Situation and Perspective of Knowledge Engineering

Rudi Studer[1], Stefan Decker[2], Dieter Fensel[3], and Steffen Staab[1]

[1]*Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany*

{studer,staab}@aifb.uni-karlsruhe.de

[2]*Gates Bldg, Stanford University, Stanford, CA, 94306, USA*

stefan@db.stanford.edu

[3]*Division of Mathematics & Computer Science, Vrije Universiteit Amsterdam,*

De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

dieter@cs.vu.nl

Abstract. Knowledge Engineering was in the past primarily concerned with building and developing knowledge-based systems, an objective which puts Knowledge Engineering in a niche of the world-wide research efforts - at best. This has changed dramatically: Knowledge Engineering is now a key technology in the upcoming knowledge society. Companies are recognizing knowledge as their key assets, which have to be exploited and protected in a fast changing, global and competitive economy. This situation has led to the application of Knowledge Engineering techniques in Knowledge Management. The demand for more efficient (business to) business processes requires the interconnection and interoperation of different information systems. But information access and integration is not an algorithmic task that is easy to solve: much knowledge is required to resolve the semantic differences of data residing in two information systems. Thus Knowledge Engineering has become a major technique for information integration. And, last but not least the fast growing World Wide Web generates an ever-increasing demand for more efficient knowledge exploitation and creation techniques. Here again Knowledge Engineering technologies may become the key technology for solving the problem. In this paper we discuss these recent developments and describe our view of the future.

1 Introduction

In the Eighties, Knowledge Engineering (KE) was set up as a new discipline in Artificial Intelligence with the objective of providing methods and tools for constructing knowledge-based systems in a systematic and controllable way. Research in KE resulted in major achievements with respect to the structuring of knowledge models as well as the systematic construction and reuse of such knowledge models ([76], [64]). In spite of these technical achievements, KE did not get a widespread attention in the past - maybe this situation mirrored the still rather limited success story of knowledge-based systems in general.

However, this situation has started to change dramatically in recent years. Two developments may be identified as the key elements for driving this process: First, the incredibly fast growth of the World Wide Web has established a knowledge sharing infrastructure, and, second, *knowledge* has been identified as a key production factor besides labor and capital. The development of knowledge as an additional production factor is reflected in a rather young discipline, viz: Knowledge Management. Knowledge Management is concerned with acquiring, structuring, representing, and distributing knowledge within and between organizations. KE methods and tools may support all these processes in a successful way [21].

The first development established the World Wide Web as one of the main information sources for performing business processes, as well as private tasks. However, getting overwhelmed by more or less irrelevant information, Web users ask for more semantic and yet efficient knowledge creation, integration, and exploitation techniques. Again, KE methods and tools provide first solutions and a promising starting point for developing more advanced techniques. All these aspects may be summarized in a more general way: the emergence of knowledge societies puts KE in the status of a key discipline for meeting the demands of the future.

In this paper we summarize recent developments and describe our view of the future. The description is structured according to a three-layer structure of methods and applications (cf. Figure 1):

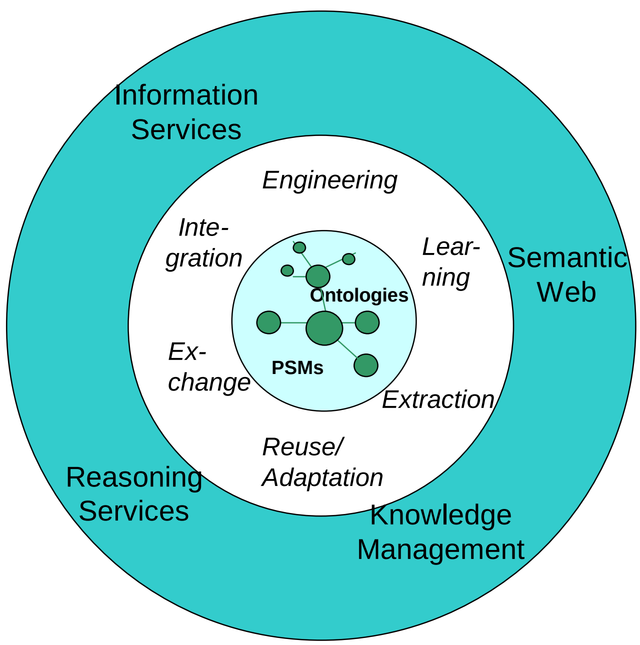

Figure 1 Basic Methods and Application Areas

- The kernel is defined by two major achievements of KE research: the notion of *ontologies* and *problem-solving methods*. These two concepts provide the backbone for building structured and reusable knowledge models. We will discuss these notions in section 2 of the paper.

- These two types of knowledge components ask for a collection of *engineering* methods for constructing complex knowledge models. Methods for aligning and integrating ontologies provide means for building-up new ontologies from existing ones. These construction processes may be enhanced by *learning techniques*, which support the extension of ontologies by proposing new concepts and relations. Problem-solving methods are building blocks for realizing the reasoning components of knowledge systems. However, *reuse* of problem-solving methods requires their *adaptation* to the current contexts, analogous to the reuse of software components in general. Furthermore, ontologies may be used to guide the *extraction* and *integration* of relevant information from various - more or less structured - sources. Since ontologies provide a common conceptual view on such sources, they can be exploited for bridging the gap between the conceptualizations of information, as envisioned by the information consumer, and the heterogeneous (low-level) conceptualizations, as offered by the various information sources. In addition, the need for the *exchange* of information and of ontologies themselves is growing, especially in multi-agent system environments. Section 3 addresses these aspects in some more detail.
- New application areas that will - according to our view - constitute the major application areas for KE concepts and methods in the future will be discussed in section 4. *Intelligent information services* and furthermore *intelligent reasoning services* will become major Web applications exploiting the notions of ontologies and problem-solving methods. In addition, transforming the World Wide Web into a *Semantic Web* is a major challenge for knowledge-based techniques. Last but clearly not least, *Knowledge Management* based on semantically enhanced intranets is a first class application area, both from a commercial and a research point of view.

As outlined, this paper will focus on rather new and promising applications of KE methods and techniques. A more 'classical' description of KE may be found in [76], a discussion of various knowledge elicitation methods and tools is given in [25].

2 Achievements

In this section we will discuss some major concepts, which were developed in the KE field in the last fifteen years. We will first outline the notion of the so-called modeling approach and then discuss two fundamental concepts in that modeling framework: Problem-Solving Methods and Ontologies.

2.1 Knowledge Engineering as a Modeling Framework

In the beginning, Knowledge Engineering was viewed as being equivalent with transferring knowledge from an expert into a knowledge base. Approaches based on this viewpoint often failed. The reasons were that experts are often unaware of experiences they use to solve their problems. Hence, crucial pieces of knowledge are not directly accessible and need to be constructed and structured during the knowledge acquisition phase. This observation has led to several modeling frameworks. In these frameworks the construction of a knowledge-based system means building a computer model with the aim of realizing problem-solving capabilities that are comparable to a domain expert. This knowledge acquisition process is therefore seen as a model construction process [16]. Some observations can be made about this modeling view of building a knowledge-based system. First, like every model, such a model is only an approximation of reality. Second, the modeling process is a *cyclic* process. New observations may lead to a refinement, modification, or completion of the already constructed model. In fact, the model may even guide the further acquisition of knowledge. Third, the modeling process depends on the subjective interpretations of the knowledge engineer. Therefore this process is typically *faulty* and an evaluation of the model with respect to reality is indispensable for the creation of an adequate model.

2.2 Problem-Solving Methods

Knowledge-based systems are computer systems that deal with complex problems by making use of knowledge. This knowledge may be acquired from humans or automatically derived with deductive, abductive, or inductive techniques. This knowledge is mainly represented declaratively rather than encoded using complex algorithms. This declarative representation of knowledge economizes the development and maintenance process of these systems and improves their understandability. Therefore, knowledge-based systems originally used simple and generic inference mechanisms, like inheritance and forward or backward resolution, to compute solutions for given problems. This approach, however, turned out to become infeasible for many real-world tasks. Indeed, it also contrasted with human experts who exploited knowledge about the dynamics of the problem-solving *process* in order to solve their problems.

[15] provided several examples where knowledge engineers encoded implicit control knowledge by ordering production rules and premises of these rules such that the generic inference engine exhibited the dynamic behavior they aimed at. Making the control knowledge explicit and regarding it as an important part of the entire knowledge contained by a knowledge-based system is the rationale that underlies *Problem-Solving Methods (PSMs)* [8]. PSMs refine the generic inference engines mentioned above to allow a more direct control of the reasoning process. Since this control knowledge is specified independently from the application domain, *reuse* of this strategically knowledge is enabled for different domains and applications. In addition, PSMs abstract from specific representation formalisms, in contrast to general inference engines that rely on a specific representation of the knowledge. Finally, PSMs decompose the reasoning task of a knowledge-based system in a number of subtasks and inference actions that are connected by knowledge roles. As such, PSMs are special types of software architectures [66]: software architectures for describing the *reasoning* part of knowledge-based systems.

Several problem solving method libraries are now available (see e.g. [13], [60], [7], [8], [31]) and a number of problem-solving method specification languages have been proposed, ranging from informal notations (e.g. CML [65]) to formal modeling languages (e.g. KARL [32], see [29] for a survey).

2.3 Ontologies

Ontologies have become a popular research topic and have been investigated by several Artificial Intelligence research communities, including KE, natural-language processing and knowledge representation. More recently, the notion of ontology is also becoming widespread in fields such as intelligent information and reasoning services, and knowledge management (see section 4). Ontologies meet a major demand in these fields: they establish a shared and common understanding of a domain that can be communicated across people and computers.

Many definitions of ontologies have been given in the last decade, but one that, in our opinion, best characterizes the essence of an ontology is based on the definition in [46]: *An ontology is a formal, explicit specification of a shared conceptualization.* A ‘conceptualization’ refers to an abstract model of some phenomenon in the world by identifying 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 understandable, which excludes natural language. 'Shared' reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group. Basically, the role of ontologies in the knowledge engineering process is to facilitate the construction of a domain model. An ontology provides a vocabulary of terms and relations with which a domain can be modeled.

Depending on their level of generality, different types of ontologies may be identified that fulfill different roles in the process of building a knowledge-based system [79]. Among others, we distinguish the following ontology types:

- *Domain ontologies* capture the knowledge valid for a particular type of domain (e.g. electronic, medical, mechanic, digital domain).
- *Generic or commonsense ontologies* aim at capturing general knowledge about the world and provide basic notions and concepts for things like time, space, state, event etc. [41]. As a consequence, they are valid across several domains.
- *Representational ontologies* do not commit themselves to any particular domain. Such ontologies provide representational entities without stating what should be represented. A well-known representational ontology is the *Frame Ontology* [46], which defines concepts such as frames, slots and slot constraints allowing knowledge engineers to express knowledge in an object-oriented or frame-based way.

Part of the research on ontologies is concerned with envisioning and constructing a technology, which enables the large-scale reuse of ontologies on a worldwide level. In order to enable reuse as much as possible, ontologies should be small modules with a high internal coherence and a limited amount of interaction between the modules. This requirement and others are expressed in design principles for ontologies ([47], [48]) and are addressed in the Ontolingua server [27] or the SKC project (Scalable Knowledge Composition) [53].

The notion of PSMs and ontologies have been reflected in various modeling frameworks, among others in CommonKADS [64], MIKE [4], and PROTÉGÉ ([26], [45]).

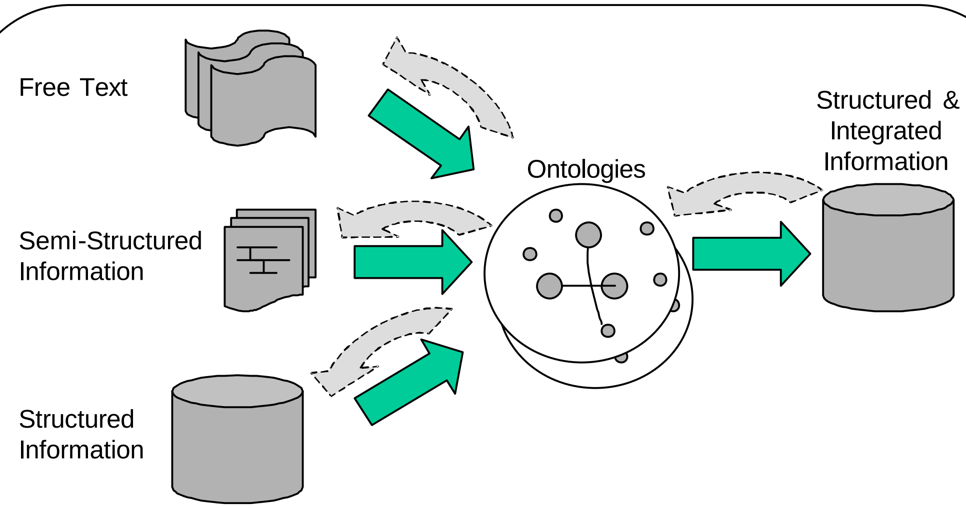

Figure 2 Information integration and extraction from, resp. information exchange between various information sources.

3 Basic Methods and Techniques

Accessing, finding or summarizing information remains a difficult task given the sheer amount of information to be found at many information sources and given their large number available through current technologies, such as the WWW. The reasons underlying this problem are manifold, however one of the major causes lies in the large gap between the conceptualizations of information, such as envisioned by the user (or one information system), and the actual storage and provision of information through an(other) information system. By and large, the question remains how to bridge this gap at all and how to bridge it in a way that minimizes the engineering task for large numbers and many varieties of information sources. Free text, semi-structured information (e.g. XML data), and database information all exhibit similar problems when it comes to providing a common conceptualization for underlying information.

As mentioned above, domain ontologies describe shared conceptualizations for particular domains of interest on a high level of technical abstraction. They describe relevant concepts, their relationships, and axioms about the relationships that enforce a well-defined semantics on the conceptualization. Regarding information integration, exchange and extraction, domain ontologies allow the precise description of a conceptualization common to varying information sources. Thus, ontologies offer themselves as (partial) solutions to the problems of integrating/exchanging/extracting information and communicating it to the user (or a database/intelligent agent) in a concise way (cf. Figure 2)**.** Sections 3.1 and 3.2 will survey methods for ontology-based information integration, extraction, and exchange. Then, we mostly skip issues of traditional ontology engineering as these have been well researched and presented in the past [10], however we glimpse into the future, when we look at how to reuse and adapt existing ontologies and problem-solving methods (Section 3.3) and how to build and maintain ontologies semi-automatically through learning techniques (Section 3.4).

3.1 Information Integration and Extraction

Integration of different data sources, *e.g*. relational databases, text files, HTML files, XML repositories, is now commonly approached through two layers (cf., e.g., [80], or the TSIMMIS approach [43]).

The first layer provides a *wrapper* for each source that encapsulates the source and provides access mechanisms that are (almost) independent of the actual structure of the source. Wrappers allow user queries or commands to be converted into source specific queries. Naturally, not all sources support the same type of queries, e.g. a wrapper may support only a few of all possible SQL commands when it encapsulates a file system. Research in wrapper construction has blossomed over the last years, many approaches for different storage types, like (rather) conventional data storage [44], HTML [63], or XML [22], have been proposed, and automatic wrapper induction has been investigated [5]. In fact, there are some approaches that already cover a larger variety of data formats, e.g. Ontobroker [20]. Since much interesting information is found in natural language texts, wrapper construction starts to include information retrieval [40] and natural language information extraction techniques [24]. At the current stage, however, developers of wrapper technology mostly concentrate on the techniques of semantic parsing [49] and mostly neglect natural language syntax (as, e.g., used in [73], [70]). Common to all these approaches are knowledge models that the data from the sources is mapped into. Though these models are often very simple and sometimes even only available in the developers' minds, they lay down the structures that may be accessed from outside the wrapper. The simplest models are nested string lists used by [63], the object exchange model of TSIMMIS

[43] already provides more and explicit structures, and many other approaches directly rely on an explicit ontology ([20], [50], [24], [73], [70], [49], [5], [40]), some recent approaches even formulate data conversion in terms of (ontological) declarative rules ([11], [17]).
While wrappers provide one common data format, the second layer consists of one or several *mediators* that integrate the heterogeneous information sources. For example, one source may contain administrative information about employees, while the second source may store information about employees' expertise. A mediator then integrates the wrapped sources in order to provide a query facility for employees with particular expertise that would be available for a project during a particular time interval. For this task the mediator must export 'fused' objects, the information about which stem partially from the first and partially from the second source.

This fusion task, however, requires the alignment of different knowledge models, or to put it crisp it requires *ontology alignment*. Alignment of schemata has been a hot topic in the database community [68] and like in ontology engineering proper (cf. [51]), linguistic cues may help considerably in determining appropriate candidates for concept alignment ([55], [18], [42]). Ontology alignment, however, needs support way beyond concept alignment. For instance, ontologies often evolve independently from each other, and an alignment may stop to work correctly if the source ontologies change. To minimize this effect, Jannink et al. [54] propose to construct minimal *ontology articulation* using an ontology algebra. While this appears to be a promising approach, further research will have to tackle many remaining issues, e.g. the integration of ontologies with axioms or the (semi-)automatic adaptation and composition of ontologies for new applications.

3.2 Ontologies for Exchange

Very similar to the problem of extracting and integrating information from various sources is the situation of exchanging knowledge (cf. arrows from right to left in Figure 2). Such situations may occur between formerly isolated legacy information systems, between intelligent agents, or when an information systems makes information available to a naive user in some intuitive form, like tables or computer-generated text. For multi-agent systems, FIPA [38] has proposed an architecture that contains ontologies. Though currently only few multi-agent systems actually employ an explicit knowledge model (e.g., [1]), it is widely accepted that in the future, ontologies will be crucial for semantically flexible and adequate communication between agents [52]. Naturally, the core problem is again the integration and alignment of different ontologies, as in a heterogeneous environment one may only rarely expect to meet another agent with a known, standard ontology. The discussions about extending the W3C RDF proposal that are currently going underway exactly reflect this general need for exchange of information and ontologies in an unrestricted environment, viz. the World Wide Web [83] (cf. also the development of OIL [36]).

3.3 Component Reuse/Adaptation

The reuse and adaptation of problem-solving methods and ontologies has been a major research topic in KE for a long time. Especially, frameworks as CommonKADS [13] and PROTÉGÉ [26] put a lot of effort in developing concepts and methods for component reuse. With easy access to existing components through the Internet this area gets more and more interest.

The UPML architecture (cf. [34]) unifies most of the existing proposals and provides a standard for describing and exchanging components of knowledge-based systems. It decomposes the entire description of a knowledge-based system into six different elements (see Figure 3): a *task* that defines the problem that should be solved by the knowledge-based system, a *problem-solving method* that defines its reasoning process, and a *domain model* that describes its domain knowledge. Each of these elements is described independently to enable the reuse of task descriptions in different domains, the reuse of problem-solving methods for different tasks and domains, and the reuse of domain knowledge for different tasks and problem-solving methods.

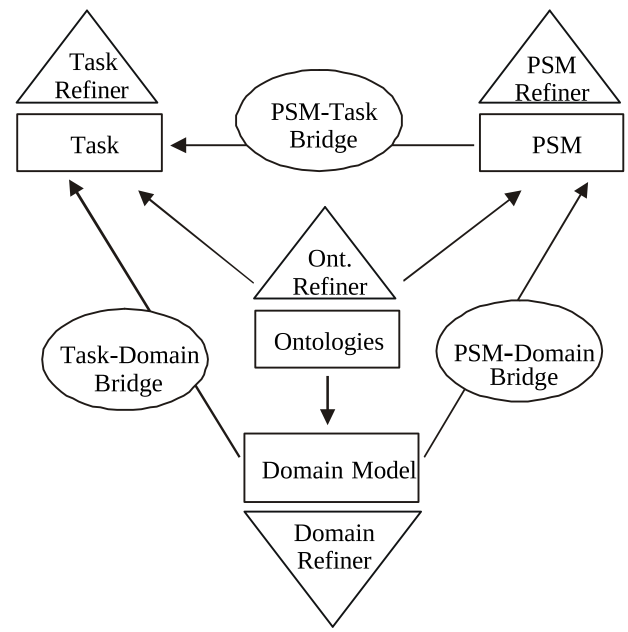

Figure 3 UPML Architecture for Knowledge-based Systems (taken from [34]).

Ontologies ([46], [59]) provide the terminology used in tasks, problem-solving methods and domain definitions. Again this separation enables knowledge sharing and reuse. For example, different tasks or problem-solving methods may share parts of the same vocabulary and definitions.

The fifth elements of a specification of a knowledge-based system are *adapters*[1] which are necessary to adjust the other (reusable) parts to each other and to the specific application problem. UPML provides two types of adapters: *bridges* and *refiners*. Bridges explicitly model the relationships between two separate parts of an architecture, e.g. between domain and task or task and problem-solving method. Refiners are used to express the stepwise adaptation of other elements of a specification, e.g. a task is refined or a problem-solving method is refined [30]. Generic problem-solving methods and tasks may be refined to more specific ones by applying a sequence of refiners to them. Again, separating generic and specific parts of a reasoning process enhances reusability. The main distinction between bridges and refiners is that bridges change the input and output of components making them fit together, whereas refiners may change internal details like subtasks of a problem-solving method. Bridges may be used to connect a generic problem-solving method like *hill-climbing* with a specific task like *diagnostic problem solving* (i.e., bridges model task-specific refinements of problem-solving methods) and refiners are used to specialize a

[1]Adapters correspond to the *transformation operators* of [78].

generic search method to a problem-solving method like *hill-climbing* (i.e., refiners specialize the algorithmic paradigm of a method).

3.4 Ontology learning

A major, expensive bottleneck for knowledge-based applications lies in the engineering itself. This situation remains in stark contrast with the "human ideal" of an intelligent agent that bootstraps its knowledge with some - though comparatively little - help from the outside world through learning. This motivated some researchers to investigate how such an agent may automatically acquire its vocabulary for communication [74]. While complete solutions to this overarching objective appears to remain in the distant future as of now, support for ontology engineering tools that works along similar principles may be found in a number of applications. For example, [14], [28] and [6] take advantage of linguistic structures that may be found in texts from the respective domains. These systems facilitate the determination of frequent and important domain vocabulary. They cluster word meanings according to correlations with other words. For instance, nouns that are found with a particular class of verbs, such as "plumbers", "researchers", "fathers" all may appear as subjects for the verb "drive", and, hence, fall into a common cluster, viz. "humans". In fact, in domain-specific tasks these systems even reach a hit rate that compares with that of humans though they tend to exhibit less accuracy [81]. Another trend is that researchers try to move on from the learning of noun meanings, to verb meanings [81] and on to implicit conceptual relations, such as ubiquitous in ontologies [57], and hence towards a more complete picture of knowledge found in domain ontologies.

4 New Application Areas

Due to changing economical conditions, new application areas have been created, which deploy the above-mentioned techniques.

4.1 Intelligent Information Services

Knowledge Engineering technology is used in a variety of information services, while leaving the paths of conventional Expert Systems.

GETESS (GErman Text Exploitation and Search System; cf. [70]) is a domain-specific information service system that aims at the fruitful combination of techniques from information retrieval, information extraction, and knowledge engineering. Its objective is the automatic search and preparation of tourism-specific information on the world wide web such that naïve users may find information about hotels, sight-seeing, and their likes more easily than with common search machines.

The core idea of GETESS is that the restriction to a particular domain makes it feasible to engineer a domain ontology that bridges between information extraction from web pages and databases, storage in a database and presentation through an intuitive interface taking advantage of the techniques presented above. Only when precise semantic retrieval fails, the system falls back onto an information retrieval strategy that accounts for unmodeled knowledge that is yet available on the gathered web pages.

Ontobroker [20] and OntoServer (c.f. http://ontoserver.aifb.uni-karlsruhe.de) integrate a repository of ontologies, an inference engine and translators to formats like RDF, XML-DTDs and Frame-Logic. OntoServer delivers built-in general-purpose deductive reasoning facilities. Ontobroker relies on the notion of a community defining a group of web users who share a common understanding and thus can agree on an ontology for a given field.

Therefore, both the information providers and the clients have complete knowledge of the available ontological terms, a prerequisite for building a community web portal [69].

SHOE [50] exploits a small extension to HTML which allows web page authors to annotate their web documents with machine-readable knowledge. In SHOE HTML pages are annotated via ontologies to support information retrieval based on semantic information. In contrast to Ontobroker and WebKB it is possible to introduce arbitrary extensions to ontologies. No central provider index is maintained, so to find all agents that confer to a given ontology, the web has to be searched completely. SHOE offers very limited inferencing capabilities in contrast to e.g. Ontobroker.

The WebKB [58] set of tools enables storing, organizing and retrieving knowledge or document elements in Web-accessible files. WebKB uses the conceptual graph formalism for representing the semantic content of Web documents by embedding conceptual graph statements into HTML tags. These statements are based on an ontology defining the concepts and relations, which may be used for annotating the HTML documents.

The RiboWeb system [2] is an online data resource for the ribosome, a vital cellular apparatus responsible for building proteins in all organisms. The system uses four principle ontologies to organize its knowledge (molecular biology, ribosome, scientific publications, types of computations). The representation of the computational capabilities it can perform allows explicit reasoning about possible actions and sequences of actions.

All the above-mentioned approaches have one thing in common: they try to integrate different knowledge from various sources and present them to a user agent. They can be seen as initial building blocks of the *Semantic Web* (see section 4.3).

4.2 Intelligent Reasoning Services

Whereas systems like Ontobroker (cf. [20], [35]) or SHOE [50] provide query access to static information sources, IBROW[2] (cf. [9], [34]), an IST project running in the 5th European Framework program, has the goal to develop a broker for the access of *dynamic reasoning services* at the WWW. It will provide customizable reasoning service in addition to information access.

The objective of IBROW is to develop intelligent brokers that are able to distributively configure reusable components into knowledge systems through the World-Wide Web. The WWW is changing the nature of software development to a distributive plug & play process, which requires a new kind of managing software: *intelligent software brokers*. On successful completion of the project, an intelligent WWW-broker is provided that can handle Web requests for reasoning services based on problem-solving methods. The broker is able to handle both the customer and the supplier side of such a request. It will access libraries on the Web, search for appropriate reasoning services, verify their requirements, ask additional information to the customer if needed, adapt the reasoning services to the particular domain knowledge, plug them together, and execute them via CORBA in a distributed way. Therefore, the user does no longer buy, download and install software. Instead he uses it as a computational service provided via the network (cf. [39]).

The overall picture of IBROW is illustrated in Figure 4. The intelligent broker will be able to handle requests for reasoners from various customers. Based on these requests it will access different libraries available on the Web and will search them for candidate reasoning services which will be adapted and configured into a knowledge system for the customer.

---

[2]http://www.swi.psy.uva.nl/projects/IBROW/home.html.

Library providers will have to make sure that their libraries comply with the description language UPML [34] and the interoperability protocol.

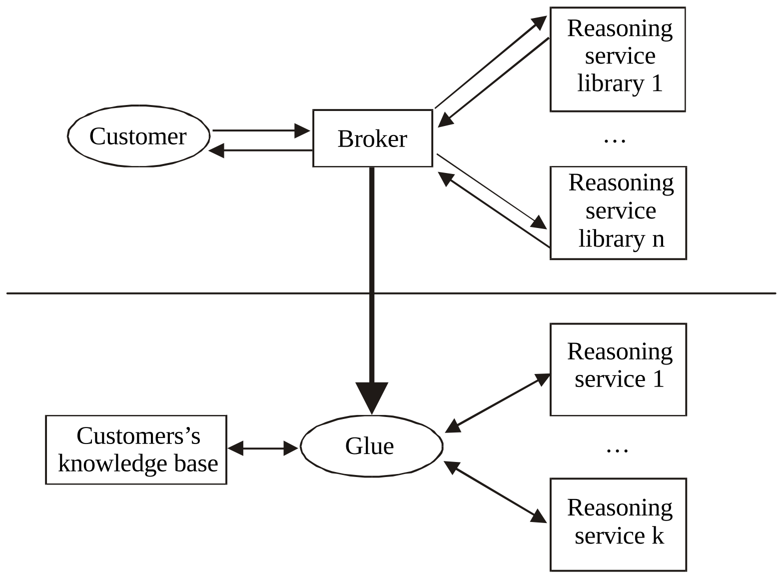

Figure 4 IBROW: Brokering Dynamic Reasoning Services in the WWW

IBROW opens the way to a new form of electronic commerce, in which the services are intelligent reasoning services. Different business models can be anticipated. In the business-to-consumer (B2C) area one can think of end users who want to solve a concrete problem such as classifications of plants, filtering of Web pages, or the selection of suitable algorithms for different kinds of data. Based on stated user requirements IBROW technology configures a suitable reasoner from generic knowledge components, and distributively executes it to provide the consumer with an answer. Depending on the popularity of the consumer request, one can decide to store the configured service for later reuse, or to throw it away. Commercial exploitation of such services would require consumers to pay, either per use or through subscription.

In a business-to-business (B2B) context, IBROW technology can be used to construct half products, which then need further processing by industries before delivering end products to consumers. For instance, a car manufacturer could be interested in a service that helps him developing and/or adapting a new car design. In another scenario the IBROW broker provides a service to configure the skeletal structure of knowledge system, which then needs to be refined for end consumers based on their particular needs. Yet another model is that IBROW technology provides an underlying infrastructure to support knowledge engineers in selecting, testing, adapting, refining and combining generic components into concrete application systems.

IBROW moves work on problem-solving methods into the direction of multi-agent systems. Here matchmaking between user request on the on side and competence descriptions of available agents on the other side as well as delegation and execution of tasks to heterogeneous agent societies are important topics in this area (see for example RETSINA ([19], [77]). Linking both areas more closely will be done in the near future.

4.3 The Semantic Web

The Internet and especially the World Wide Web are growing at a tremendous rate. More and more information is becoming directly available for human consumption. The crucial point, however, is that so far information is available for humans and it is very hard to built automated agents, which support humans in processing and filtering information. There are technologies like the World-Wide Web Wrapper Factory (W4F) [63], which support the creation of machine processable data from Web resources that have initially been created for human readers. But the generation of these wrappers is costly and they are hard to maintain: Once the structure of an HTML page has changed, the wrapper has to be changed as well. This is especially a problem if several sources are integrated into one application.

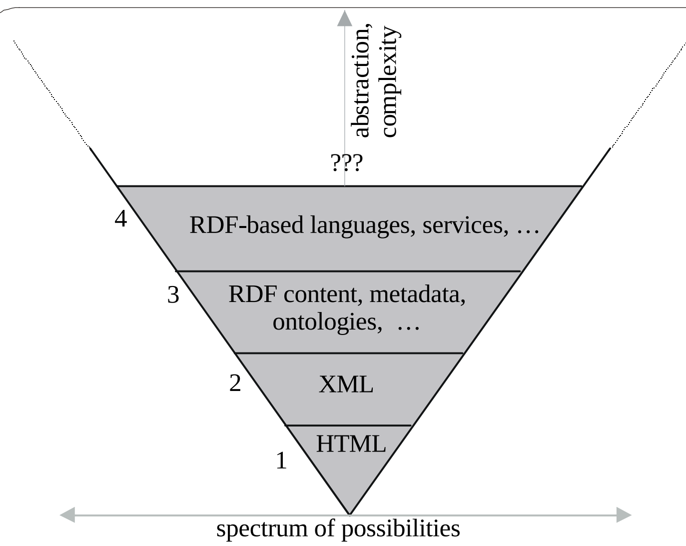

Figure 5 Development of the Semantic Web

The solution to this problem comes in form of formally defined, linked data on the Web. The W3C Working Groups have created the XML-based W3C-recommendation Resource Description Framework (RDF) [83] for expressing this kind of data. RDF is the first widely used Knowledge Representation language, and it enables the building of knowledge bases on a global scale. Together with reasoning and processing units RDF provides means for interoperability of distributed knowledge, often referred to as “The Semantic Web” (cf. [12]) by the W3C. Recent developments define ontology languages on top of RDF(S) (cf. OIL [36]) or provide generic mechanisms to define such languages as extensions of RDF(S) [71].

Figure 5 illustrates a possible roadmap to the Semantic Web. In the Semantic Web, as opposed to the Web as we know it today, the machines will be able to navigate, integrate and process information in a meaningful way. Automation of the Web promises a spectrum of possibilities the benefits of which are hard to foresee. To name just a simple one, semantic retrieval of knowledge will focus searches in a much more concise way. Application will thrive through two key technologies that we have elaborated on before, viz. ontologies as well as reasoning and inference services based on PSMs.

4.4 Knowledge Management

Knowledge Management (KM) receives more and more interest: Companies recognize that in the knowledge economy what organizations know is becoming more important than the traditional sources of economic power - capital, land, plants, and labor. It is important to recognize that KM is a multidisciplinary application area and that single solutions from one discipline do usually not work in a complex environment. Disciplines involved are e.g. management sciences, sociology, document management, ergonomics, computer supported cooperative work (cf. [23], [82]) and Knowledge Engineering. Exploiting and protecting intellectual assets (cf. [61]) is related to the objectives of Knowledge Engineering, which aims at knowledge modeling, persistent storage and retrieval of knowledge.

The "old" task of KE was the „engineering" of knowledge with the construction of a KBS in mind. This is usually not the case in Knowledge Management, as the outcome of a Knowledge Management Strategy may not be a KBS, not even a computer-based system at all. Even changes in the culture of an organization may support Knowledge Management. However, from an IT-point of view an Organizational Memory Information System (OMIS) (cf.[75], [3], [21], [56], [72]) or Knowledge Portals [69] play an important role in KM.

An example OMIS technology is given in On-To-Knowledge[3], an IST-project running in the 5th European Framework program [37]. The goal of On-To-Knowledge is to support efficient and effective knowledge management. It focuses on acquiring, maintaining, and accessing weakly-structured on-line information sources:

- *Acquiring*: Text mining and extraction techniques are applied to extract semantic information from textual information (i.e., to acquire information).
- *Maintaining*: RDF and XML are used for describing syntax and semantics of semi-structured information sources. Tool support enables automatic maintenance and view definitions on this knowledge.
- *Accessing*: Push services and agent technology support users in accessing this knowledge.

For achieving these objectives, a three-layered architecture for information access will be developed and implemented (see Figure 6). At the lowest level (the *information layer*), weakly structured information sources are processed to extract machine-processable meta-information. The intermediate level (the *representation layer*) uses this meta-information to provide automatic access, creation, and maintenance of these information sources. The highest level (the *access layer*) uses agent-based techniques as well as state-of the art querying and visualization techniques that fully employ formal annotations to guide user access of information.

At all levels, *ontologies* are the key asset in achieving the described functionality. Ontologies are used to annotate unstructured information with structural and semantic information. Ontologies are used to integrate information from various sources and to formulate constraints over their content. Finally, ontologies help to improve user access to this information. Users can define their own personalized view, their user profile and their information agents in terms of an ontology. On-To-Knowledge especially focuses on working with large, distributed, and heterogeneous ontologies.

---

[3] http://www.ontoknowledge.org

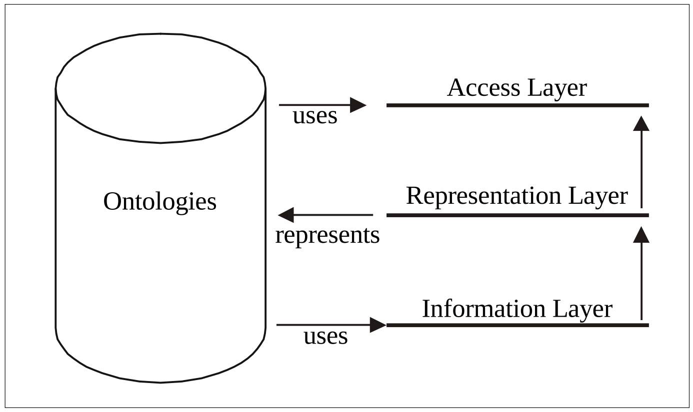

Figure 6 The Architecture of On-To-Knowledge

This tool environment is embedded into a *methodology* that provides guidelines for introducing knowledge management concepts and tools into enterprises, helping knowledge providers to present their knowledge efficiently and effectively. The methodology includes the identification of goals that should be achieved by knowledge management systems and is based on an analysis of business processes and the different roles knowledge workers play in organizations.

Current application cases of On-To-Knowledge are Organizational Memories of large organizations, help desks in call centers, and virtual enterprises.

5 Conclusion

Research and development in Knowledge Engineering has resulted in a clear understanding of the various kinds of knowledge which play an important role in realizing knowledge-based systems. Problem-solving methods and ontologies are the most notable concepts that are based on these foundations. By providing conceptual models that distinguish these types of knowledge the reuse-oriented development of knowledge-based systems was made feasible.

However, in recent years Knowledge Engineering concepts, methods and tools have gained considerable importance beyond the development of knowledge-based systems. Emerging application areas like *Semantic Web, Information and Reasoning Services* as well as *Knowledge Management* have an obvious need for such conceptual knowledge models.

Only by exploiting these knowledge models the semantic services that are sketched in Figure 7 and that have been discussed in this paper may be provided. Current technology like WWW search engines provide support for automatic information retrieval, thus helping in information source finding. The remaining tasks of extracting and using the information to solve a given task are left to the human user. Projects like Ontobroker [20], GETESS [70] and On-To-Knowledge (see section 4.4) add an additional level of service by providing automated information extraction support, thus helping the user in semantic information access and interpretation. Finally, projects like IBROW [9] also provide reasoning services that support users in task fulfillment.

Let us take a travel-planning task as an example. Current techniques provide a pill of web pages where information can be found. Intermediate services provide answers to precise questions that ask for traveling connections by specifying locations, dates, and maximal prices. Services like those offered by IBROW support in the overall planning of trips where several constraints, e.g. for the means of transport, have to be met. This will be enabled through the Semantic Web, as it is considered by the W3C.

Multi-agent systems are a further discipline that may profit from the current work in Knowledge Engineering. As outlined in the context of the IBROW-project the configuration of reasoning services has tight relationships to the matchmaking problem in multi-agent systems. Furthermore, the cooperation between agents heavily relies on a shared understanding of the task and domain at hand. Here, ontologies will play an important role in the near future.

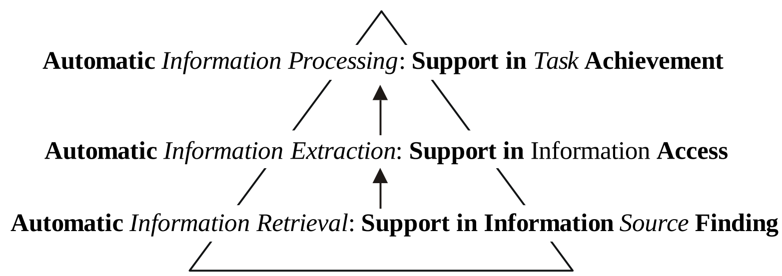

Figure 7 Three layers of services

Thus, our view on the future of Knowledge Engineering is very optimistic: future WWW applications as well as future Knowledge Management systems will have to integrate knowledge models and knowledge-based components in order to meet the needs of their users. However, the Knowledge Engineering community will have to put more effort in cooperating with other disciplines and using standards for making their methods and tools more widely known and available.

References

[1] M. Albers, C. Jonker, M. Karami, and J. Treur: An Electronic Market Place: Generic Agent Models, Ontologies and Knowledge. In: H.S. Nwana and D.T. Ndumu (eds.): *Proceedings of the Fourth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'99)*, The Practical Application Company Ltd., 1999.

[2] R.B. Altmann, M. Bada, X.J. Chai, M. Whirl Carillo, R.O. Chen, and N.F. Abernethy: RiboWeb: An Ontology-based System for Collaborative Molecular Biology. *IEEE Intelligent Systems* 14 (5), (September/October 1999), pp. 68-76.

[3] K.-D. Althoff, F. Bomarius and C. Tautz: Using Case-Based Reasoning Technology to Build Learning Software Organizations. In: *Proc. of the 1st Workshop Building, Maintaining, and Using Organizational Memories (OM-98)*, 13th European Conference on AI (ECAI ’98), Brighton, 1998. http:// SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-14/.

[4] J. Angele, D. Fensel, D. Landes, and R. Studer: Developing Knowledge-Based Systems with MIKE. *Journal of Automated Software Engineering* 5 (4), (October 1998), pp. 389-418.

[5] M. Bauer and D. Dengler: TrIAs: Trainable Information Assistants for Cooperative Problem Solving. In: *Proc. Agents’99*, pp. 260-267.

[6] B. Biebow, and S. Szulman. TERMINAE: A Linguistics-based Tool for the Building of a Domain Ontology. In: *Proceeding of the 11th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW '99)*, Dagstuhl Castle, Germany, May 26-29, 1999, Lecture Notes in Artificial Intelligence, vol. 1621, Springer, Berlin, 1999, pp. 49-66.

[7] V. R. Benjamins: Problem Solving Methods for Diagnosis and their Role in Knowledge Acquisition. *International Journal of Expert Systems: Research and Application* 8 (2), 1995, pp. 93-120.

[8] V. R. Benjamins and D. Fensel: Special Issue on Problem-Solving Methods. *International Journal of Human-Computer Studies (IJHCS)* 49 (4), 1998, pp. 305-313.

[9] V. R. Benjamins, E. Plaza, E. Motta, D. Fensel, R. Studer, B. Wielinga, G. Schreiber, Z. Zdrahal, and S. Decker: An Intelligent Brokering Service for Knowledge-Component Reuse on the World-WideWeb. In *Proceedings of the 11th Workshop on Knowledge Acquisition, Modeling, and Management (KAW´98)*, Banff, Canada, April 1998.

[10] V. R. Benjamins (ed.): *IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends*, Stockholm, August, 1999.

[11] C. Beeri and T. Milo: Schemas for Integration and Translation of Structured and Semi-structured Data. *International Conference on Database Theory 1999*, pp. 296-313.

[12] T. Berners-Lee: *Weaving the Web.* San Francisco, CA. Harper, 1999.

[13] J. Breuker and W. Van de Velde (eds.): *The CommonKADS Library for Expertise Modeling*. IOS Press, Amsterdam, The Netherlands, 1994.

[14] P.P. Buitelaar: CoreLex: An Ontology of Systematic Polysemous Classes. In: *Proceedings of International Conference on Formal Ontology in Information Systems (FOIS'98)*, Trento, Italy, June 6-8, 1998, IOS Press.

[15] W.J. Clancey: The Epistemology of a Rule-Based Expert System - a Framework for Explanation. *Artificial Intelligence* 20 (1983), pp. 215—251.

[16] W.J. Clancey: The Knowledge Level Reinterpreted: Modeling How Systems Interact. *Machine Learning* 4 (1989), pp. 285-291.

[17] S. Cluet, C. Delobel, J. Simeon, K. Smaga: Your Mediators Need Data Conversion!. In: *Proceedings ACM SIGMOD International Conference on Management of Data,* June 2-4, 1998, Seattle, Washington, USA, pp. 177-188.

[18] W. Cohen: Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity. In: *Proceedings ACM SIGMOD International Conference on Management of Data*, June 2-4, 1998, Seattle, Washington, USA, pp.201-212.

[19] K. Decker, K. Sycara, and M. Williamson: Middle-Agents for the Internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, August 1997.

[20] S. Decker, M. Erdmann, D. Fensel, and R.Studer: Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In: R. Meersman et al. (eds.), *Database Semantics: Semantic Issues in Multimedia Systems, Proceedings TC2/WG 2.6 8th Working Conference on Database Semantics (DS-8)*, Rotorua, New Zealand, Kluwer Academic Publishers, Boston, 1999.

[21] S. Decker and F. Maurer: Special Issue on Organizational Memory and Knowledge Management. *International Journal of Human-Computer Studies (IJHCS)* 51 (3), September 1999.

[22] A. Deutsch, M. Fernandez, D. Florescu, A. Levy and D. Suciu. A Query Language for XML. In: *Proc. of the 8th International World Wide Web Conference,* Toronto, May 11-14, 1999, Elsevier.

[23] R. Dieng, O. Corby, A. Giboin and M. Ribière: Methods and Tools for Corporate Knowledge Management. In: *Proc. of the 11th Knowledge Acquisition, Modeling and Management for Knowledge-based Systems Workshop (KAW'98)*, Banff, 1998.

[24] D. Embley, Y. Jiang, and Y.-K. Ng: Record-Boundary Discovery in Web Documents. *SIGMOD Record* 28 (2), pp. 467-477.

[25] H. Eriksson: A Survey of Knowledge Acquisition Techniques and Tools and their Relatioship to Software Engineering. *Journal of Systems and Software* 9 (1992), pp. 97-107.

[26] H. Eriksson, Y. Shahar, S.W. Tu, A.R. Puerta, and M.A. Musen: Task Modeling with Reusable Problem- Solving Methods. *Artificial Intelligence* 79 (1995), pp. 293-326.

[27] A. Farquhar, R. Fikes, and J. Rice: The Ontolingua Server: A Tool for Collaborative Ontology Construction. *International Journal of Human-Computer Studies* 46 (1997), pp. 707-728.

[28] D. Faure and C. Nedellec: Knowledge Acquisition of Predicate Argument Structures from Technical Texts using Machine Learning: the System ASIUM. In: *Proceeding of the 11th European Workshop on*

Knowledge Acquisition, Modeling, and Management (EKAW '99), Dagstuhl Castle, Germany, May 26-29, 1999, Lecture Notes in Artificial Intelligence, vol. 1621, Springer, Berlin, 1999, pp. 325-330.

[29] D. Fensel: Formal Specification Languages in Knowledge and Software Engineering. *The Knowledge Engineering Review* 10 (4), 1995.

[30] D. Fensel: The Tower-of-Adapter Method for Developing and Reusing Problem-Solving Methods. In E. Plaza et al. (eds.), *Knowledge Acquisition, Modeling and Management*, Lecture Notes in Artificial Intelligence (LNAI) 1319, Springer-Verlag, 1997.

[31] D. Fensel: *Problem-Solving Methods: Understanding, Development, Description, and Reuse*. Lecture Notes in Artificial Intelligence 1791, Springer-Verlag, Berlin, 2000.

[32] D. Fensel, J. Angele, and R. Studer, The Knowledge Acquisition and Representation Language KARL. *IEEE Transactions on Knowledge and Data Engineering* 10 (4), 1998, pp. 527-550.

[33] D. Fensel and V. R. Benjamins: Key Issues for Automated Problem-Solving Methods Reuse. In: *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, Brighton, UK, August 1998, pp. 63-67.

[34] D. Fensel, V. R. Benjamins, E. Motta, and B. Wielinga: UPML: A Framework for Knowledge System Reuse. In: *Proceedings of the International Joint Conference on AI (IJCAI-99)*, Stockholm, Sweden, July 31 - August 5, 1999.

[35] D. Fensel, S. Decker, M. Erdmann, and R. Studer: Ontobroker: The Very High Idea. In: *Proc. of the 11th Int. Flairs Conf. (FLAIRS-98)*, Sanibel Island, May 1998, pp. 131-135.

[36] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann and M. Klein: OIL in a Nutshell. In: *Proceedings 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, Juan-les-Pins, October 2000, Lecture Notes in Artificial Intelligence, Springer-Verlag.

[37] D. Fensel, F. van Harmelen, H. Akkermans, M. Klein, J. Broekstra, C. Fluyt, J. van der Meer, H.-P. Schnurr, R. Studer, J. Davies, J. Hughes, U. Krohn, R. Engels, B. Bremdahl, F. Ygge, U. Reimer, and I. Horrocks: OnToKnowledge: Ontology-based Tools for Knowledge Management. In: *Proceedings of the eBusiness and eWork 2000 (EMMSEC 2000) Conference*, Madrid, Spain, October 2000.

[38] FIPA 98 Specification Part 12. Ontology Service. [http://www.fipa.org/spec/FIPA98.html]

[39] G. Flammia and M. McCandless: From Software to Service: the Transformation of Srink-wrapped Software on the Internet. *IEEE Expert* 12 (2), 1997.

[40] J. Fowler, B. Perry, M. H. Nodine, B. Bargmeyer: Agent-Based Semantic Interoperability in InfoSleuth. *SIGMOD Record* 28 (1), 1999, pp. 60-67.

[41] N. Fridman-Noy and C.D. Hafner: The State of the Art in Ontology Design. *AI Magazine* 18 (3), 1997, pp. 53-74.

[42] N. Friedman Noy and M.A. Musen: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: *Proc. 17th National Conf. on AI (AAAI'2000)*, Austin, July 2000, pp. 450-455.

[43] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J.D. Ullman, and J. Widom: The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems* 8 (2), March 1997, pp. 117-132.

[44] M. R. Genesereth, A. M. Keller, O. M. Duschka. Infomaster: An Information Integration System. In: *Proc. ACM SIGMOD Conference*, Tucson, 1997, pp. 539-542.

[45] W.E. Grosso, H. Eriksson, R.W. Fergerson, J.H. Gennari, S.W. Tu and M.A. Musen: Knowledge Modelling at the Millenium: The Design and Evolution of Protégé-2000. In: *Proc. of the 12th Workshop on Knowledge Acquisition, Modeling and Management*, Banff, Alberta, October 1999.

[46] T.R. Gruber: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5 (2), 1993, pp. 199-221.

[47] N. Guarino: Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal of Human-Computer Studies* 43 (2/3), 1995, pp. 625-640.

[48] N. Guarino and Ch. Welty: Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis. In: *Proc. 14th European Conf. on AI (ECAI'2000)*, Berlin, August 2000, IOS Press.

[49] U. Hahn: Making Understanders out of Parsers: Semantically Driven Parsing as a Key Concept for Realistic Text Understanding Applications. *International Journal of Intelligent Systems*, 4 (3), 1989, pp. 345-393.

[50] J. Heflin, J. Hendler, and S. Luke: Reading between the Lines: Using SHOE to Discover Implicit Knowledge from the Web. In *Proc. AAAI-98 Workshop on AI and Information Integration*, Madison, WI, July 1998.

[51] E. Hovy: Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. In: *Proc. of the 1st Int. Conf. on Language Resources and Evaluation*, Granada, Spain, May 1998.

[52] M. N. Huhns and L. M. Stephens: Agents on the Web: Personal Ontologies. *IEEE Internet Computing* 3 (5), 1999, pp. 85-87.

[53] J. Jannink, P. Mitra, E. Neuhold, S. Pichai, R. Studer, and G. Wiederhold: An Algebra for Semantic Interoperation of Semistructured Data. In: *Proc. IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, Chicago, Nov. 1999

[54] J. Jannink, P. Srinivasan, D. Verheijen, and G. Wiederhold. Encapsulation and Composition of Ontologies. In: *Proc. AAAI Workshop on Information Integration*, Madison, WI, July 1998.

[55] P. Johannesson: Linguistic Support for Analysing and Comparing Conceptual Schemas. *Data and Knowledge Engineering* 21 (2), 1997, pp. 165-182.

[56] O. Kühn and A. Abecker, Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges. *J. of Universal Computer Science* 3, 8 (August 1977), Special Issue on Information Technology for Knowledge Management, Springer Science Online. URL: http://www.iicm.edu/jucs_3_8/corporate_memories_for_knowledge.

[57] A. Mädche and S. Staab: Learning Conceptual Relations from Texts. In: *Proc. 14th European Conf. on AI (ECAI'2000)*, Berlin, August 2000, IOS Press.

[58] P. Martin and P. Eklund: Embedding Knowledge in Web Documents. In: *Proceedings of the 8th Int. World Wide Web Conf. (WWW'8)*, Toronto, May 1999, Elsevier Science B.V., 1999.

[59] R. Mizoguchi, J. Vanwelkenhuysen, and M. Ikeda: Task Ontologies for Reuse of Problem-Solving Knowledge. In: N. J. I. Mars (ed.), *Towards Very Large Knowledge Bases*, IOS Press, 1995.

[60] E. Motta: *Reusable Components for Knowledge Modeling*. IOS Press, Amsterdam, The Netherlands, 1999.

[61] I. Nonaka and H. Takeuchi, *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, 1995

[62] S. D. Richardson, W. B. Dolan, L. Vanderwende: MindNet: Acquiring and Structuring Semantic Information from Text. In: *Proc. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 1998, Vol. 2, pp. 1098-1102.

[63] A. Sahuguet and F. Azavant: Building Light-weight Wrappers for Legacy Web Data-sources Using W4F. In: *Proc. International Conference on Very Large Databases (VLDB)*, 1999, pp. 738-741.

[64] A. Th. Schreiber, H. Akkermans, A. Anjewerden, R. de Hoog, N. Shadbolt, W. van de Velde, B. Wielinga: *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press, 1999.

[65] A. TH. Schreiber, B. Wielinga, J. M. Akkermans, W. Van De Velde, and R. de Hoog: CommonKADS. A Comprehensive Methodology for KBS Development. *IEEE Expert* 9 (6), 1994, pp. 28-37.

[66] M. Shaw and D. Garlan: S*oftware Architectures. Perspectives on an Emerging Discipline.* Prentice-Hall, 1996.

[67] M. P. Singh, P. Cannata, M. N. Huhns, N. Jacobs, T. Ksiezyk, K. Ong, A. P. Sheth, C. Tomlinson, and D. Woelk: The Carnot Heterogeneous Database Project: Implemented Applications. *Distributed and Parallel Databases*, 5(2): 207-225, 1997.

[68] S. Spaccapietra and C. Parent: View Integration: A Step forward in Solving Structural Conflicts. *IEEE Trans. on Knowledge and Data Engineering* 6 (2), 1994, pp. 258-274.

[69] S. Staab, J. Angele, S. Decker, A. Hotho, A. Maedche., H-P. Schnurr, S. Studer, Y. Sure: Semantic Community Web Portals. In: *Proceedings of the 9th International World Wide Web Conference (WWW'9)*, Amsterdam, The Netherlands, May 2000, Elsevier.

[70] S. Staab, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.- P. Schnurr, R. Studer, H. Uszkoreit and B. Wrenger: GETESS - Searching the Web Exploiting German Texts. In: *Proceedings of the 3rd International Workshop on Cooperating Information Agents* (*CIA '99)*, Upsala, Sweden, 1999, LNCS 1652, Berlin, Heidelberg: Springer.

[71] S. Staab, M. Erdmann, A. Maedche and S.Decker: An Extensible Approach for Modeling Ontologies in RDF(S). In: *Proc. Workshop Semantic Web: Models, Architectures and Management,* at the Fourth European Conference on Digital Libraries, Lisbon, Portugal, September 2000.

[72] S. Staab and H.-P. Schnurr: Smart Task Support through Proactive Access to Organizational Memory. *Journal of Knowledge-based Systems*, Elsevier, September 2000.

[73] S. Staab. *Grading Knowledge: Extracting Degree Information from Texts*. Lecture Notes in Artificial Intelligence 1744, Berlin: Springer, 1999.

[74] L. Steels and F. Kaplan: Situated Grounded Word Semantics. In: *Proceedings of the International Joint Conference on AI (IJCAI-99),* Stockholm, Sweden, July 31 - August 5, 1999, pp. 862-867.

[75] E.W. Stein: Organizational Memory: Review of Concepts and Recommendations for Management. *International Journal of Information Management* 15, 1995, pp. 17-32.

[76] R. Studer, V.R. Benjamins, and D. Fensel: Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering* 25 (1998), pp. 161-197.

[77] K. Sycara, J. Lu, M. Klusch, and S. Widoff: Matchmaking among Heterogeneous Agents on the Internet. In: *Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace,* Stanford, USA, 1999.

[78] W. van de Velde: A Constructive View on Knowledge Engineering. In: *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI-94)*, Budapest, Hungaria, August 1994.

[79] G. van Heijst, A. Th. Schreiber, and B. J. Wielinga: Using Explicit Ontologies in KBS Development. *International Journal of Human-Computer Studies* 46 (2/3), 1997, pp.183-292.

[80] G. Wiederhold and M. Genesereth: The Conceptual Basis for Mediation Services. *IEEE Expert* 12 (5), (Sep.-Oct. 1997), pp. 38-47.

[81] P. Wiemer-Hastings, A. Graesser, and K. Wiemer-Hastings: Inferring the Meaning of Verbs from Context. In: *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society,* Mahwah, NJ, Lawrence Erlbaum, 1998, pp. 1142-1147.

[82] K. Wiig, R. de Hoog, and R. van der Speek, Supporting Knowledge Management: A Selection of Methods and Techniques. *Expert Systems With Applications* 13 (1997), pp. 15-27.

[83] W3C. *Resource Description Framework (RDF)*. http://www.w3.org/RDF/