Ontologies and Knowledge Representation for Information Retrieval from Knowledge Databases and the Internet

Omerović, Sanida; Tomažič, Sašo; Radenković, Božidar; Milutinović, Veljko

ABSTRACT

Recording knowledge in a common framework that would make it possible to seamlessly share global knowledge remains a central challenge for researchers. This annotated survey of the literature examines ideas about concept representation that address this challenge.

General Terms: Concepts, Knowledge, Meaning, Modeling, Ontology, Semantics
Additional Key Words and Phrases: data, relations, representation

1. INTRODUCTION

The information world that we live in today presents us with a vast amount of data stored separately in books, newspapers, radio, TV, Internet, etc., all of them increasingly digitized. Moreover, there is an exponential increase in these data day to day so that the ability of an average computer-educated person to find a specific data element or subject-related useful piece of information is decreasing rapidly.

2. ONTOLOGY

The term ontology is taken from philosophy, where it means the study of being or existence ("What exists?", "What is?", "What am I?"). Questions about being exemplify and highlight the most basic problems in ontology: how to find a subject, a relationship, and an object to talk about. Within the more limited scope of the works cited in this paper, an ontology is a concept that groups together other (in some sense "like") concepts as shown in Figure 1. This grouping of concepts is brought under a common specification in order to facilitate knowledge sharing.

2.1 Ontology definition

Even within the limited scope of information sharing, the term ontology has been defined from many different view points and with different degrees of formality. Ontologies mostly include metadata such as concepts, relations, axioms, instances, etc [NAVIGLI 2003]. Ontologies can be viewed as mediators in the acquisition of knowledge from concepts. Therefore, ontologies lie between the concepts (that they subsume) on one hand and the overarching knowledge domain (within which they are embedded) on the other. Here are some ontology definitions from the viewpoint of concepts:

- A "specification of a shared conceptualization" [GRUBER 1993].
- An arrangement of concepts that represents a view of the world that can be used to structure information [CHAFFEE 2000].
- A conceptual model shared between autonomous agents in a specific domain [MOTIK 2002].

And here are some ontology definitions from the viewpoint of knowledge:

- An organized enumeration of all entities of which a knowledge-relation system is aware [HALLADAY 2004].
- A description of the most useful, or at least most well-trodden, organization of knowledge in a given domain [CHAN 2004].

Given ontology definitions from both of these concept and knowledge perspectives, the next issue is how ontologies can be organized.

2.2 Ontology organization

Modeling is used to achieve a consistent organization among and within ontologies; moreover finding (or inventing) consistent descriptive metadata for ontology modeling purposes is cited as the main obstacle to the introduction of ontology-based knowledge management applications into commercial environments [WARREN 2006]. Various approaches are suggested to address these problems.

Created from subsumed components, ontologies can unite classes, relationships, and entities that are equivalent but differently expressed. Ontologies themselves can be combined to model a certain knowledge domain [CHAN 2004] by organizing them as a set of terms and constraints in some form of ontology vocabulary. Alternatively, [MOTIK 2002] presents the organization of an ontology as a set consisting of: a relation, a sub-concept, an instance, a property, and a concept. In order to achieve interoperability between information presented in different ontologies, an application can consolidate ontologies into one, through a process of ontology mapping (Figure 1).

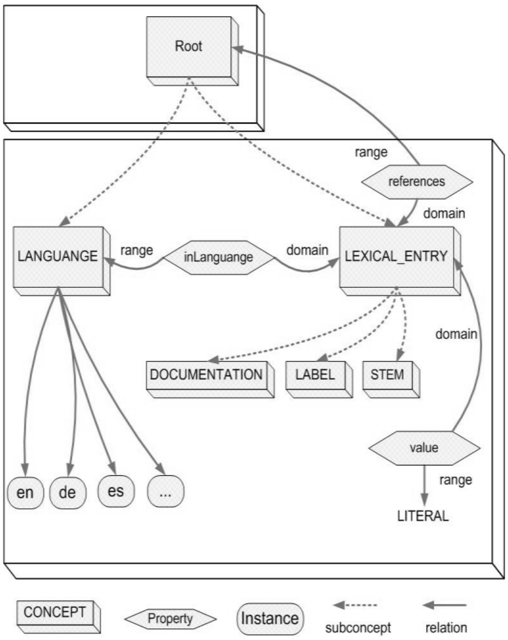

Figure 1. A Lexical Ontology-Instance-Model Structure. Each instance of a ROOT concept may have a lexical entry which reflects various lexical properties of an ontology entity, such as a label, stem, or textual documentation. Before interpreting a model, the interpreter must filter out a particular view of the model (whether a particular model can be observed as a concept, a property, or an instance) – it is not possible to consider multiple interpretations simultaneously. However, it is possible to move from one interpretation to another - if something is viewed as a concept in one case, in another case it is possible to interpret the same thing as an instance. ***This is similar to the vocabulary switching process proposed in [SCHATZ 1997].***

Ontologies can be organized as a set of hypercubes, where each hypercube represents a single concept [SCHLOOSSER 2002]. A hypercube composed of peers supporting a ***TravelService*** concept, in a peer to peer network, is presented in Figure 2.

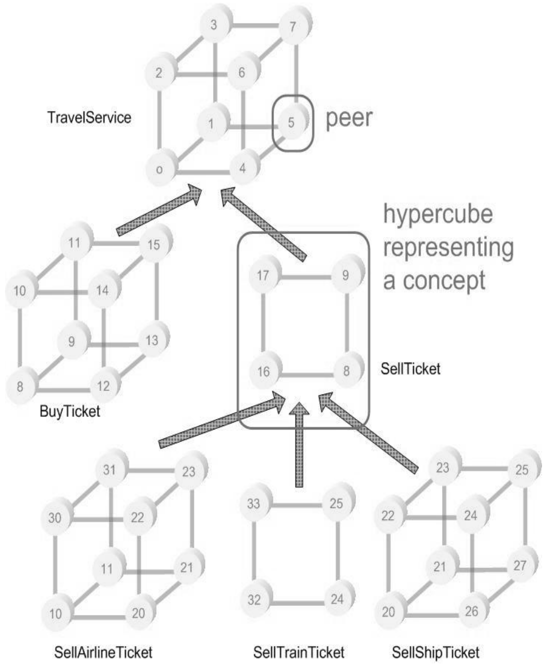

Figure 2. A peer to peer (P2P) ontology-structured network topology representing the process of buying and selling airline, train, and ship tickets. The internal peer organization of a hypercube is structured so that the network can support queries that are logical combinations of ontology and concepts . Every peer should be able to become a root of a tree spanning all nodes in the network. Also, to maintain the network symmetry, crucial for P2P networks, any node in the network should be allowed to accept and integrate new nodes in the network. Querying the network works in two routing steps. The first step is to propagate a query to those concept clusters that contain peers that the query is aiming at. In the second step, a broadcast is carried out within each one of these concept clusters, optimally forwarding the query to all peers in the clusters. This involves shortest-path routing as well as restricted broadcast in the concept coordinate system.

Some software tools for grouping and organizing ontologies are:

- Ontology library systems [DING 2001],
- Automatic ontology derivation [GAUCH 2002] from hierarchical collections of documents like Open Directory Project [OPEN 2002],
- Protégé [PROTEGE 2006],
- KAON [KAON 2001], etc.

One sees that the primary purpose of all of this effort, with an eye towards enterprise applications, is to so define and organize ontologies as to facilitate information sharing among originally incompatible data elements – possibly with the assistance of software tools that have been developed to automate this effort. This leads directly to a consideration of the ways that ontologies so constructed are used in applications to support information sharing.

```
<?xml version="1.0" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:address="http://
daml.umbc.edu/ontologies/ittalks/address#" xmlns="http://
daml.umbc.edu/ontologies/ittalks/address#" xml:base="http://
daml.umbc.edu/ontologies/ittalks/address">
- <owl:Ontology rdf:about="">
<rdfs:comment>This file describe the postal address. We are more
consider about the delivery aspect rather than accurate geographical
localtion, such as latitude and longitude.It is created by Li Ding -- http://
www.csee.umbc.edu/~dingli1/, Harry Chen -- http://www.csee.umbc.edu/
~hchen4/, Lalana Kagal -- http://www.cs.umbc.edu/~lkagal1/, Tim Finin -
- http://www.csee.umbc.edu/~finin/.</rdfs:comment>
<owl:versionInfo>$Revision: 1.1 $</owl:versionInfo>
</owl:Ontology>
- <owl:Class rdf:ID="Address">
<rdfs:label>Address</rdfs:label>
<rdfs:comment>Address</rdfs:comment>
</owl:Class>
- <owl:DatatypeProperty rdf:ID="roomNumber">
<rdfs:domain rdf:resource="#Address" />
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string" />
</owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="streetAddress">
<rdfs:domain rdf:resource="#Address" />
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string" />
</owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="city">
<rdfs:domain rdf:resource="#Address" />
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string" />
</owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="state">
<rdfs:domain rdf:resource="#Address" />
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string" />
</owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="zip">
<rdfs:domain rdf:resource="#Address" />
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string" />
</owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="country">
<rdfs:domain rdf:resource="#Address" />
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string" />
</owl:DatatypeProperty>
</rdf:RDF>
```

Figure 3. DAML ontology for the concept of *address* in OWL [OWL04]. The concept *address* is observed as a class, with the following subclasses: roomNumber, streetAddress, city, state, zip, and country.

2.3 Ontology use

Ontologies cover a broad range of knowledge. They are variously presented in this section as applied to information systems, software agents, automatic translation process, photo descriptors, and text mining. Many different applications use ontologies to explicitly declare the knowledge embedded in them [PEREZ 2002]. As an illustration, a DARPA Agent Markup Language DAML [DAML 2007] ontology record for the concept of ***address*** is presented in Figure 3.

2.3.1. Information systems

Ontologies can serve as a basis for an information system as is the case in Ontology-Driven Information Systems (ODIS) [GUARIN 1998]. ODIS system illustrates how four distinct general ontology types can be involved in building an information system in terms of Top-level, Domain, Task and Application ontologies. This unified hierarchical organization is presented in Figure 4.

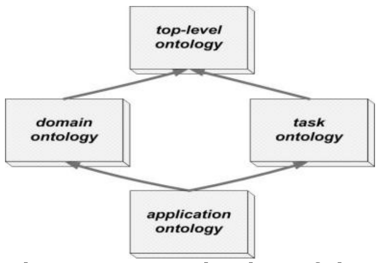

Figure 4. Organization of the ontologies in the Ontology-Driven Information System. Top-level ontologies describe very general concepts like space, time, matter, object, event, action etc., which are independent of a particular problem or domain. Domain ontologies and task ontologies describe, respectively, the vocabulary related to a generic domain (like medicine, or automobiles) or a generic task or activity (like diagnosing or selling), by specializing the terms introduced in the top-level ontology. Application ontologies describe concepts depending both on a particular domain and tasks related to a specific application.

2.3.2. Software agents

Ontologies can also be applied to agent-based computing environments. One approach in that direction is presented in [SMIRNOV 2001] where ontologies act as multi-agents in three forms:

- An application oriented ontology - a conceptual model that describes a real-world application domain depending on the particular domain and problem.
- A resource ontology – a knowledge source description in application ontology terms.
- A request ontology – a user request description in application ontology terms.

In agent-based computing environments, devices, software agents and services are expected to seamlessly integrate and cooperate with each other in support of human objectives – i.e. anticipating needs, negotiating for the service, acting on our behalf, and delivering services in an anywhere, anytime fashion. To serve as the core for such an environment, the authors [CHEN 2003] propose an intelligent Context Broker (CB). CB has the ability to integrate and reason over retrieved information in order to maintain a coherent model of the space, the devices, the agents, and the people in it. An ontology graph based on OWL [OWL 2004] supporting the work of an intelligent CB, is presented in Figure 5.

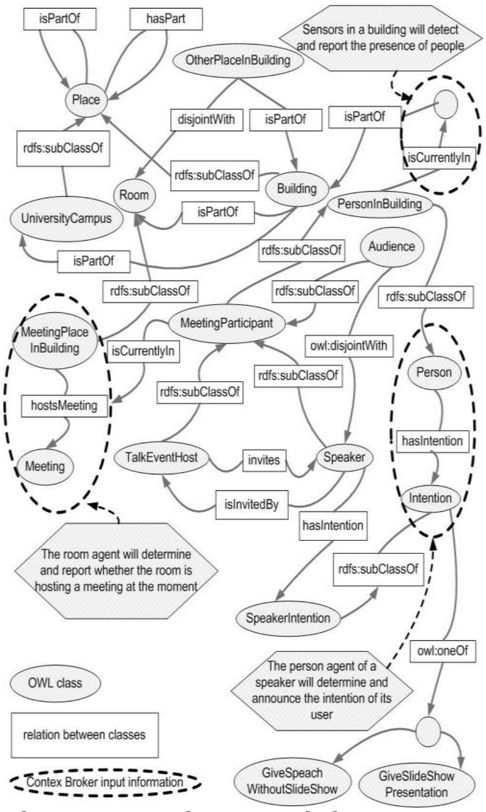

Figure 5. Ontology graph for context broker processing support. It consists of 17 classes and 32 property definitions. Each one of the classes and properties are used to describe "Person", "Place", and "Intention" from retrieved data. The "Person" class defines the most general properties about a person in an intelligent space (i.e., conference room, office room, and living room). The "Place" class defines the containment relationship properties (i.e., isPartOf, and hasPartOf) and naming properties of a place (like fullAddressName). The "Intention" class defines the notion of user intentions; for example, a speaker's intention to give a presentation and an audience's intention to receive a copy of the presentation slides and handouts. Each oval with a solid line represents an OWL [OWL 2004] class. Each oval with a broken line indicates the kind of information that CB will receive from other agents and sensors in the environment.

"Semantic interoperability" represents a type of communication between two software agents that work in overlapping domains, whether they use the same or different notations and vocabularies. Ontology-based agents such as given by OntoMerge and OntoEngine [DOU

2004] are offered as one possible solution to implement such "semantic interoperability". Like its name, OntoMerge is a tool for ontology merging, and OntoEngine is a tool to automate reasoning over merged ontologies. Here, the merger of two related ontologies is obtained by taking the union of the terms and the axioms defining them. Reasoning is automated by means of an inference mechanism that uses a dataset of several ontologies as input and automatically projects the conclusions into one or several target ontologies, as output. Another example of automated reasoning mechanism on top of ontologies is Ontology Inference Layer. It uses the Fast Classification of Terminologies [HORROC 2003] as a system to provide reasoning support for ontology design, integration, and verification [FENSEL 2001].

2.3.3. Natural language automatic translation

OntoLearn [NAVIGLO 2003], a system that automatically associates multiword English and Italian terms, is a practical example of the use of ontologies in automatic translation. This automated learning system extracts relevant domain terms from a corpus of text, relates them to appropriate concepts in a general-purpose ontology, and detects taxonomic and other semantic relations among concepts. The main features of semantic interpretation in OntoLearn are:

- A determination of the right concept (finding the sense) behind each component of a complex term (semantic disambiguation).
- An identification of the semantic relations holding among concepts to build a complex concept.

An example of semantic net in OntoLearn is presented in Figure 6.

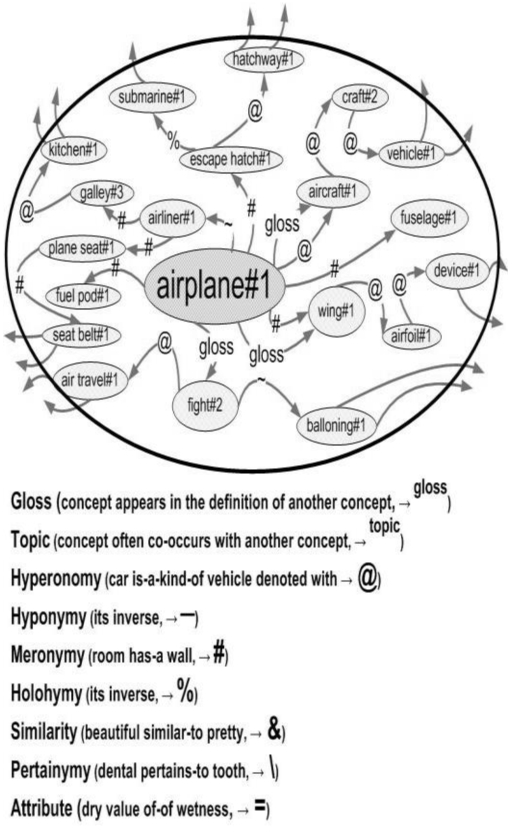

Figure 6. OntoLearn semantic net for the concept *airplane* (sense number 1, airplane#1). The system automatically builds semantic nets by using the following lexicosemantic relations: Hyperonomy, Hyponymy, Meronymy, Holohymy, Pertainymy, Attribute, Similarity, Gloss, and Topic.

2.3.4. Photo descriptors

Ontologies can also be used as a tool for describing photos, in order to help in the photo retrieval process. In [SCHREIBER 2001], the ontology-based photo annotation tool consists of the following two features:

- A photo annotation ontology and
- A subject matter vocabulary.

The photo annotation tool provides the description template for annotation construction and consists of the following features:

- A subject matter feature - what does the photo depict?
- A photograph feature - how, when, and why was the photo made?
- A medium feature - how is the photo stored?

The subject matter vocabulary is a domain-specific ontology for the animal domain (basically describing photo's subject matter). It consists of the following four elements:

- An agent (for example: "an ape"),
- An action (for example: "eating"),
- An object (for example: "a banana"), and
- A setting (for example: "in a forest at down").

2.3.5. Text mining

Text mining usually reefers to a process of automatically obtaining information from texts. An example of a text mining software tool based on ontologies is Artequakt [ALANI 2003]. It has implemented the Knowledge Extraction Tool (KET), which searches online documents and extracts knowledge that matches the given classification structure. Artequakt links KET with ontologies in order to achieve efficient information extraction from Web pages. An example of the Artequakt knowledge extraction process is presented in Figures 7a and 7b.

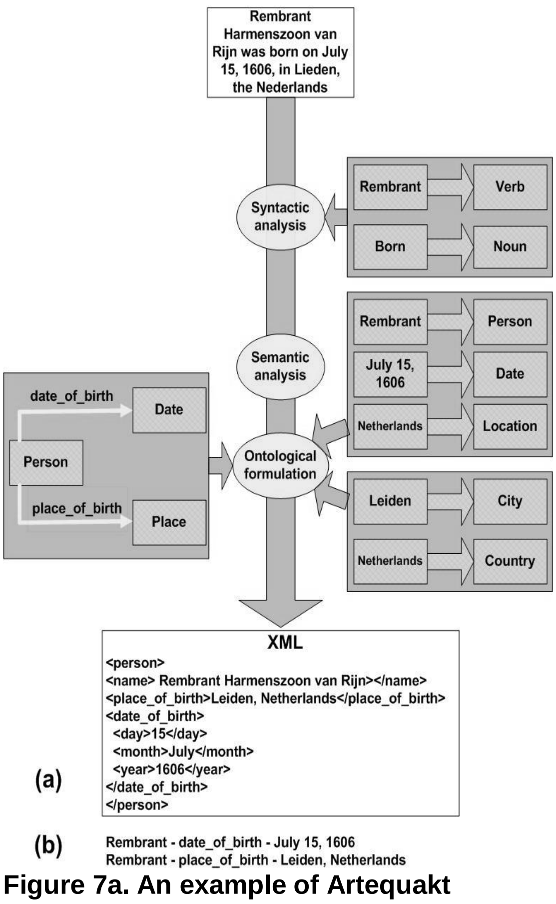

Figure 7a. An example of Artequakt knowledge extraction from a Web page.

When a Web page is recognized to match an input query, it is further processed in a form of syntactic analysis, semantic analysis, and ontological formulation. Outputs are extracted knowledge triplets from the web page in XML syntax, as shown in example (a). After the web page extracted information is presented in a form of XML, it is further processed in a form of ontology, with corresponding instances and relationships, as shown in example (b).

This section has presented various scientific contributions related to ontology definition, organization, and use. All seek to provide a common means of knowledge representation suitable for further processing. However, the thing that most of the ontologies lack (generally) is atomicity. They usually start in the middle of the problem, not at the beginning and, as a result, tend to be domain specific. They can find concepts in mammography, but not in seismological reports. As such, they can be considered to be content theories about the types of objects, properties of objects and relations between objects that are possible in a specific knowledge domain [CHANDRASEKARAN 1999]. Working up from this basis, the final section focuses on knowledge representation as bringing together both concepts and ontologies.

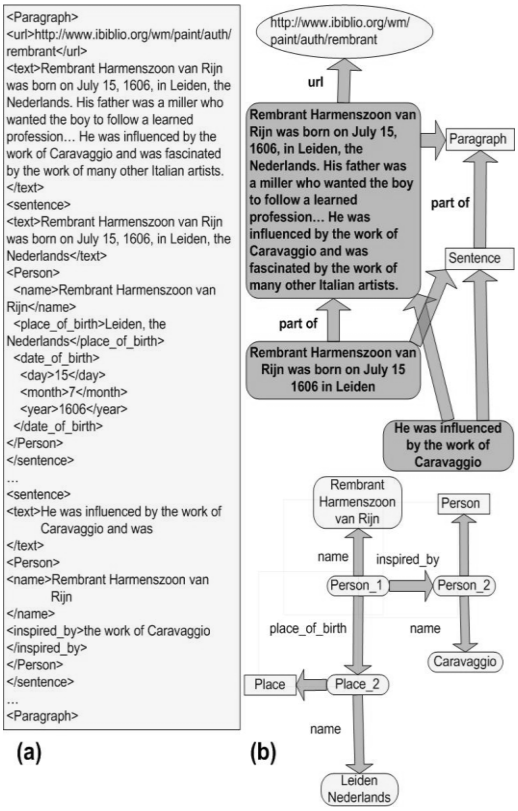

Figure 7b. Automatic Artequakt ontology population process. Based on the XML file of extracted information from the web page (a), the corresponding instances and relations are made (b), supported by Protégé [PROTEGE 2006].

3. KNOWLEDGE REPRESENTATION

A person can experience knowledge as information at its best. Loosely stated, it is information in support of or in conflict with some hypothesis or it serves to resolve a problem or to answer some specific question. This kind of knowledge that may be expected as the outcome of information processing – or it may be something new and surprising. Information as initially gathered is often fragmented and unstructured and in that form is not suitable for further exchange and processing across different systems. Moreover, a priori one does not usually have a firm grip on what the atoms of knowledge are, how they are connected, how populated, and how one can retrieve or deduce new knowledge from them. In order to answer some of these important questions, the next section begins by examining different definitions of knowledge followed by a discussion of knowledge organization and concluding with practical applications of how knowledge representation uses both concepts and ontologies.

3.1 Knowledge definition

Knowledge and *concept* are among most abstract terms in human vocabulary. Just as stated with the term of *concept*, all of the characteristics of knowledge cannot be captured within a single definition. Therefore, we start with some abstract definitions of knowledge:

- The content of all cognitive subject matter [MERRILL 2000].
- A critical resource for any activity [SMIRNOV 2001]: enterprise activity [YOON 2002], intelligent systems [GUO 2005] etc.
- A net made of entities and relationships [MILLIGAN 2003] where relationships between entities provide meaning, and entities derive their meaning from their relationships.

Some more concrete definitions of knowledge related to both concepts and ontology are:

- Conceptual models of information items or systems, including principles that can lead a decision system to resolution or action [HALLADAY 2005].
- In scale-free networks only two types of nodes exist: a few nodes with many connections, and many nodes with very few connections. Concept organization in a scale-free network can be considered as knowledge [HALLADAY 2004].

Because knowledge based on entities and relationships (upgraded in the form of concepts and ontologies) represent the foundation for many intelligent systems, this introduces the problem of how to organize the knowledge in a uniform manner to make it suitable for further processing. In order to provide answers to this important question we next discuss knowledge organization issues.

3.2 Knowledge organization

Generally speaking, knowledge organization is directly related to a particular way of thinking [YOON 2002]. There are many ways to characterize this. [MERRIL 2000] describes process of thinking consisting of knowledge objects. These knowledge objects variously describe the subject matter, the content or what is to be taught. From this perspective, four types of knowledge objects are essential for knowledge organization:

- Entities - things or objects.
- Actions - procedures that can be preformed by a learner on/to/with entities or their parts.
- Processes - events that occur often as a result of some action.
- Properties - qualitative or quantitative descriptors for entities, actions, or processes.

[HALLADAY 2004] simulates the acquisition of knowledge that has been previously organized for education purposes by introducing the concept of clusters in knowledge objects. As the subject matter of an area is learned, the relevant clusters undergo a phase transition among the connections that make up the way the cluster was originally formed.

Starting from another point of view, [PEREZ02] specifies knowledge organization using five components: concepts, relations, functions, axioms, and instances.

In [LAND01] the organization of knowledge is distinguished by the level of formality and by the level of individuality. Formality levels can be expressed as:

- Implicit knowledge, i.e., not well structured, and cannot be easily articulated, or
- Explicit knowledge, i.e., formally represented using a precise and sufficiently formal knowledge representation scheme.

(While it is possible to conceptually distinguish between explicit and implicit knowledge, in practice these are seldom separated, because new knowledge is created through the dynamic interaction and combination of both types.)

Knowledge can also be organized according to the level of individuality as:

- Individual knowledge - as resides in the brains of the individual, and is owned by the individual, or
- Collective knowledge - distributed and shared among members within same team, different teams, and organizations.

Database organization is a common form of explicit knowledge representation that facilitates both mathematical analysis and computer processing. To establish a database organization [ZELLWEGER 2003] uses navigation structures like a network of topic lists, topic data and data relationships (such as one-to-one, one-to-many, many-to-many and many-to-one). Such a database structure is presented in Figure 8a. A structure of semantic relationships database nodes is illustrated in 8b.

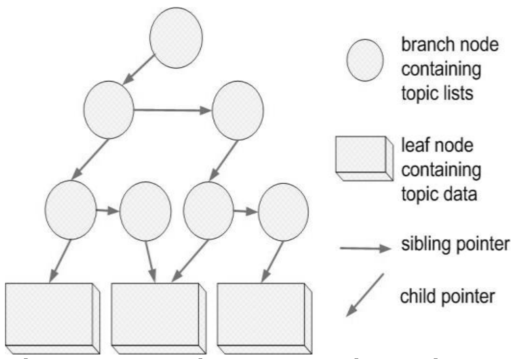

Figure 8a. Database network graph structure. Data flow starts from the root node and progresses downwards through one or more branch nodes to form paths that link to the leaf nodes. Each branch node has a sibling pointer and a child pointer. The sibling pointer creates a list of topics and the child pointer connects each list to a successor node (either another branch or leaf node). The advance is that any number of topic lists can link to the same topic data. ***It is analogous to the situation where different words can link to the same concept.***

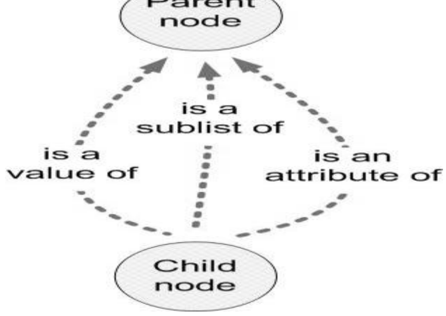

Figure 8b. Relationships between parent-child nodes within a database organization. As a knowledge structure, each parent-child node pair represents a semantic relationship like "is a sublist of", "is an attribute of", and "is a value of".

Now that several proposed ways of characterizing knowledge organization are presented, the following section focuses on various possibilities in knowledge use which combine with both concepts and ontologies use.

3.3 Knowledge use

Clearly, a main use of knowledge is in problem-solving [MERRIL 2000]. A successful knowledge structure incorporates the information required for an interested party to solve a particular problem. If the required knowledge components and their relationships are incomplete, then the party will not be able to use the available information efficiently. Problem solving by computer requires not only an appropriate knowledge representation, but also proper algorithms or heuristics to manipulate the knowledge components. A successful problem-solving sequence passes through the following phases:

- knowledge integration,
- knowledge modeling,
- knowledge storage, and
- knowledge retrieval.

3.3.1 Knowledge integration

To begin the information at hand must be cleaned up to remove redundancy, subsumption and contradiction between different knowledge entities – which is the task of knowledge integration [GUO 2005]. [MEDSKER 1995] cites the benefits of knowledge integration in one expert system:

- Existing knowledge can be reused.
- Knowledge acquired from different sources usually has a better validity and is more comprehensive than from only one source.
- Knowledge integration by computer can build a knowledge-based system faster and less expensively than can human experts.

In order to integrate different knowledge sources, the relationships between these different sources must be made explicit. In the process of integrating knowledge hidden relationships may be uncovered that reveal new knowledge. To assure that all such relations remain consistent both before and after knowledge integration one requires a knowledge modeling process – which is the next step.

3.3.2 Knowledge modeling

Knowledge modeling takes the way one thinks about data, information and knowledge from the real world (a human cognition process) and combines it with knowledge models from the information world [WEIQI 2004]. As a consequence, knowledge models incorporate the set of information entities such as data, ontology, rules, logic, and propositions. An example of such a knowledge modeling process is presented in Figure 9.

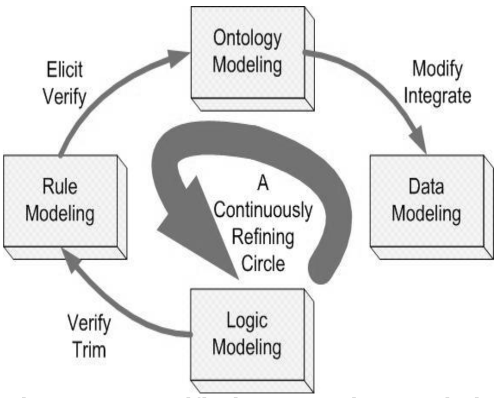

Figure 9. A Unified Knowledge Modeling process consisting of knowledge models: data, ontology, rule, and logic, forming an inner and outer circle. In the inner circle processes are carried out as follows: data can be used to build ontologies, rules can be formed on the top of these ontologies, and logic can be inferred from these rules. Each knowledge model forms the underlying base for the next model, in contrast to the outer cycle. In the outer cycle each newly built model can be useful to the previously built model: the ontology model can be used in modifying and integrating a data model, a rule model can be used in eliciting and verifying an ontology model, and a logic model can be used in verifying and trimming a rule model.

[CHAN 2004] presents another possible way to model knowledge by using Knowledge Modeling Systems (KMS) based on an Inferential Modeling Technique (IMT). IMT first models the domain objects and relations before deciding what tasks are involved and what problem-solving method to adopt. Thus KMS consists of two primary modules:

- A class module – gathers user knowledge on classes of objects, the attributes and values associated with each class, and the relationships between the classes, all related to the problem-solving domain.
- A task module – represents an organized structure or a sequence of activities that is performed to accomplish some objective in the problem-solving process.

The main benefit of building a KMS is to gain a shared and reusable knowledge base. The shared and reusable knowledge base paradigm leads us into the next section where our paper discusses how to store knowledge in such a knowledge base, once it is modeled in a uniform manner.

3.3.3 Knowledge storage

There is still no machine that can simulate the efficient way that the human brain stores its data and thinks about them, but generally a person does not even have many static records in his or her head. Over time, mankind has invented increasingly sophisticated means to store knowledge – by writing on stone, papyrus, and paper, later adding recorded speech and film. Now, all kinds of knowledge records are stored digitally in machines. With advances in computer science, knowledge stored in knowledge basis has begun to serve as the foundation for intelligent systems [GUO 2005]. But the lack of consistency in the vast amount of implicit knowledge poses a particular storage problem. [LAND 2001] has designed a specialized conceptual framework to first capture, organize and finally store implicit knowledge in the domain of software engineering. The two phases of this proposed conceptual framework are:

- Knowledge Capture (KC) and
- Knowledge Organization (KO)

as indicated in Figure 10.

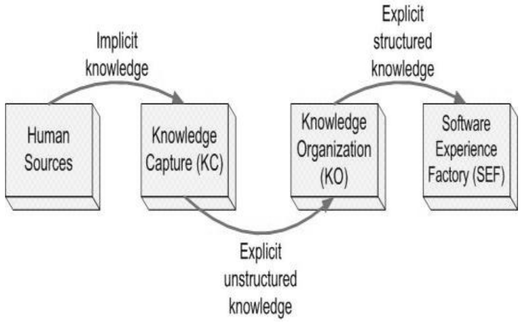

Figure 10. The process of storing implicit knowledge. Knowledge Capture (KC) extracts implicit knowledge (related to software development) residing in the minds of the parties involved, with other mechanisms such as anecdotes, case studies, lessons learned, best practices, failures, successes, etc. The knowledge retrieved with KC is explicit, but it lacks structure and organization, thus Knowledge Organization (KO) is necessary. KO usually includes transcription (translation from voice or video formats to written form), summarization (production of the main points from transcribed data), and coding (assigning symbols to transcribed data). The output of KO is an explicitly structured knowledge, suitable for further exchange and comparison in a computer system; and serves to populate the Software Experience Factory (SEF). SEF represents the storage of explicit and structured knowledge related to software development.

After knowledge from different sources has been integrated, modeled in a uniform manner, and stored in a knowledge base, the next step is the purpose of it all: the extraction of knowledge, or knowledge retrieval.

3.3.4 Knowledge retrieval

Knowledge discovery in databases or data mining refers to the nontrivial extraction of implicit, previously unknown, and potentially useful information from the data stored in Databases [FRAWLEY 1992]. Two types of queries and answers for efficient knowledge retrieval in the database domain are cited by [HAN 1996]:

- A simple data query – to find a stored data item in the database (which corresponds to a basic retrieval statement in a database system).

- A knowledge query – to find a certain rules and other kinds of complex knowledge in observed database.

The answering to a database query can take two forms:

- Direct answers that are simple examples of data or knowledge from a database.
- An answer to a query using intelligence – by first analyzing the intent of the query and then providing generalized, neighborhood, or associated information relevant to the query (by means of data summarization, concept clustering, rule discovery, etc).

One possible way to increase efficiency in a Web pages domain knowledge retrieval process is to collect user feedback from the pages visited (so that in future iterations, user searches can better refine and match the system searches). [TSAI 2003] presents a multi-agent framework that iteratively collects user's feedback and updates the user Web page profile. Its task cycle is presented in Figure 11.

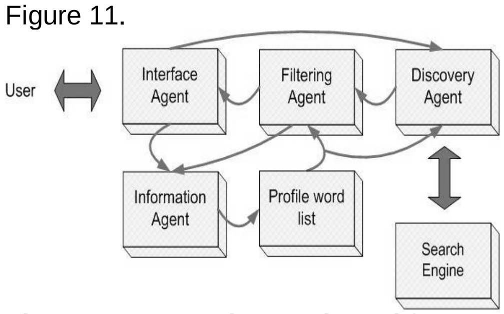

Figure 11. A Multi-agent based framework for efficient knowledge retrieval from World Wide Web. The framework consisting of agents involves the agents' task cycle as follows: An interference agent receives a user's query and redirects it to existing search engines. Then, an Information agent analyzes the Web pages chosen by the user and derives a temporary user profile. A Discovery agent, based on a user profile, performs query expansion and modification. A Filtering agent ranks the retrieved Web pages correspondingly to the user profile and recommends the most relevant web pages in future queries. The user labels useful pages which are then further processed in profile updating. The knowledge retrieval procedure continues iteratively until a user terminates the search.

Knowledge conceptualization is a special form of knowledge retrieval processing. The knowledge conceptualizing tool [FUJIHARA 1997] uses concept clustering and ranking techniques by applying a Vector Space Model [SALTON 1975] and a Probability Ranking Principle [ROBERTSON 1997]. An interview transcript, containing several question and answer pairs and consisting basically of unstructured conversational sentences, represents the system input. After processing, the outputs are a set of rules and facts extracted from the input data, thereby forming a new knowledge representation.

This section has presented a list of papers related to knowledge definition, organization, and use. Problem-solving in some sense is the final goal of every knowledge use. Therefore, most of the papers presented focused on how to get there through knowledge representation as a stepwise layered process consisting of knowledge integration, knowledge modeling, knowledge storage, and knowledge retrieval. In this way it is possible to combine different knowledge representations and merge them in order to answer a particular question or, some more general, problem-solving issue.

4. CONCLUSION

The research efforts presented here are focused on knowledge representation by ontologies populated with concepts. Concepts, ontologies, and knowledge representation are almost impossible to separate in practice, since there is no clear distinction where the use of concepts stops and use of ontologies begins in knowledge representation. Therefore, most of the research efforts presented are a combination of all tree topics. Thus the survey can be viewed as an annotated guide to this literature.

This paper sheds more light on a selected number of different avenues leading to the same future goal of knowledge retrieval based on conceptual queries, as opposed to the current state of the art based on semantic queries. As indicated in this paper, statements "I am a PhD" and "I have a doctorate," are two different semantical entities, but they both represent the same concept. Therefore, a semantic query (e.g., focused on only one of the above two statements) will be able to retrieve only a subset of relevant knowledge, while a conceptual query (focused on both statements above, as well as all other statements supporting the same concept) would retrieve the full set of relevant knowledge. A trivial solution to the problem is, for each relevant concept, to create a case structure that includes all statements supporting that particular concept. This solution is based on exhaustive approaches, and has no practical value. Practical value lies in the many sophisticated approaches discussed in this survey paper.

The authors believe that this survey will benefit both those who want to enter the field of knowledge retrieval quickly, and those who would like to extend the state-of-art. To the best of our ability, all of the relevant work up to the present has been cited and discussed. For those who are concerned with implementation, there are examples of numerous working systems. Quite clearly there is no overarching "Killer Ap"; the results achieved so far in this domain remain both tentative and incomplete. Much work remains to be done.

ACKNOWLEDGMENTS

The authors would like to thank Charles Milligan of Sun Microsystems, USA, and Gerald O'Nions of StorageTek, France, who initiated research interest in this exciting field. Also, to Dr. Tom Lincoln, University of Southern California, USA, Dr. Roger Shannon, Duke University, USA, Dr. William Robertson, Dalhousie University, Canada, Ognjen Šćekic, B.Sc., University of Belgrade, Serbia, and Djordje Popović, B.Sc., IPSI Belgrade, Serbia, for constructive feedback on this paper.

REFERENCES

[ALANI 2003] Alani, H., Kim, S., Millard, D., Weal, M., Hall, W., Lewis, P., Shadbolt, N., "Automatic Ontology-Based Knowledge Extraction from Web Documents," IEEE, Intelligent Systems, Jan-Feb 2003, pp. 14- 21 Vol. 18, Issue 1.

[CHAFFEE 2000] Chaffee, J., Gauch, S., "Personal Ontologies form Web navigation," ACM Press New York, USA, 2000

[CHAN 2004] Chan, C., "The Knowledge Modelling System and its Application," Canadian Conference on Electrical and Computer Engineering, 2-5 May 2004, pp. 1353 - 1356 Vol.3.

[CHANDRASEKARAN 1999] Chandrasekaran, B., Josephson , J., Benjamins, V., "What are Ontologies, and why do we need them?," IEEE, Intelligent Systems and Their Applications, Jan-Feb 1999, pp. 20-26 Vol. 14, Issue 1.

[CHEN 2003] Chen, H., Finin, T., "An Ontology for Context Aware Pervasive Computing Environments," Cambridge University Press, September 2003, Vol. 18, Issue 03.

[COMP 1988] "Principles of Database and Knowledge Base Systems," Computer Science Press, Rockville, Maryland, USA, 1988, Vol. 1.

[DAML 2007] DAML ontologies, DARPA, USA, www.daml.org/ontologies

[DING 2001] Ding, Y., Fensel, D., "Ontology Library Systems: The key to successful Ontology Re-use," Proceedings of the 1st Semantic Web working symposium, Stanford, CA, USA, 30 July – 1 August 2001.

[DOU 2004] Dou, D., McDermott, D., Qi, P., "Ontology Translation by Ontology Merging and Automated Reasoning," Yale University, New Haven, USA, 2004.

[FENSEL 2001] Fensel, D., Harmelen, F., Horrocks, I., McGuinness, D., Patel-Schneider, P., "OIL: An Ontology Infrastructure for the Semantic Web," IEEE, Intelligent systems, Mar-Apr 2001, pp. 38- 45, Vol.16, Issue 2.

[FRAWLEY 1992] Frawley, W., Piatetsky-Shapiro, G., Matheus, J., "Knowledge Discovery in Databases: An Overview," The American Association for Artificial Intelligence, USA, 1992.

[FUJIHARA 1997] Fujihara, H., Simmons, D., "Knowledge Conceptualization Tool," IEEE Transactions on Knowledge and Data Engineering archive, March 1997, pp. 209 – 220 Vol. 9, Issue 2.

[GAUCH 2002] Gauch, S., Madrid, J., Induri, S., Ravindran, D., Chadalavada, S., "KeyConcept: A conceptual Search Engine,"Information and Telecommunication Technology Center, Technical Report: ITTC-FY2004-TR-8646-37, University of Kansas, USA, 2002.

[GRUBER 1993] Gruber, T., "A Translation Approach to Portable Ontologies," Knowledge Acquisition, Nol. 5, No. 2, 1993, pp. 199–220.

[GUARIN98] Guarino, N., "Formal Ontology and Information Systems," Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998.

[GUO 2005] Guo, P., Fan, L., Ye, L., Cao, J., "An Algorithm for Knowledge Integration and Refinement," Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.

[HALLADAY 04] Halladay, S., Milligan, C., "The Application of Network Science Principles to Knowledge Simulation," Proceedings of the 37th Annual Hawaii International Conference on System Sciences, Hawaii, 5-8 Jan. 2004.

[HALLADAY 05] Halladay, S., Milligan, C., "Knowledge VS. Intelligence, IPSI Belgrade, Proceedings of the IPSI-2005 Montengro conference, Sveti Stefan, Montenegro, 2005.

[HAN 1996] Han, J., Huang, Y., Cercone, N., Fu, Y., "Intelligent Query Answering by Knowledge Discovery Techniques," IEEE Transactions on Knowledge and Data Engineering, June 1996, pp. 373-390 Vol. 8, No. 3.

[HORROC 2003] Horroc, I., "Fast Classification of Terminologies," School of Computer Science, University of Manchester, UK, 2003.

[KAON 2001] KAON, University of Karlsruhe, Germany, 2001, http://kaon.semanticweb.org.

[LAND 2001] Land, L., Aurum, A., Handzic, M., "Capturing Implicit Software Engineering Knowledge," IEEE Computer Society, 13th Australian Software Engineering Conference, 2001, pp. 108-114.

[MEDSKER 95] Medesker, L., Tan, E. Turban "Knowledge Acquisition from Multiple Experts: Problems and Issues," Expert Systems with Applications, 1995, pp.35-40 Vol.9, Issue 1.

[MERRIL 2000] Merrill, M., "Knowledge Objects and Mental Models," Proceedings of the International Workshop on Advanced Learning Technologies, Palmerston North, New Zealand 12 Apr. - 12. June 2000, pp. 244-246.

[MILLIGAN 2003] Milligan, C., Halladay, S., "The Realities and Facilities Related to Knowledge Representation," IPSI Belgrade, Proceedings of the IPSI-2003 Montengro conference, Sveti Stefan, Montenegro, 2003.

[MOTIK 2002] Motik, B., Maedche, A., Volz, R., "A Conceptual Modeling Approach for Semantic-driven Enterprise Applications," Springer Berlin / Heidelberg, Book on the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE: Confederated International Conferences CoopIS, DOA, and ODBASE 2002. Proceedings 2002 Vol. 2519.

[NAVIGLI 2003] Navigli, R., Velardi, P., Gangemi, A., "Ontology Learning and Its Application to Automated Terminology Translation," IEEE, Intelligent Systems, 2003, pp. 22-31.

[OPEN 2002] "The Open Directory Project," 2002, http://www.dmoz.org.

[OWL 2004] OWL, Web Ontology Working Group, 2004, http://www.w3.org/2004/OWL/

[PEREZ 2002] Gomez-Perez, A., Corcho, O., "Ontology Languages for the Semantic Web," IEEE, Intelligent Systems, Jan-Feb 2002, pp. 54-60 Vol.17, Issue 1.

[PROTEGE 2006] Protégé, Stanford University, USA, 2006, http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary.

[ROBERTSON 1994] Robertson, S., Jones, K., "Simple, Proven Approaches to Text Retrieval," Technical report TR356, Cambridge University Computer Laboratory, UK, 1994.

[ROBERTSON 1997] Robertson, S., "The Probability Ranking Principle in Information Retrieval," Morgan Kaufmann Publishers, Readings in information retrieval archive 1997 pp. 281 – 286.

[SALTON 1975] Salton, G., Wong, A., "A Vector Space Model for Automatic Indexing," Communications of the ACM, 1975, pp. 613 - 620 Vol. 18, Issue 11.

[SCHATZ 1997] Schatz, B., "Information Retrieval in Digital Libraries: Bringing Search to the Net," Science, 17 January 1997 pp. 327-334 Vol. 275.

[SCHLOSSER 2002] Schlosser, M., Sintek, M., Decker, S., Nejdl, W., "HyperCup – Hupercubes, Ontologies and Efficient Search on P2P Networks," International Workshop on Agents and Peer-to-Peer Computing, Bologna, Italy, July 2002.

[SCHREIBER 2001] Schreiber, A., Dubbeldam, B., Wielemaker, J., Wielinga, B.,"Ontology-Based Photo Annotation," IEEE, Intelligent Systems, May-Jun 2001, pp. 66- 74 Vol. 16, Issue 3.

[SMIRNOV 2001] Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., "Ontology Management in Multi-agent System for Knowledge Logistics," Proceedings of the International Conferences on Info-tech and Info-net, Beijing, China 2001, pp. 231-236 Vol.3.

[TSAI 2003] Tsai, T., Lee, W., "An Interactive Agent-based System for Concept-based Web Search," Expert Systems with Applications, May 2003, pp. 365-373 Vol. 24, Issue 4.

[WEIQI 2004] WeiQi, C., JuanZi, L., KeHong, W., "CAKE: The Intelligent Knowledge Modeling Web Services for Semantic Web," The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings 26-28 May 2004 Xiamen, China, pp. 209-216.

[W3C 2007] World Wide Web consortium, http://www.w3.org

[YOON 2002] Yoon, T., Fujisue, K., Matsushima, K., "The progressive Knowledge Reconstruction and its Value Chain Management," Engineering Management Conference, 2002. IEMC '02. 2002 IEEE International, 2002, pp. 298- 303 Vol.1.

[ZELLWEGER 2003] Zellweger, P., "A Knowledge–based Model to Database Retrieval," Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 30 Sept.- 4 Oct. 2003, pp. 747- 753.