supply chain scenario*[77] which was adopted from the Workflow Management Coalition(WfMC)[107]. An example of an interchange format is illustrated in figure 1.

Ontologies have also been applied to *medical applications*. For example, a methodology for integrating medical terminologies was presented in [31]. This is the aim of the *ONIONS* methodology[32] developed by the same group. In the same context, the European project *GALEN*[c] [80] which aims at capturing information from the clinical domain. In [1], the authors present a system, called *Sophia* which acts as a knowledge server for web-based medical applications. An ontology for bioinformatics(*TAMBIS*) is presented in [6]. Most of the applications in this area are based on terminological resources like the *GUM* ontology [7], the *CYC* ontology [57], the *Unified Medical Language System(UMLS)* [71], etc.

Another area to which ontologies have been applied is that of *ontology-based brokering*. These are specifically designed agent systems which serve as brokers between heterogeneous systems. They use ontologies to facilitate the information brokering task. Representative applications are: the *Ontobroker*[17] which was used, among others, in the KA^2 project[8]; the *onto2agent*[5] used to select publicly available ontologies on the web for a given application based on a *Reference Ontology* developed by the same group to classify candidate ontologies; the *OBSERVER*[63] system used to provide semantically rich information to a user who subscribes to an information management system on the web which supported by selected ontologies; the *IMPS(Internet-based Multi-agent Problem Solving)*[16] system which uses software agents to conduct knowledge acquisition on-line using distributed resources. Terminological ontologies(like *WordNet*) were used to underpin the whole process.

A related area of applications is that of *knowledge retrieval*. A representative application in this area is the *PlanetOnto* which provides an integrated set of tools to support news publishing based on ontology-driven document enrichment[19]. To support this project two ontology-specific tools were developed: the *Tadzebao* and *WebOnto* both described in [18]. The former aims to support a dialectical approach in ontology design and maintenance while the latter provides editing and browsing facilities. The goal of *Tadzebao* is to provide guidance for knowledge engineers around ongoing dialogues for designing ontologies. This can be used as a negotiation tool for proposed changes in an ontology with the additional flexibility that *Tadzebao* offers: the integration of discussion about an artefact and its representation in the same visual metaphor. Another application in this area is the *knowledge-enhanced search* approach used in the *FindUR* project[62]. McGuinness describes a search tool, deployed at the AT&T research labs, which uses ontologies to improve the search experiences from the perspectives of recall and precision as well as ease of query formation. A similar approach which deploys *content matching* techniques is described in [44] where the authors present the *OntoSeek* system designed to

[c]The project is electronically accessible from the URL: http://www.cs.man.ac.uk/mig/galen

support content-based access to the web.

In the broader context of *knowledge management*(hereafter, KM) ontologies are useful to support crucial KM tasks and activities. For an overview of the field with emphasis on the role of AI in KM we point the interested reader to O'Leary's review in [74]. Here, we will use O'Leary's thesis that the goal of KM is to create valuable information by employing the so called, *converting and connecting* processes([75], in order to identify the role of ontologies in KM. The processes identified were: convert (i)individual to group knowledge, (ii)data to knowledge, (iii)text to knowledge, and connect (iv)people to knowledge, (v)knowledge to knowledge, (vi)people to people, and (vii)knowledge to people. We argue that ontologies could be used in most of these processes, either by playing a major role or by supplying the supporting infrastructure that helps an organisation to implement them. In the following paragraph we mention indicative examples from the ontology research literature to justify this claim.

In particular, ontologies provide part of the infrastructure for conversion processes(i to iii as listed above) and help in the connection activities(iv to vii as listed above). Conversion processes (i) seem to benefit more from the presence of ontologies as this is the underlying principle in their construction. Methodological[93] and collaborative approaches([87],[8]) in ontology building, convert individual to group knowledge in the form of an ontology. Processes (ii) and (iii) use other AI technology like data and text mining techniques with ontologies being the guide to the 'right' data or text repository[17]. Ontologies seems to be more helpful in the connecting processes. Process (iv) is concerned with the so called, 'pull' technology, which aims at pulling knowledge residing in vast repositories to people. The means which used to pull that knowledge are, mainly, search engines and intelligent agents. Examples of ontology use in this area are given in [62] and [44]. Process (v) actually highlights the main contribution of ontologies: enabling communication and interoperability between systems. The best way to cite indicative work here is to point to reviews and collections such as [93] and [41]. Process (vi) is not directly related to ontologies as it is more concerned with technological means such as Intranets. However, we should mention the work on collaboration and discussion aided by ontologies[86]. In contrast with process (iv), process (vii) is concerned with 'push' technology. Means to achieve this are designated systems that focus on content and push knowledge to the user instead of waiting for the user to pull out that knowledge. As in (iv), ontologies play a major role here since they are concerned with content and semantically enriched information. Example uses are described in [22] and [17].

Finally, after having presented the processes that help to achieve the goal of KM we close this section on KM by describing main KM tasks and activities and how ontologies are related to them. These are summarised in figure 2 and described in the following paragraph.

On the right hand side of figure 2, we illustrate the main KM tasks and activities. We identify four main KM tasks: *acquiring*, *analysing*, *using*, and *preserving* knowl-

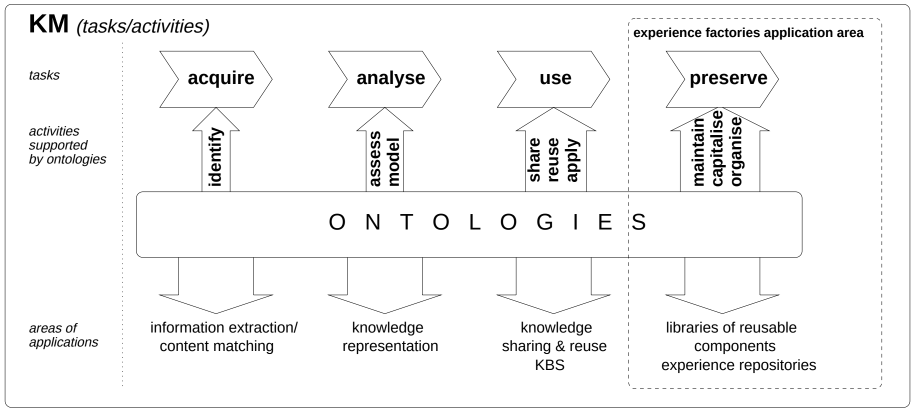

Figure 2: Ontologies in knowledge management: activities supported by ontologies help to achieve knowledge management tasks resulting in a range of application areas.

edge. We argue that these tasks are accomplished by activities which are supported by ontologies. In particular, the knowledge acquisition task, is accomplished by *identifying* activities which are supported by ontologies. This results in the application area of information extraction and/or content-matching. In the same manner, ontologies in the area of knowledge representation are used to *model* and *assess* the environment, which are activities employed in the *analysing* knowledge task. The *using* knowledge task, includes the *apply*, *share*, and *reuse* activities, which are supported by ontologies with such application areas as knowledge sharing and reuse, and KBSs. The last task of the KM tasks/activities diagram is *preserving* knowledge. It is accomplished by activities such as *organising*, *maintaining*, and *capitalising* which are partially aided by ontologies. The resulting application area is that of libraries of reusable knowledge components and experience repositories. The knowledge preservation task and its accompanying activities along with the relevant ontologies are the area of overlap with experience factories[d] as denoted by the box surrounding the task in figure 2.

The last area to report is that of *systems engineering*. In section 7 we already described systems like the *AIRCRAFT* and the *Boeing* experiment with the use of the *EngMath* ontology which was also used in the construction of the *PhysSys* ontologies set. Other representative applications are the *ATOS(Advanced Technology Operations System)*([47]) system which was designed to meet specific needs of *spacecraft operations* such as the need for coordination of different agent applications who had to commit to a common ontology. In [26], the authors describe the *Integrated Development Support Environment(IDSE)*, a commercial computational

[d]Kalfoglou and Robertson investigate the overlap of ontologies and experience factories in [52].

environment that supports the integration of enterprise models. The integration is underpinned by axioms representing semantic constraints and relationships between different tools which are interpreted and enforced semi-automatically. This information is contained in a method ontology, the *IDEF1X*[e], accessed by a truth maintenance system that enforces rules and constraints defined in the method.

The use of ontological axioms has been inspirational and many researchers are investigating the practicality of deploying ontologies in *software design*. Gruber hints the role of these axioms in ontology deployment:

> "Ontologies are often equated with taxonomic hierarchies of classes, class definitions, and the subsumption relation, but ontologies need not be limited to these forms. Ontologies are also not limited to conservative definitions, that is, definitions in the traditional logic sense that only introduce terminology and do not add any knowledge about the world. To specify a conceptualisation one needs to state axioms that do constrain the possible interpretations for the defined terms."[37]

A working example for the use of ontological axioms in software design is described in [53]. The authors point out that the role anticipated by ontological axioms is rarely delivered: to restrict the possible interpretations ontological constructs could have. To operationalise this role and enforce it in an integrated development environment they invented a **multi-layered** architecture in which ontological axioms are separated from other ontological constructs included in a system that uses the underlying ontology. These are enforced to comply to the axiomatisation in order to verify the consistency of the system with respect to domain knowledge as explicitly represented in the underlying ontology[50]. Ultimately, this layered metaphor can be extended to check the ontological axioms themselves against another set of axioms, meta-axioms, which could come from another ontology. This facilitates the conformance check of an application to ontology and can be extended to check dependencies among ontologies themselves[51]. Moreover, it supports the integration of ontologies in applications while preserving their identity as being a separate layer in the multi-layer architecture. The approach is illustrated in figure 3.

One of the early contributions that used ontological commitments was that of the `Comet`[61] and `Cosmos` [60] systems. Both systems aim at developing knowledge bases by capturing the set of ontological commitments that define the interdependencies among key terms in the ontology. Their role is to assess the impact of changes in their world and provide context-specific guidance to their users on what modules may be relevant to include in the design, and what design modifications will be required in order to include them[59]. The key idea behind this work was to make use of the ontological commitment expressed by the underlying ontology in the system's development process.

Similarly, in the **DISCOVER** project[106], the role of ontological commitment

[e]Electronically accessible from the URL: http://www.idef.com/overviews/idef1.html

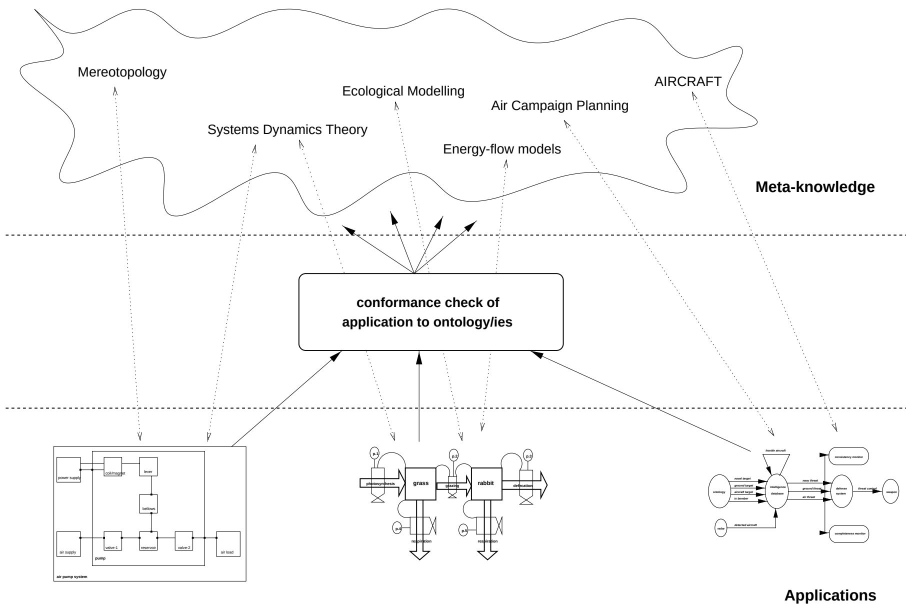

Figure 3: The multi-layer approach: it enforces the conformance check of an application to ontology/ies. Applications are using meta-knowledge constructs from various ontologies(mereotopology, system dynamics theory, etc.) in an integrated environment which enables checks on the use of those constructs against their axiomatised definitions.

was further analysed and operationalised. The authors state that ontological commitment is a key issue for knowledge sharing and reuse and they applied existing verification techniques from the KBSs literature to check the commitment of a knowledge base to an ontology. In that project the role of the ontology was to act as a background body of knowledge against which a knowledge base can be validated.

There are also a number of applications related with projects undertaken by various organisations involving academic and industrial partners. We already mentioned some of them in the previous sections. We complete our coverage here by describing one of the first projects in this area which was the *Knowledge Sharing Effort(KSE)*[72] aimed to realise the benefit of sharing and reusing large knowledge bases. The distinguishable contribution of this project was the Knowledge Interchange Format(KIF) framework. Other projects are the *High Performance Knowledge Bases(HPKB)* programme [15] which aims at fostering the development of technologies that can increase the rate at which we can write knowledge bases. The *Intelligent Brokering Service for Knowledge-Component Reuse on the World-Wide Web(IBROW^3)* project investigates means for supporting comprehensive reuse. The idea behind this project is to provide a brokering service that plays the role of a mediator between customers and PSM providers to support the configuration of customised knowledge systems that solve customers' problems. A library of reusable components[70] has been constructed based on the work of Motta in parametric design [69]. The *Knowledge Reuse and Fusion/Transformation(KRAFT)* [79] aimed to enable the sharing and reuse of information contained in heterogeneous databases and knowledge bases. In the area of planning the *SPAR*[89] project draws on the range of previous work in planning activity ontologies to create a practically useful *Shared Planning and Activity Representation*.

9. Problems, tradeoffs, and solutions

Despite the fact that ontologies have been applied with success in a variety of fields there are reported problems and attempts have been made to identify tradeoffs and find potential solutions. We report on the problems first. In [73] the author discusses impediments in the use of ontologies. He points out the difficulty in library ontologies, scale-up, interfacing and raises the issue of formality in ontology development. O'Leary argues also for the difficulty in establishing a consensus: "ontologies are chosen after a political decision had been made, therefore it is impossible to choose an ontology that maximises the utility of all agents in process and the group."[73]. Research in the area of studying the experts' behaviour provides an evidence of the apparent lack of consensus. For example, many researchers argue that experts disagree about even well-established features of their domain(see, [27], [65]). Others studied the behaviour of experts([82]) and found that they often held different views about a supposedly standard terminology in their field. Furthermore, in [2] the authors state: "expert knowledge is comprised of context-dependent, personally-constructed, highly-functional but fallible abstractions". This suggests that we should routinely expect evolution of experts' views, especially in domains where there is disagreement on used terms. However, we have to point out here that in situations where there is a lack of consensus among the experts regarding the domain of interest then the principle of ontology does not apply by definition: an ontology represents consensual and commonly agreed terminology about a domain of interest. In that respect we agree with O'Leary's thesis and as a rule of thumb we can say that in domains where there is literally no consensus among the domain experts then building an ontology is pointless. This, however, should not be interpreted as a guideline to build ontologies only when experts agree: this will rarely happen, as the studies described above suggest, therefore we have seen the most successful ontology stories coming from domains where the 'majority' of experts agree on used terms. The issue here is to find the right balance between commonly agreed terminology and usability of ontology. This is actually one of the ontology design principles: minimal ontological commitment[38]. We should also mention that experience with task models in the KBS community indicates a broad degree of consensus with respect to the structure of KBS tasks, like diagnosis, parametric design, scheduling, etc. The key factor here is the effective support for KBS development rather than achieving community-wide consensus, a goal of generic ontologies.

Other problematic areas have been identified: Uschold and colleagues raise the issue of lack of translators when the representation formalisms used are not the same in the context of their experiment for ontology reuse[92]. They argued that, "the translation activity involved was an intensive one and lack of automatic support is an important disadvantage". The issue of ease of reuse was also the focal point of an empirical study performed in the context of the *HPKB* project. In [14], the authors report that ease of reuse is closely related to the type of ontology: it was found that very generic ontologies provide less support and are less useful than domain-specific ones. The latter scored a constant 60% rate of reuse in the *HPKB* study in contrast with the poor 22% rate of reuse scored by broad ontologies. However, as the authors argue, these results should not undermine their role in structuring ontologies: "Although the rate of reuse of terms from very general ontologies may be significantly lower, the real advantage of these ontologies probably comes from helping knowledge engineers organise their knowledge bases along sound ontological lines."[14].

Another important drawback is the lack of rigorous evaluation techniques for ontologies. For example, in an experiment of extending the *HPKB* upper ontology[3] the author states: "...validation remains an important issue, i.e., the *PhysSys*, *EngMath* and *topology* ontologies are capable of being validated by reference to literature in their application fields...but ontologies such as the *HPKB* upper level and *SPAR* do not capture knowledge in such well understood fields, therefore this form of validation is not possible". The issue of maintenance has also been acknowledged and studied by many. Robertson neatly summarises the points made: "the cost of producing an ontology is not just in inventing the domain-specific formal language but in maintaining it once the system is deployed, since perfect ontologies cannot be guaranteed. Over-commitment to perfecting an ontology causes failure either during development(through irreconcilable arguments over what the ontology should be) or after deployment(through inappropriate human interpretation of inference system inputs or outputs)"[81]. In the long run this cost might hinder further deployment of ontologies. However, it is not easily predictable and quantifiable since there are various angles of viewing this problem. For instance, if we accept that ontology rarely stabilises then we should expect to include in our budget along with the cost of constructing, costs for maintaining the ontology we use as well as the system which uses it. How common is ontology instability? We don't know since we have very little experience with the long-term use of large libraries of ontologies. However, this is a debatable point[64] and we find projects where ontology was deployed on the rationale that it was stable(i.e., in parametric design ontology: [69]), and projects where this is not taken for granted as ontology is expected to change over time(i.e., Aitken's HPKB experiment[3]).

We now shift our attention to potential solutions to some of the problems mentioned above. With respect to the problem of library ontologies, made by O'Leary, the online libraries of ontologies(i.e., *Ontolingua*) are a potential solution especially when the maintenance and update facilities that are envisaged[25] will be fully integrated. To facilitate the familiarisation task, systems like *OntoSaurus* and *WebOnto*[18] aim to help the engineer accomplishing this task. The issue of interfacing has attracted a lot of attention by the community. It is seen from different angles: 'integration', 'merging', 'mapping', are some of the terms used. A summary of these approaches is given in [76] whereas Visser and colleagues analyse the nature of the problem in [104]. Some of the solutions proposed and applied are the *ontological mediation algorithms*[11], the *ontology clustering*[105], as well as the approaches used in projects like the creation of the *AIRCRAFT* ontology and in [94]. In addition to these, the *OBSERVER* [63] and *ONIONS*[31] systems, the *Partially Shared Views(PSV)* scheme[56], the *encapsulation and composition* technique[46] in the context of the *Scalable Knowledge Composition(SKC)*[f] project provide alternative solutions.

Even with this plethora of techniques the situation remains unsettled. There is no comparative analysis which identifies potential advantages and important drawbacks and no common practices to be followed. This has started to change with the proposal of frameworks that characterise ontologies, like the one originally presented by Uschold in [91] which was further analysed in [95]. These frameworks can be used to share experiences, discuss tradeoffs, and disseminate knowledge regarding attempts to apply ontologies. A small example of this is the instantiation of Uschold's framework, made by Kalfoglou and Robertson in the context of the *PhysSys* ontologies set[52]. Another source of information is from comparative analyses. For example, in [29], the authors compare and analyse the state-of-the-art in ontology design. Ushcold and Jasper present a cost-benefit analysis of three commonly used approaches in knowledge sharing[96]. In a larger context, Kalfoglou and colleagues, compare various meta-knowledge types, analyse their cost-benefits, and identify pragmatic aspects in using meta-knowledge[49]. In similar fashion Menzies and colleagues analyse issues with meta-knowledge in [64] and Kalfoglou speculates on the role of formal ontologies in knowledge maintenance in [48].

We close this section by summarising the points made and speculating on the future of ontology applications. We observe a shift of interest by the community from very generic, broad ontologies to domain-related ones tailored to serve particular applications. We also saw evidence in the reported systems above that ontologies can improve systems design in such areas as knowledge sharing and reuse and contribute to enhance their reliability by consistency checking. This could have an impact by reducing production costs, shortening development times and communicating context among applications and across organisations. It also improves the quality of the resulted systems with respect to verification of their correctness against domain knowledge. However, there are serious obstacles to overcome. The most important being, the considerably high cost of constructing an ontology from scratch, the lengthy learning curve which has to be traversed in order to become familiar with an ontology before integrating it in the system, the lack of rigid maintenance

[f]Electronically accessible from the URL: http://www-db.stanford.edu/SKC/

strategies, and the dearth of metrics for assessing ontology.

10. Resources

As we stated earlier, an exhaustive review of the ontologies field is impractical and overwhelming for the reader. However, for the sake of disseminating up-to-date information on ontologies we have selected and include here pointers to publicly available online resources. These are:

- a comprehensive collection of ontology-related research in alphabetical order, maintained by Peter Clark:
  http://www.cs.utexas.edu/users/mfkb/related.html
- a similar collection maintained by Enrico Franconi:
  http://www.cs.man.ac.uk/franconi/ontology.html
- a list maintained by Adam Farquhar:
  http://ksl-web.stanford.edu/kst/ontology-sources.html
- a catalogue with classified information on ontologies prepared by Yannis Kalfoglou for a panel debate that took place in the SEKE'99 conference:
  http://www.dai.ed.ac.uk/daidb/people/homes/yannisk/seke99panelhtml.html

In addition to these periodically updated online resources there are several overviews in the literature. These are, the Uschold and Gruninger review[93], the comparative review of Fridman-Noy and Hafner in [29], the survey of ontology research in [13], an overview of ontologies and PSMs in [34], and a review of planning ontologies by Tate in [89]. There are also special issues in referred journals devoted to ontology research: with respect to their role in IT[45], their involvement in KBSs[103], and their uses[99]. In addition, we should mention the volume edited by Guarino in [41].

Acknowledgements

The research described in this paper is supported by a European Union Marie Curie Fellowship (programme: Training and Mobility of Researchers).

1. N.F. Abernethy, J.F. Wu, M. Hewett, and R.B. Altman. Sophia: A Flexible, Web-Based Knowledge Server. *IEEE Intelligent Systems*, 14(4):79–85, July 1999.
2. N.M. Agnew, K.M. Ford, and P.J. Hayes. Expertise in Context: Personally Constructed, Socially Elected, and Reality-Relevant? *International Journal of Expert Systems*, 7(1), 1993.
3. S. Aitken. Extending the HPKB-Upper-Level Ontology: Experiences and Observations. In A. Gomez-Perez and R. Benjamins, editors, *Proceedings of Workshop on Applications of Ontologies and Problem Solving Methods, ECAI'98, Brighton, England*, August 1998.
4. G. Arango. *Software Reusability/Domain Analysis Methods*, pages 17–49. Ellis Horwood, Chichester, Enland, 1994.
5. J. Aspirez, A. Gomez-Perez, A. Lozano, and S. Pinto. (onto)2agent: An ontology-based www broker to select ontologies. In *Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods, ECAI'98, Brighton, England*, pages 16–24, August 1998.

6. P. Baker, C. Goble, S. Bechhofer, N. Paton, R. Stevens, and A. Brass. An ontology for bioinformatics applications. *Bionformatics*, 15(6):510–520, 1999.
7. J. Bateman, B. Magnini, and G. Fabris. The Generalized Upper Model(GUM) knowledge-base: Organisation and use. In N.J.I. Mars, editor, *Proceedings of the 2nd International Conference on Knowledge Building and Knowledge Sharing(KB & KS'95), Twente, The Netherlands*, pages 60–72, Amsterdam, NL, 1995. IOS Press.
8. R. Benjamins and D. Fensel. The Ontological Engineering Initiative-KA2. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy*, pages 287–301. IOS Press, June 1998.
9. M. Blazquez, M. Fernadez, J.M. Garcia-Pinar, and A. Gomez-Perez. Building Ontologies at the Knowledge Level using the Ontology Design Environment. In *Proceedings of the 11th Knowledge Acquisition, Modelling and Management Workshop, KAW98, Banff, Canada*, April 1998.
10. P. Borst, H. Akkermans, and J. Top. Engineering Ontologies. *International Journal of Human-Computer Studies*, 46:365–406, 1997.
11. A.E. Campbell and S.C. Shapiro. Algorithms for Ontological Mediation. Technical Report 98-03, Department of Computer Science and Engineering, State University of New York at Buffalo, January 1998.
12. B. Chandrasekaran, J.R. Josephson, and R. Benjamins. The Ontology of Tasks and Methods. In *Proceedings of the 11th Knowledge Acquisition Modeling and Management Workshop, KAW'98, Banff, Canada*, April 1998.
13. B. Chandrasekaran, R. Josephson, and R. Benjamins. What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, 14(1):20–26, January 1999.
14. P. Cohen, V. Chaudhri, A. Pease, and R. Schrag. Does prior knowledge facilitate the development of knowledge-based systems? In *Proceedings of the Sixteenth National Conference on Artificial Intelligence, AAAI'99, Orlando, FL, USA*, pages 221–226, July 1999.
15. P. Cohen, R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Gunning, and M. Burke. The DARPA High Performance Knowledge Bases project. *AI Magazine*, 19(4):25–49, 1998.
16. L. Crow and N. Shadbolt. Acquiring and Structuring Web Content with Knowledge Level Models. In R. Studer and D. Fensel, editors, *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and Management(EKAW'99), Dagstuhl, Germany*, pages 83–101. Springer Verlag, May 1999.
17. S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In R & et.al. Meersman, editor, *Proceedings of DS-8, Semantic Issues in Multimedia Systems, Boston, MA, USA*, pages 351–369, 1999.
18. J. Domingue. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In *Proceedings of the 11th Knowledge Acquisition, Modelling and Management Workshop, KAW'98, Banff, Canada*, April 1998.
19. J. Domingue and E. Motta. Planet-Onto: From News Publishing to Integrated Knowledge Management Support. IEEE Intelligent Systems(In Press), 2000.
20. Enterprise Integration Laboratory. EIL. TOVE Project, University of Toronto, Canada. available from http://www.ie.utoronto.ca/EIL/tove/ontoTOC.html, July 1995.
21. A. Farquhar, R. Fikes, and J. Rice. The ontolingua server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46(6):707–728, June 1997.
22. D. Fensel, V.R. Benjamins, E. Motta, and B. Wielinga. UPML: A Framework for knowledge system reuse. In *Proceedings of the 16th International Joint Conference*

on Artificial Intelligence, IJCAI'99, Stockholm, Sweden, pages 16–21, August 1999.

23. M. Fernandez. Overview for Methodologies for Building Ontologies. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden*, August 1999. Available from: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.
24. M. Fernandez, A. Gomez-Perez, and N. Juristo. METHONTOLOGY: From Ontological Arts Towards Ontological Engineering. In *Proceedings of the AAAI-97 Spring Symposium Series on Ontological Engineering, Stanford, CA, USA*, pages 33–40, March 1997.
25. R. Fikes and A. Farquhar. Distributed Repositories of Highly Expressive Reusable Ontologies. *IEEE Intelligent Systems*, 14(2):73–79, March 1999.
26. F. Fillion and C. Menzel. Using ontologies to enable enterprise model integration. in Appendices of the paper: "Ontologies: principles, methods and applications", Uschold,M. and Gruninger,M., The Knowledge Engineering Review, 11(2), p.130-136, 1996.
27. A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh. Inconsistency handling in multi-perspective specifications. *IEEE Transactions on Software Engineering*, 20(8):569–578, 1994. also as a Research Report DoC 93/2.
28. M.S. Fox and M. Gruninger. On Ontologies and Enterprise Modelling. In *Proceedings of International Conference on Enterprise Integration Modelling Technology 97.* Springer-Verlag, 1997.
29. N. Fridman-Noy and C.D. Hafner. The State of the Art in Ontology Design: A Survey and Comparative Review. *AI Magazine*, 18(3):53–74, 1997.
30. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1995.
31. A. Gangemi, D. Pisanelli, and G. Steve. Ontology Integration: Experiences with Medical Terminologies. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontology in Information Systems, FOIS'98, Trento, Italy*, pages 163–178, June 1998.
32. A. Gangemi, G. Steve, and F. Giacomelli. ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration. In P. van der Vet, editor, *Proceedings of the Workshop on Ontological Engineering, ECAI'96, Budapest, Hungary*, August 1996.
33. R. Genesereth and R. Fikes. *Knowledge Interchange Format.* Computer Science Dept., Stanford University, 3.0 edition, 1992. Technical Report, Logic-92-1.
34. A. Gomez-Perez and R. Benjamins. Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden*, August 1999. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.
35. A. Gomez-Perez and MD. Royas-Amaya. Ontological Reengineering for Reuse. In *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management,(EKAW99), Dagstuhl, Germany*, pages 139–157, May 1999.
36. T. Gruber and G. Olsen. An ontology for engineering mathematics. In J. Doyle, P. Torasso, and E. Sandewall, editors, *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning, San Mateo, CA, USA.*, pages 258–269, 1994.
37. T.R. Gruber. A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.
38. T.R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43:907–928, 1995.
39. M. Gruninger and M.S. Fox. Methodology for the Design and Evaluation of Ontologies. In *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in*

Knowledge Sharing, Montreal, Quebec,Canada, August 1995.

40. N. Guarino. Formal Ontology and Information Systems. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy*, pages 3–15. IOS Press, June 1998.
41. N. Guarino, editor. *Formal Ontology In Information Systems*, Frontiers in Artificial Intelligence and Applications. IOS Press, June 1998. ISBN: 90-5199-399-4.
42. N. Guarino, M. Carrara, and P. Giaretta. Formalizing Ontological Commitments. In *In Proceedings of the 12th National Conference on Artificial Intelligence(AAAI'94), Seattle, Washington, USA*, 1994.
43. N. Guarino and P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In *Proceedings of the 2nd International Conference on Knowledge Building and Knowledge Sharing(KB&KS'95), Twente, The Netherlands*, April 1995.
44. N. Guarino, C. Masolo, and G. Vetere. OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems*, 14(3):70–80, May 1999.
45. N. Guarino and R.(eds.) Poli. The Role of Ontology in the Information Technology. *International Journal of Human-Computer Studies*, 43(5/6):623–965, 1995.
46. J. Jannink, S. Pichai, D. Verheijen, and G. Wiederhold. Encapsulation and Composition of Ontologies. In *Proceedings of the AAAI'98 Workshop on Information Integration, Madison, WI, USA*, July 1998.
47. M. Jones, J. Wheadon, D. Whitgift, M. Niezatte, M. Timmermans, R. Rodriquez, and R. Romero. An agent-based approach to spacecraft mission operations. In N.J.I. Mars, editor, *Proceedings of 2nd International Conference on Knowledge Building and Knowledge Sharing(KB&KS'95), Twente, The Netherlands*, pages 259–269. IOS Press, April 1995.
48. Y. Kalfoglou. The Role of Formal Ontologies. In *Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslauten, Germany*, pages 401–405, June 1999. Position paper presented at the Panel: Knowledge Maintenance. Also as: Research Paper No.952, Dept. of AI, University of Edinburgh.
49. Y. Kalfoglou, T. Menzies, K-D. Althoff, and E. Motta. Meta-knowledge in systems design: panacea...or undelivered promise? *The Knowledge Engineering Review(submitted)*, 2000.
50. Y. Kalfoglou and D. Robertson. A Case Study in Applying Ontologies to Augment and Reason about the Correctness of Specifications. In *Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslauten, Germany*, pages 64–71, June 1999. Also as: Research Paper No.927, Dept. of AI, University of Edinburgh.
51. Y. Kalfoglou and D. Robertson. Managing Ontological Constraints. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden*, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/, August 1999. Also as: Research Paper No.948, Dept. of AI, University of Edinburgh.
52. Y. Kalfoglou and D. Robertson. Applying Experienceware to support ontology deployment. In *Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, SEKE2000, Chicago, IL, USA*, pages 266–275, July 2000.
53. Y. Kalfoglou, D. Robertson, and A. Tate. Using Meta-Knowledge at the Application Level. *Journal of Artificial Intelligence Research, submitted*, 2000. Also as: Research Paper No.956, Dept. of AI, University of Edinburgh.
54. K. Knight and S. Luk. Building a Large Knowledge Base for Machine Translation.

In *Proceedings of the American Association of Artificial Intelligence Conference-AAAI 94, Seattle, USA*, July 1994.

55. J. Lee, M. Gruninger, Y. Jin, T. Malone, A. Tate, G Yost, and other members of the PIF working group. The PIF Process Interchange Format and framework. *Knowledge Engineering Review*, 13(1):91–120, February 1998.
56. J. Lee and T. Malone. Partially Shared Views: A Scheme for Communicating between Groups Using Different Type Hierarchies. *ACM Transactions on Information Systems*, 8(1):1–26, 1990.
57. D.B. Lenat and R.V. Guha. *Building large knowledge-based systems. Representation and inference in the Cyc project.* Addison-Wesley, Reading, Massachusetts, 1990.
58. W. Mark. Ontologies as Representation and Re-Representation of Agreement. In *Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning, KR'96, Massachusetts, USA*, 1996. Position paper presented on the panel: *Ontologies: What are they and where's the research.*
59. W. Mark, J. Dukes-Schlossberg, and R. Kerber. Ontological Commitment and Domain-Specific Architectures: Experience with Comet and Cosmos. In N.J.I.Mars, editor, *Proceedings of the 2nd International Conference on Knowledge Building and Knowledge Sharing(KB & KS'95), Twente, The Netherlands*, pages 33–45, April 1995.
60. W. Mark, J. Schlossberg, S. Tyler, and J. McGuire. Cosmos: A System for Supporting Engineering Negotiation. *Concurrent Engineering: Research and Applications*, 2:173–182, 1994.
61. W. Mark, S. Tyler, J. McGuire, and J. Schossberg. Commitment-Based Software Development. *IEEE Transactions on Software Engineering*, 18(10):870–884, October 1992.
62. L.D. McGuinness. Ontological Issues for Knowledge-Enhanced Search. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98), Trento, Italy*, pages 302–316. IOS Press, June 1998.
63. E. Mena, V. Kashyap, A. Illarramendi, and A. Sheth. Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98), Trento, Italy*, pages 269–283. IOS Press, June 1998.
64. T. Menzies, K-D. Althoff, Y. Kalfoglou, and E. Motta. Issues with Meta-Knowledge. *International Journal of Software Engineering and Knowledge Engineering(to appear)*, 10(4), August 2000.
65. T. Menzies and B. Clancey. Special Issue on Situated Cognition. *International Journal of Human-Computer Studies*, 49, 1998. Editorial.
66. G.A. Miller. WORDNET: an online lexical database. *International Journal of Lexicography*, 3(4):235–312, 1990.
67. R. Mizoguchi, J. van Welkenhuysen, and M. Ikeda. Task ontology for reuse of problem-solving knowledge. In N.J.I. Mars, editor, *Proceedings of the 2nd International Conference on Knowledge Building and Knowledge Sharing(KB & KS'95), Twente, The Netherlands*, pages 46–57, Amsterdam, NL., 1995. IOS Press.
68. T.P. Moran and J.M. Carroll. *Design Rationale: Concepts, Techniques, and Use.* Lawerence Erlbaum Associates, 1996. ISBN: 0-8058-1567-8.
69. E. Motta. *Reusable Components for Knowledge Models: Case Studies in Parametric Design Problem Solving*, volume 53 of *Frontiers in Artificial Intelligence and Applications.* IOS Press, 1999. ISBN: 1-58603-003-5.
70. E. Motta, D. Fensel, M. Gaspari, and R. Benjamins. Specifications of Knowledge Components for Reuse. In *Proceedings of the 11th International Conference on Soft-*

ware Engineering and Knowledge Engineering, SEKE'99, Kaiserslauten, Germany, pages 36–43, June 1999.
71. National Library of Medicine, Bethesda, Maryland, USA. *UMLS Knowledge Sources*, 1997.
72. R. Neches, R.E. Fikes, T. Finin, T.R. Gruber, T. Senator, and W.R. Swartout. Enabling Technology for Knowledge Sharing. *AI Magazine*, 12(3):36–56, 1991.
73. D. O'Leary. Impediments in the use of explicit ontologies for KBS development. *International Journal of Human-Computer Studies*, 46(2):327–337, 1997.
74. D. O'Leary. Knowledge Management Systems: Converting and Connecting. *IEEE Intelligent Systems*, 13(3):30–33, June 1998.
75. D. O'Leary. Using AI in Knowledge Management: Knowledge Bases and Ontologies. *IEEE Intelligent Systems*, 13(3):34–39, June 1998.
76. S. Pinto, A. Gomez-Perez, and J. Martins. Some Issues on Ontology Integration. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden*, August 1999. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.
77. S.T. Polyak. A Supply Chain Process Interoperability Demonstration using the Process Interchange Format(PIF). Research Paper No.917, Department of Artificial Intelligence, University of Edinburgh, February 1998.
78. C. Potts. Supporting Software Design: Integrating Design Methods and Design Rationale. In P.T. Moran and M.J. Carroll, editors, *Design Rationale: Concepts, Techniques, and Use*, chapter 10, pages 295–321. Lawrence Erlbaum Associates, 1996.
79. A. Preece, K. Hui, P. Gray, P. Marti, T. Bench-Capon, D. Jones, and Z. Cui. The KRAFT Architecture for Knowledge Fusion and Transformation. In *Proceedings of the 19th SGES International Conference on Knowledge-based Systems and Applied Artificial Intelligence(ES'99), Cambridge, England*. Springer Verlag, December 1999. Best Technical Paper Award.
80. A.L. Rector, W.A. Nowlan, and the GALEN Consortium. The GALEN Project. *Computer Methods and Programs in Biomedicine*, 45:75–78, 1995.
81. D. Robertson. Pitfalls of Formality in Early System Design. In *Proceedings of the 1998 ARO/NSF Monterey Workshop on Increasing the Practical Impact of Formal Methods for Computer-Aided Software Development, Carmal, California*, 1998.
82. M. Shaw. Validation in a Knowledge Acquisition System with Multiple Experts. In *proceedings of the International Conference on 5th Generation Computer Systems*, pages 1259–1266, 1988.
83. D. Skuce. Viewing Ontologies as Vocabulary: Merging and Documenting the Logical and Linguistic Views. In *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Quebec, Canada*, August 1995.
84. J. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove,CA,USA, 2000. ISBN: 0-534-94965-7.
85. R. Studer, V.R. Benjamins, and D. Fensel. Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, 25(1-2):161–197, 1998.
86. T. Summer and S. Buckingham-Shum. From Documents to Discourse: Shifting Conceptions of Scholarly Publishing. In *proceedings of the CHI'98: Human Factors in Computing Systems, Los Angeles, CA, USA*, pages 95–102. ACM Press, 1998.
87. B. Swartout, R. Patil, K. Knight, and T. Russ. Toward Distributed Use of Large-Scale Ontologies. In *Proceedings of the 10th Knowledge Acquisition, Modeling and Management Workshop(KAW'96),Banff,Canada*, November 1996.
88. W. Swartout and A. Tate. Ontologies - Guest editors' introduction. *IEEE Intelligent Systems*, 14(1):18–19, January 1999.

89. A. Tate. Roots of SPAR - Shared Planning and Activity Representation. *The Knowledge Engineering Review*, 13(1):121–128, 1998.
90. M. Uschold. Knowledge level modelling: concepts and terminology. *The Knowledge Engineering Review*, 13(1):5–29, February 1998.
91. M. Uschold. Where are the Killer Apps? In Gomez-Perez,A. and Benjamins,R., editor, *Proceedings of Workshop on Applications of Ontologies and Problem Solving Methods, ECAI'98, Brighton, England*, August 1998.
92. M. Uschold, P. Clark, M. Healy, K. Williamson, and S. Woods. An Experiment in Ontology Reuse. In *Proceedings of the 11th Knowledge Acquisition Workshop, KAW98, Banff, Canada*, April 1998.
93. M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11(2):93–136, November 1996.
94. M. Uschold, M. Healy, K. Williamson, P. Clark, and S. Woods. Ontology Reuse and Application. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98), Trento, Italy*, pages 179–192. IOS Press, June 1998.
95. M. Uschold and R. Jasper. A Framework for Understanding and Classifying Ontology Applications. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden*, August 1999. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.
96. M. Uschold, R. Jasper, and P. Clark. Three Approaches for Knowledge Sharing: A Comparative analysis. In *Proceedings of the 12th Knowledge Acquisition, Modelling and Management Workshop, KAW'99, Banff, Canada*, October 1999.
97. M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The Knowledge Engineering Review*, 13(1), February 1998. Also available as AIAI-TR-195 from AIAI, University of Edinburgh.
98. M. Uschold and King.M. Towards a methodology for building ontologies. In *Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada*, 1995.
99. M. Uschold and A.(eds.) Tate. Putting Ontologies to Use. *The Knowledge Engineering Review*, 13(1):1–128, 1998.
100. A. Valente, T. Russ, R. MacGrecor, and W. Swartout. Building and (Re)Using an Ontology for Air Campaign Planning. *IEEE Intelligent Systems*, 14(1):27–36, January 1999.
101. P. van der Vet and N. Mars. Structured system of concepts for storing, retrieving, and manipulating chemical information. *Journal of Chemical Information and Computer Science*, 33:564–568, 1993.
102. P. van der Vet and N. Mars. Bottom-Up Construction of Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 10(4):513–526, 1998.
103. G. van Heijst, A. Schreider, and B.(eds.) Wielinga. Using Explicit Ontologies in KBS Development. *International Journal of Human-Computer Studies*, 46(2/3):183–292, 1997.
104. P.R.S. Visser, D.M. Jones, T.J.M. Bench-Capon, and M.J.R. Shave. Assessing Heterogeneity by Classifying Ontology Mismatches. In N. Guarino, editor, *Proceedings of 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy*, pages 148–162. IOS Press, June 1998.
105. P.R.S. Visser and V.A.M Tamma. An Experiment with Ontology-based Agent Clustering. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm,Sweden*, August 1999. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.
106. A. Waterson and A. Preece. Verifying Ontological Commitment in Knowledge-based

Systems. *Knowledge-Based Systems*, 12:45–54, April 1999.

107. WfMC. Workflow Management Coalition: Abstract Specification. WFMC-TC 1012, WfMC, October 1996. Interoperability demonstration presented at the 1996 Business Process and Workflow Conference in Amsterdam.
108. T. Winograd. *Bringing Design to Software*. Addison-Wesley, 1996. ISBN: 0-201-85491-0.