Semantic Web - Ontological Engineering for Knowledge Services

Knowledge extraction and representation play an important part in service construction. The chapter analyzes state of the art knowledge extraction and knowledge representation methods. First the role of context in the understanding of knowledge is discussed and possible model for context extraction is described. Next ontology is presented as a knowledge representation. Then context and ontologies are described as two complementary perspectives for defining knowledge and representing it and a model that integrates them is outlined. Next the role of knowledge representation methods in matching Web services and in creating bootstrapping ontology to represent Web services is summarized. In addition, the chapter provides model-based implementations of services in the fields of e-government, medical analysis, and crisis response.

1 Context

1.1 Related work on context

Context has been researched from many aspects, including the aspects of artificial intelligence, natural languages, conversations, formalism of knowledge, goal planning, human expertise in context, knowledge representation, and expert systems.

McCarthy (1987) in his paper *Generality in Artificial Intelligence* mentioned some of the main problems existing in the field. The formalization of the notion of context was defined as one of the main problems. McCarthy argued that a most general context does not exist.

Consequently, the formalization of context and a formal theory of introducing context as formal objects were developed (McCarthy & Buvac, 1997). Context was introduced as an abstract mathematical entity with properties useful in artificial intelligence. The abstract definition of context was developed in the Cyc project in the form of microtheories (Guha, 1991).

The formal theory of context was used to resolve lexical ambiguity and reason about disambiguation (Buvac, 1996).

The blackboard model of problem solving arose from the Hearsay speech understanding systems (Erman, Hayes-Roth, Lesser, & Reddy, 1980). These ideas were then extended into the standard blackboard architecture in Hearsay-II. The blackboard model has proven to be popular for AI problems and in the years since HS-II a variety of blackboard-based systems have been developed. HS-III was developed to integrate alternative representations. HS-III had a context mechanism that allowed the integration of knowledge to resolve uncertainty.

Blackboard architectures have been used for interpretation problems such as speech understanding (Lesser, Fennell, Erman, & Reddy, 1975), signal understanding (Carver & Lesser, 1992), and image understanding (Williams, Lowrance, Hanson, & Riseman, 1977) and for planning and control (Hayes-Roth, 1985).

Blackboard architecture will be implemented in the context recognition model. The different attributes of the current "world state" are translated into text and added in turn to the blackboard. The data represented in the blackboard model serve as the input to the context recognition algorithm.

1.2 Information seeking and information retrieval

Information seeking is the process in which people turn to information resources in order to increase their level of knowledge in regard to their goals (Modica, Gal, & Jamil, 2001). Information seeking has influenced the way modern libraries operate (using instruments such as catalogs, classifications, and indexing) and has affected the World Wide Web in the form of search engines.

Although the basic concept of information seeking remains unchanged, the growing need for the automation of the process has called for innovative tools to assign some of the tasks involved in information seeking to the machine level. Thus, databases are extensively used for the efficient storage and retrieval of information. In addition, over the years techniques from the realm of Information Retrieval (Salton & McGill, 1983) were refined to predict the relevance of information to a person's needs and to identify appropriate information for a person to interact with. Finally, the use of computer-based ontologies (Smith & Poulter, 1999) was proposed to classify the available information based on some natural classification scheme that would permit more focused information seeking.

Valdes-Perez and Pereira (2000) developed an algorithm based on the concise all pairs profiling (CAPP) clustering method. This method approximates profiling of large classifications. Use of hierarchical structure was explored for classifying a large, heterogeneous collection of web content (Dumais & Chen, 2000). Another method involves checking the frequency of the possible keyphrases of articles using the Internet (Turney, 2002). However, this method is based on an existing set of keywords and uses the Internet for ranking purposes only.

There is an extensive body of literature and practice in the area of information science on ontology construction using tools such as a thesaurus (Aitchison, Gilchrist, & Bawden, 1997) and on terminology rationalization (Soergel, 1985) and matching of different ontologies (Schuyler, Hole, & Tuttle, 1993). In the area of databases and information systems many models were proposed to support the process of semantic reconciliation, including the SIMS project (Arens, Knoblock, & Shen, 1996), SCOPES (Ouksel & Naiman, 1994), dynamic classificational ontologies (Kahng & McLeod, 1996), COIN (Moulton, Madnick, & Siegel, 1998), and CoopWARE (Gal, 1999), to name a few. Ontology construction can be seen as a manual effort to define relations between concepts, while context recognition attempts to identify, in this case automatically, instances of a given situation that could be related to a concept or concepts in the ontology framework.

1.3 Context recognition

One context recognition approach addressed the creation of taxonomies from metadata (in XML/RDF) containing descriptions of learning resources (Papatheodorou, Vassiliou, & Simon, 2002). Following the application of basic text normalization techniques, an index was built, observed as a graph with learning resources as nodes connected by arcs labeled by the index words common to their metadata files. A cluster mining algorithm is applied to this graph and then the controlled vocabulary is selected statistically. However, a manual effort is necessary to organize the resulting clusters into hierarchies. When dealing with medium-sized corpora (a few hundred thousand words), the terminological network is too vast for manual analysis, and it is necessary to use data analysis tools for processing.

Therefore, Assadi (1998) employed a clustering tool that utilizes specialized data analysis functions and clustered the terms in a terminological network to reduce its complexity. These clusters are then manually processed by a domain expert to either edit them or reject them.

Several distance metrics were proposed in the literature and can be applied to measure the quality of context extraction. Prior work had presented methods based on information retrieval techniques (Rijsbergen, 1979) for extracting contextual descriptions from data and evaluating the quality of the process. Motro and Rakov (1998) proposed a standard for specifying the quality of databases based on the concepts of soundness and completeness.

The method allowed the quality of answers to arbitrary queries to be calculated from overall quality specifications of the database. Another approach (Mena, Kashyap, Illarramendi, & Sheth, 2000) is based on estimating loss of information based on navigation of ontological terms. The measures for loss of information were based on metrics such as precision and recall on extensional information. These measures are used to select results having the desired quality of information.

1.4 Web context extraction model

Several methods were proposed in the literature for extracting context from text. A class of algorithms was proposed in the IR community, based on the principle of counting the number of appearances of each word in a text, assuming that the words with the highest number of appearances serve as the context. Variations on this simple mechanism involve methods for identifying the relevance of words to a domain, using methods such as stop-lists and inverse document frequency. For illustration purposes, a description is provided of a context recognition algorithm that uses the Internet as a knowledge base to extract multiple contexts of a given situation, based on the streaming in text format of information that represents situations.

A *context descriptor* c_i from domain *DOM* is defined as an index term used to identify a record of information (Mooers, 1972). It can consist of a word, phrase, or alphanumerical term. A weight $w_i \in R$ identifies the importance of descriptor c_i in relation to the information. An example is a descriptor c_1= Address and w_1 = 42. A *descriptor set* $\{\langle c_1, w_1\rangle\}_i$ is defined by a set of pairs, descriptors and weights.

Each descriptor can define a different point of view of the concept. The descriptor set eventually defines all the different perspectives and their relevant weights, which identify the importance of each perspective.

The context is obtained by collecting all the different viewpoints delineated by the different descriptors. A *context* $C = \left\{\{\langle c_{ij}, w_{ij}\rangle\}_i\right\}_j$ is a set of finite sets of descriptors, where i represents each context descriptor and j represents the index of each set. For example, a context C may be a set of words (hence DOM is a set of all possible character combinations) defining textual information and the weights can represent the relevance of a descriptor to the information. In classic Information Retrieval, $\langle c_{ij}, w_{ij} \rangle$ may represent the fact that the word c_{ij} is repeated w_{ij} times in the textual information.

The context extraction algorithm is adapted from (Segev, Leshno, & Zviran, 2007a). The input of the algorithm is defined as tokens extracted from textual information. The sets of tokens are extracted as sentences or parsed sets of words, for example *Get Domains By Zip*, as described in Figure 1. Each set of tokens is then sent to a Web search engine and a set of descriptors is extracted by clustering the Web pages search results for each token set.

The Web pages clustering algorithm is based on the concise all pairs profiling (CAPP) clustering method (Valdes-Perez & Pereira, 2000). This method approximates profiling of large classifications. It compares all classes pairwise and then minimizes the total number of features required to guarantee that each pair of classes is contrasted by at least one feature. Then each class profile is assigned its own minimized list of features, characterized by how these features differentiate the class from the other features.

Figure 1 shows an example that presents the results for the extraction and clustering performed on tokens Get Domains By Zip. The context descriptors extracted include: $\{\langle \text{Zip Code}, (50, 2) \rangle$, $\langle \text{Download}, (35, 1) \rangle, \langle \text{Registration}, (27, 7) \rangle, \langle \text{Sale}, (15, 1) \rangle,$ $\langle \text{Security}, (10, 1) \rangle, \langle \text{Network}, (12, 1) \rangle, \langle \text{Picture}, (9, 1) \rangle, \langle \text{Free Domains}, (4, 3) \rangle$. A different point of view of the concept can been seen in the previous set of tokens Domains where the context descriptors extracted include: $\{\langle \text{Hosting}, (46, 1) \rangle, \langle \text{Domain}, (27, 7) \rangle$, $\langle \text{Address}, (9, 4) \rangle,$ $\langle \text{Sale}, (5, 1) \rangle, \langle \text{Premium}, (5, 1) \rangle, \langle \text{Whois}, (5, 1) \rangle\}$. It should be noted that each descriptor is accompanied by two initial weights. The first weight represents the number of references on the Web (i.e., the number of returned Web pages) for that descriptor in the specific query. The second weight represents the number of references to the descriptor in the textual information (i.e., for how many name token sets was the descriptor retrieved). For instance, in the above example, Registration appeared in 27 Web pages and 7 different name token sets in the text referred to it.

The algorithm then calculates the sum of the number of Web pages that identify the same descriptor and the sum of the number of references to the descriptor in the text. A high ranking in only one of the weights does not necessarily indicate the importance of the context descriptor. For example, high ranking in only Web references may mean that the descriptor is important since the descriptor widely appears on the Web, but it might not be relevant to the topic of the text (e.g., *Download* descriptor in Figure 1). To combine values of both the Web page references and the appearances in the text, the two values are weighted to contribute equally to the final weight value.

For each descriptor, c_i, the number of Web pages refer to it, defined by weight w_{i1}, and the number of times it is referred to in the text, defined by weight w_{i2}, are measured. For example, *Hosting* might not appear at all in the original textual information, but the descriptor based on clustered Web pages could refer to it twice in the text and a total of 235 Web pages might be referring to it. The descriptors that receive the highest ranking form the context. The descriptor's weight,w_i, is calculated according to the following steps:

- Set all *n* descriptors in descending weight order according to the number of Web page references:

  $\{\langle c_i, w_{i1}\rangle_{1\le i1\le n-1} | w_{i1} \le w_{i1+1}\}$

  Current References Difference Value, $D(R)_i = \{ w_{i1+1} - w_{i1,1\le i1\le n-1} \}$

- Set all *n* descriptors in descending weight order according to the number of appearances in the text:

$\{\langle c_i, w_{i2}\rangle_{\ 1\leq i2\leq n-1} | w_{i2} \leq w_{i2+1}\}$

Current Appearances Difference Value, $D(A)_i = \{\ w_{i2+1} - w_{i2,1\leq i2\leq n-1}\ \}$

- Let M_r be the Maximum Value of References and M_a be the Maximum Value of Appearances:

  $M_r = \ max_i\{D(R)_i\}$

  $M_a = \ max_i\{D(A)_i\}$

- The combined weight, w_i, of the number of appearances in the text and the number of references in the Web is calculated according to the following formula:

$$w_i = \sqrt{\left(\frac{2 * D(A)_i * M_r}{3 * M_a}\right)^2 + (D(R)_i)^2}$$

The context recognition algorithm consists of the following major phases: i) selecting contexts for each set of tokens, ii) ranking the contexts, and iii) declaring the current contexts. The result of the token extraction is a list of tokens obtained from the textual information. The input to the algorithm is based on the name descriptor tokens extracted from the textual information. The selection of the context descriptors is based on searching the Web for relevant documents according to these tokens and on clustering the results into possible context descriptors. The output of the ranking stage is a set of highest ranking context descriptors. The set of context descriptors that have the top number of references, both in number of Web pages and in number of appearances in the text, is declared to be the context and the weight is defined by integrating the value of references and appearances.

Figure 1 provides the outcome of the Web context extraction method for a DomainSpy web service textual description (see bottom right part). The figure shows only the highest ranking descriptors to be included in the context. For example, *Domain*, *Address*, *Registration*, *Hosting*, *Software*, and *Search* are the context descriptors selected to describe the DomainSpy service.

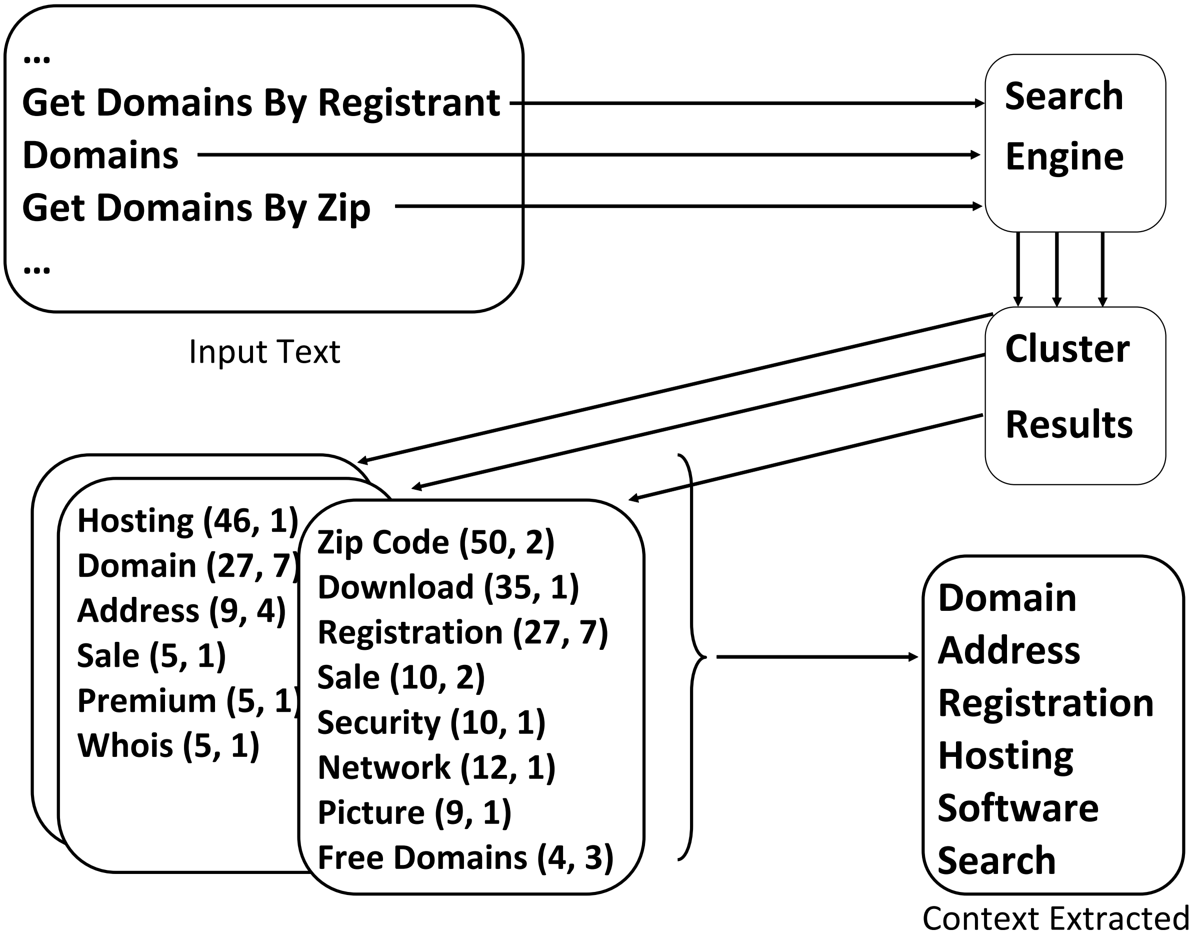


Figure 1 - Example of the Context Extraction Method

2 Ontologies

Ontologies have been defined and used in various research areas, including philosophy (where it was coined), artificial intelligence, information sciences, knowledge representation, object modeling, and most recently, eCommerce applications. In his seminal work, Bunge defines Ontology as a world of systems and provides a basic formalism for ontologies (Bunge, 1979). Typically, ontologies are represented using Description Logic (Borgida & Brachman, 1993)

(Donini, Lenzerini, Nardi, & Schaerf, 1996), where subsumption typifies the semantic relationship between terms, or Frame Logic (Kifer, Lausen, & Wu, 1995), where a deductive inference system provides access to semi-structured data.

Recent work has focused on ontology creation and evolution and in particular schema matching. Many heuristics were proposed for the automatic matching of schemata (e.g., Cupid (Madhavan, Bernstein, & Rahm, 2001), GLUE (Doan, Madhavan, Domingos, & Halevy, 2002), and OntoBuilder (Gal, Modica, Jamil, & Eyal, 2005)), and several theoretical models were proposed to represent various aspects of the matching process (Madhavan, Bernstein, Domingos, & Halevy, 2002; Melnik, 2004; Gal, Anaby-Tavor, Trombetta, & Montesi, 2005).

The realm of information science has produced an extensive body of literature and practice in ontology construction, e.g., (Vickery, 1966). Other undertakings, such as the DOGMA project (Spyns, Meersman, & Jarrar, 2002), provide an engineering approach to ontology management. Work has been done in ontology learning, such as Text-To-Onto (Maedche & Staab, 2001), Thematic Mapping (Chung, Lieu, Liu, Luk, Mao, & Raghavan, 2002), OntoMiner (Davulcu, Vadrevu, & Nagarajan, 2003), and TexaMiner (Kashyap, Dalal, & Behrens, 2001) to name a few. Finally, researchers in the field of knowledge representation have studied ontology interoperability, resulting in systems such as Chimaera (McGuinness, Fikes, Rice, & Wilder, 2000) and Protègè (Noy & Musen, 2000).

The present model of an ontology is based on Bunge's terminology. The aim is to formalize the mapping between contexts and ontologies and provide an uncertainty management tool in the form of concept ranking. When experimenting with the model the assumption is that an ontology is given, designed using any of the tools mentioned above.

An *ontology* O = (V,E) is a directed graph, with nodes representing concepts (things in Bunge's terminology (Bunge, 1977), (Bunge, 1979)) and edges representing relationships (See

Figure 2 (top) for a graphical illustration). A single concept is represented by a name and a context C. The relationship of context and ontology is the focus of the next section.

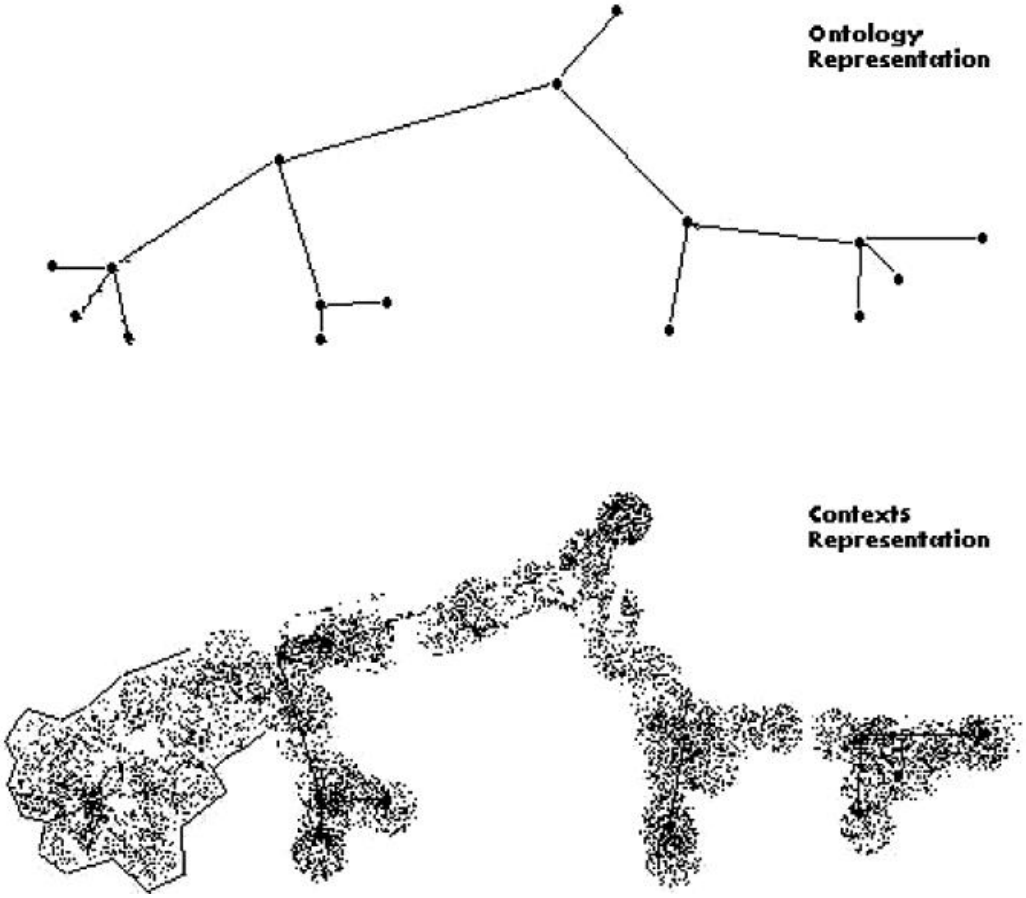


Figure 2 - Contexts and Ontology Concepts

3 Contexts to Ontologies

The relationships between ontologies and contexts can be modeled using topologies as follows. A *topological structure* (topology) in a set X is a collective family $\vartheta = (G_i \mid \mathrm{i} \in \mathrm{I})$ of subsets of X satisfying

1. $J\ finite; \Rightarrow \bigcup_{i\in \mathrm{J}} G_i \in \vartheta$

2. $J \subset \mathrm{I}; \Rightarrow \bigcap_{i\in \mathrm{J}} G_i \in \vartheta$

3. $\emptyset \in \vartheta, \mathrm{X} \in \vartheta$

The pair (X, ϑ) is called a *topological space* and the sets in ϑ are called *closed sets*. A context is now defined to be a closed set in a topology, representing a family ϑ of all possible contexts in some set X with the subset relation $\subseteq$. X is a set of sets of pairs $\langle \mathrm{c}, \mathrm{w} \rangle$, where c is a word (or words) in a dictionary and w is a weight. Note that ϑ is infinite since descriptors are not limited in their length and weights are taken from some infinite number set (such as the real numbers).

The topology is defined by the following subset relation on the context: $\forall \mathrm{C}_{\mathrm{a}} \exists \mathrm{C}_{\mathrm{b}}$ such that $C_a = \left\{\{\langle c_{ij}, w_{ij} \rangle\}_i\right\}_j \subseteq C_b = \left\{\{\langle c_{kp}, w_{kp} \rangle\}_k\right\}_p$. Stating that for each context there exists another context that includes the existing context. Identity between contexts is defined as follows: $C_a = C_b if\ c_{kp} = c_{ij}, w_{kp} = w_{ij}, \forall \mathrm{k}, \mathrm{p}$. Contexts are identical if all descriptors and their matching weights are identical.

The empty set and X are also contexts. Contexts as sets of descriptor sets are closed under intersection and union.

Contexts were previously defined as closed sets. Next the notion of order of contexts can be defined using a directed set. A *directed set* is a set S together with a relation $\geq$, which is both transitive and reflexive, such that for any two elements $a, b \in S$, there exists another element $c \in S$ with $\mathrm{c} \geq \mathrm{a}$ and $\mathrm{c} \geq \mathrm{b}$. In this case, the relation $\geq$ is said to "direct" the set.

A specific directed set is defined using contexts. A context directed set is formally defined by:

$\mathrm{C}_0 = \{\emptyset\}$

$\mathrm{C}_{\mathrm{n}} = \{DS_i, DS_i \cup DS_n | \forall DS_i \in \mathrm{C}_{\mathrm{n}-1}\}$

The definition is illustrated in Figure 3. The different descriptor sets can be viewed as a collection in a bag. One descriptor set $DS_1$is randomly selected. Let Context C_1 define all the descriptor sets that can be created out of one given context - this is only one descriptor set. Let Context C_2 be the sets of descriptors that can be created from two given descriptor sets. Context C_2 contains three descriptor sets: $DS_1$from the previous context, DS_2 which is another descriptor set selected, and the union of both descriptor sets, therefore, $\mathrm{C}_1 \leq \mathrm{C}_2$. It is possible to continue and build this directed set by adding another descriptor set to C_2 forming a new Context C_3, where $\mathrm{C}_1 \leq \mathrm{C}_3$ and $\mathrm{C}_2 \leq \mathrm{C}_3$. This process of creating the directed set can continue indefinitely.

This directed set forms a sequence where: $\mathrm{C}_1 \leq \mathrm{C}_2 \leq \mathrm{C}_3 \leq \cdots \leq \mathrm{C}_\mathrm{n} \leq \cdots$

Whenever a directed set contains contexts that describe a single topic in the real world, such as school or festival, the aim is to ensure that this set of contexts converges to one ontology concept v, representing this topic, i.e., $\mathrm{C}_\mathrm{n} \underset{\mathrm{n}\to\infty}{\to} v$. In topology theory, such a convergence is termed an accumulation point, a point which is the limit of a sequence, also called a *limit point*. Figure 2 (bottom) and Figure 3 illustrate ontology concepts as points of accumulation. The concept can be viewed as delineating a growing set of descriptors forming the context. The borders outline all of the separate descriptors sets which belong to a specific concept. An overlap between descriptors belonging to different concepts is possible, similar to dynamic taxonomies (Sacco, 2000).

To demonstrate the creation of an ontology concept let a context be a set containing a singleton descriptor set $\{\mathrm{Mathematik}, 2\}$. If another singleton descriptor set of $\{\mathrm{Musik}, 2\}$ is added, a new context which contains three descriptor sets is formed: $\big\{\{\mathrm{Mathematik}, 2\}, \{\mathrm{Musik}, 2\}, \{\mathrm{Mathematik}, 2, \mathrm{Musik}, 2\}\big\}$. As the possible sets of descriptors describing documents increase, there is increasing coverage of the accumulation point. The directed set composed of these contexts becomes more descriptive. It is possible to converge to an ontology concept, such as *Long Day School*, defined by a set, to which the context set belongs.

Basically the accumulation point forms the context which includes all the descriptor sets required to define a concept.

With infinite possible contexts, is it possible to ensure the existence of ontology concepts to which these contexts converge? The answer is yes. According to the topological definitions, contexts were defined as a subset of a topological space. All of the subsets forming the contexts were defined to be closed sets. According to Kelley (1969), the following theorem holds in regard to closed sets:

Theorem 1. *A subset of a topological space is closed if and only if it contains the set of its accumulation point.*

According to this theorem, any subset of contexts, being closed sets, will necessarily include an accumulation point. With a finite set of descriptor sets, when each time another descriptor set is added, an accumulation point, which includes all of the descriptors forming the ontology concepts, will be reached. However, the above theorem guarantees that even if there are an infinite number of descriptors sets, an accumulation point, which will also be a context, will eventually be reached. This context will include all of the descriptor sets defining the concept.

The model proposed in (Segev & Gal, 2007a) employs topological definitions to delineate the relationships between contexts and ontologies. A context is a set of descriptors and their corresponding weights. A directed set is a relation of contexts that includes all of their possible unions of sets of descriptors. An ontology concept is the accumulation point of the directed set of contexts.

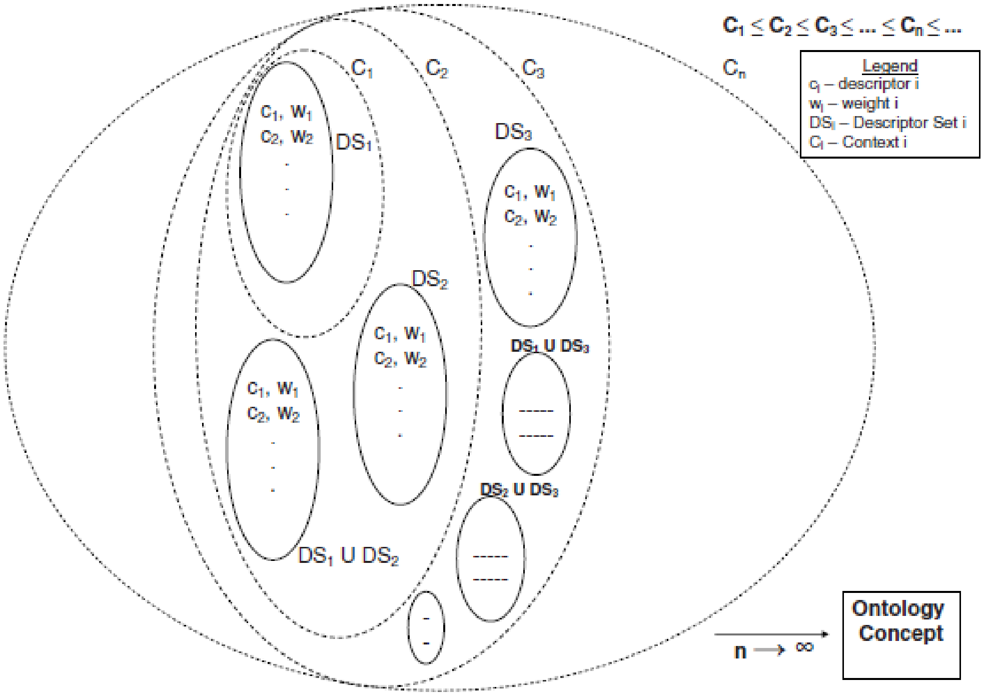


Figure 3 - Contexts Sets Converging to an Ontology Concept

With kind permission from Springer Science+Business Media: Segev A. and Gal A., Putting Things in Context: An Algorithmic Approach to Mapping Contexts to Ontologies, Journal of Data Semantics (JoDS), IX, pp. 113-140, 2007. (©Springer)

4 Web Services

4.1 Related work on Web services

In recent years, the use of services to compose new applications from existing modules has gained momentum. Web services are autonomous units of code, independently developed and evolved. The Web Service Description Language (WSDL) (Christensen, Curbera, Meredith, & Weerawarana, 2001) is used as the de facto standard for service providers to describe the interface of the Web services, i.e., their operations and input and output parameters. Therefore, Web services lack homogeneous structure beyond that of their interface. Heterogeneity stems from different ways to name parameters, define parameters, and describe internal processing. This heterogeneity encumbers straightforward integration between Web services.

Web service registries such as Universal Description, Discovery, and Integration (UDDI) were created to encourage interoperability and adoption of Web services. However, UDDI registries have some major flaws (Platzer & Dustdar, 2005). UDDI registries either are made publicly available and contain many obsolete entries or require registration. In either case, a registry stores only a limited description of the available services.

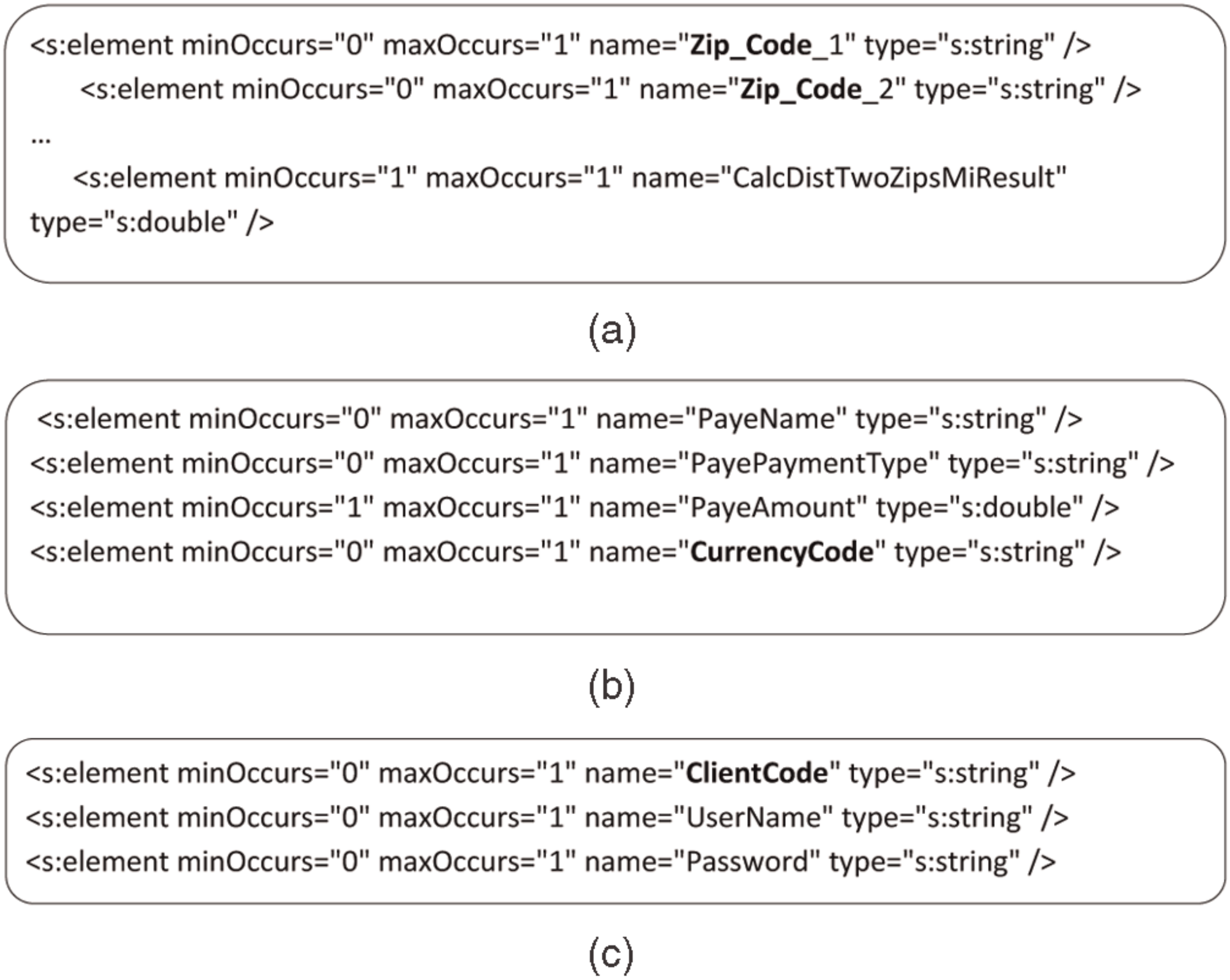
Semantic Web services were proposed to overcome interface heterogeneity. Using languages such as Ontology Web Language for Services (OWL-S) (Ankolekar, Martin, Zeng, Hobbs, Sycara, Burstein, Paolucci, Lassila, Mcilraith, Narayanan, & Payne, 2001) and WSDL Semantics (WSDL-S) (Akkiraju, Farrell, Miller, Nagarajan, Schmidt, Sheth, & Verma, 2005), Web services are extended with an unambiguous description by relating properties such as input and output parameters to common concepts and by defining the execution characteristics of the service. The concepts are defined in Web ontologies (Bechhofer, Harmelen, Hendler, Horrocks, McGuinness, Patel-Schneider, & Stein, 2004), which serve as the key mechanism to globally define and reference concepts. Formal languages enable service composition, in which a developer uses automatic or semiautomatic tools to create an integrated business process from a set of independent Web services.

Service composition in a heterogeneous environment immediately raises issues of evaluating the accuracy of the mapping. As an example, consider three real-world Web services, as illustrated in Figure 4. The three services—distance between zip codes (A), store IT contracts (B), and translation into any language (C)—share some common concepts, such as the code concept. However, these three services originate from very different domains. Service A is concerned with distance calculation and uses the zip codes as input, service B defines

CurrencyCode as part of the IT contract information to be stored, and service C uses a ClientCode as an access key for users. It is unlikely that any of the services will be combined into a meaningful composition. This example illustrates that methods based solely on the concepts mapped to the service's parameters (as in Paolucci, Kawamura, Payne, & Sycara, 2002) may yield inaccurate results.

(Segev & Toch, 2009) aim at analyzing different methods for automatically identifying possible semantic composition. Two sources for service analysis were explored: WSDL description files and free textual descriptors, which are commonly used in service repositories. Three methods for Web service classification are investigated for each type of descriptor: Term Frequency/Inverse Document Frequency (TF/IDF) (Salton & McGill, 1983) and context based analysis (Segev, Leshno, & Zviran, 2007a), and a baseline method. Contexts are defined as a model of a domain for a given term, which is automatically extracted from a fragment of text. Contexts are created by finding related terms from the Web. Unlike ontologies, which are considered shared models of a domain, contexts are defined as local views of a domain (Segev & Gal, 2007a). Therefore, contexts may be different for two fragments of information, even though their domain might be the same. The definition of context used here extends the definition of context in ubiquitous computing, which employs context as any information that can be used to characterize the situation of an entity (Dey, 2000). In many fields, context is used to describe the environment in which a service operates. In this definition, it is used to describe the related set of linguistic terms of a given text.

(Segev & Toch, 2009) propose a context-based approach to the problem of matching and ranking semantic Web services for composition. First, the use of service classification, a process that matches a service to a set of concepts, representing its affinity with a given domain, is proposed. For example, consider the services in Figure 4. The context of service A would be a set of geographical terms (such as address, city, and distance). Therefore, it would be classified to a set of concepts taken from a geographical ontology. Service B would be classified to a business transaction ontology and service C to a computer systems ontology. Second, the classification and context information is used to improve the process of service composition, ruling out compositions of unrelated services. Given a suggested composition between a number of services, the context overlap between the services is analyzed. The overlap is used to rank the probability of the composition.



```
<s:element minOccurs="0" maxOccurs="1" name="Zip_Code_1" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Zip_Code_2" type="s:string" />
…
   <s:element minOccurs="1" maxOccurs="1" name="CalcDistTwoZipsMiResult"
type="s:double" />
```

(a)

```
 <s:element minOccurs="0" maxOccurs="1" name="PayeName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="PayePaymentType" type="s:string" />
<s:element minOccurs="1" maxOccurs="1" name="PayeAmount" type="s:double" />
<s:element minOccurs="0" maxOccurs="1" name="CurrencyCode" type="s:string" />
```

(b)

```
<s:element minOccurs="0" maxOccurs="1" name="ClientCode" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="UserName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Password" type="s:string" />
```

(c)

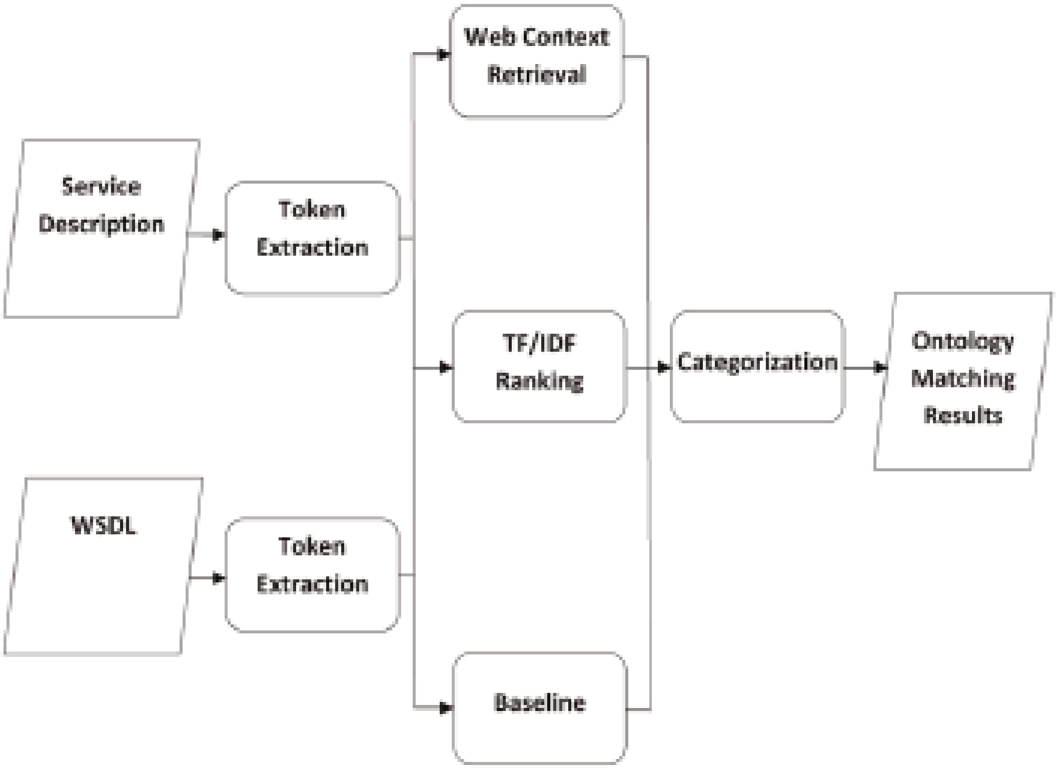
(Segev & Toch, Context-Based Matching and Ranking of Web Services for Composition, IEEE Transactions on Services Computing, 2009) (© 2009 IEEE)

Figure 4 - Service tagging is misleading: an example. (a) Returns distance in miles or kilometers given 2 zip codes. (b) Store IT contracts. (c) Translation into and out of any language.

Figure 5 depicts the stages of the categorization process, including the different methods evaluated. The assumption is that each Web service is described using a textual description, which is part of the metadata within UDDI registries, and a WSDL document describing the syntactic properties of the service interface.

Three methods are examined for the service classification analysis: TF/IDF, Web context extraction, and a baseline for evaluation purposes. The baseline method is a simple reflection (identity function) of the original bag of tokens, extracted from the service descriptions to a bag of tokens representing sets of words. The basic data structure used by all the methods is a ranked bag of tokens, which is processed and updated in the different stages. The results of the service analysis process are used by the TF/IDF and Web context extraction methods. After the different analysis methods were applied, the final categorization is achieved by matching the bag of tokens to the concept names of each of the ontologies.

The field of Web service composition is very active. However, most approaches require clear and formal semantic annotations to formal ontologies (Oh, 2006; Paolucci, Kawamura, Payne, & Sycara, 2002; Akkiraju, Farrell, Miller, Nagarajan, Schmidt, Sheth, & Verma, 2005; Klusch, Fries, Khalid, & Sycara, 2005). Since most services that are currently active in the World Wide Web do not contain any semantic annotations, finding methods that enable composition without semantic annotation is a necessity. Initial work has been done in discovering services directly by querying syntactic Web services through their WSDL documentation (Vouros, Dimitrokallis, & Kotis, 2008; Toch, Gal, & Dori, 2005). (Segev & Toch, 2009) provide an analysis of different ways for extracting information from syntactic Web services and using this information in the context of composition, rather than Web service discovery.



(Segev & Toch, Context-Based Matching and Ranking of Web Services for Composition, IEEE Transactions on Services Computing, 2009) (© 2009 IEEE)

Figure 5 – The Web Service Categorization Process

The field of automatic annotation of syntactic Web services contains several works relevant. (Patil, Oundhakar, Sheth, & Verma, 2004) presented a combined approach towards automatic semantic annotation of Web services. The approach relies on several matchers (string matcher, structural matcher, and synonym finder), which are combined using a simple aggregation function. Duo et al. (Duo, Juan-Zi, & Bin, 2005) presented a similar method, which also aggregates results from several matchers.

(Oldham, Thomas, Sheth, & Verma, 2004) showed that using a simple machine learning technique, namely, Naïve Bayesian Classifier, improves the precision of service annotation.

Machine learning is also used in a tool called Assam (Heß, Johnston, & Kushmerick, 2004), which uses existing annotation of semantic Web services to improve new annotations. While machine learning effectively improves the efficiency of the semantic annotation, the corpus size used for learning is small, as WSDL documents contain very little text. The approach in (Segev & Toch, 2009) is complementary to machine learning methods, as it suggests and provides further information, in the form of textual descriptions and Web context. This information can be used by learning methods to improve annotations.

Another relevant field is search engines for syntactic Web services. Works by (Platzer & Dustdar, 2005; Dong, Halevy, Madhavan, Nemes, & Zhang, 2004) present search engines for WSDL documents. The search engines use a multitude of information retrieval techniques, including vector space representation, TF/IDF, and text clustering. The main drawback of applying these techniques to WSDL is the relatively short content of a WSDL document, which limits the precision and recall of the search engine.

More recently, several works suggested using information about the Web service composition to provide a better annotation process. (Bowers & Ludäscher, 2005) proposed to explore the relation between input and output parameters of the same operation to infer the semantics from the parameters. If the semantics of the input parameter is known and the logic of the operation is known, then the semantics of the output parameter can be inferred automatically.

(Belhajjame, Embury, Paton, Stevens, & Goble, 2008) suggest using information about the composition (the term workflow is used in their work) in which the service is used. The composition structure reveals operational constraints between parameters of different operations and can be used to support or disqualify annotations. The aforementioned work by (Bowers & Ludäscher, 2005) shows the potential of using external information for improving annotations. (Segev & Toch, 2009) share a similar vision, arguing for the utilization of external information. However, the intention is to produce domain-specific semantic annotation rather than operational semantics. Therefore, the Web and public ontologies, rather than the workflow or procedural description of the Web services, are used as information resources.

Context-based semantic matching for Web services composition has become a focus of interest. An initial prior work describes a context mediator that facilitates semantic interoperability between heterogeneous information systems (Sciore, Siegel, & Rosenthal, 1994). A recent work presents a context-based mediation approach (Mrissa, Ghedira, Benslimane, Maamar, Rosenberg, & Dustdar, 2007) which was used to solve semantic heterogeneities between composed Web services.

5 Bootstrapping Ontologies

Ontologies are used in an increasing range of applications, notably the Semantic Web, and essentially have become the preferred modeling tool. However, the design and maintenance of ontologies is a formidable process (Noy & Klein, 2004; Kim, Lee, Shim, Chun, Lee, & Park, 2005). Ontology bootstrapping, which has recently emerged as an important technology for ontology construction, involves automatic identification of concepts relevant to a domain and relations between the concepts (Ehrig, Staab, & Sure, 2005).

Previous work on ontology bootstrapping focused on either a limited domain (Zhang, Troy, & Bourgoin, 2006) or expanding an existing ontology (Castano, Espinosa, Ferrara, Karkaletsis, Kaya, Melzer, Moller, Montanelli, & Petasis, 2007). In the field of Web services, registries such as the Universal Description, Discovery, and Integration (UDDI) have been created to encourage interoperability and adoption of Web services. A registry only stores a limited description of the available services. Ontologies created for classifying and utilizing Web services can serve as an alternative solution. However, the increasing number of available Web services makes it difficult to classify Web services using a single domain ontology or a set of existing ontologies created for other purposes. Furthermore, a constant increase in the number of Web services requires continuous manual effort to evolve an ontology.

The Web service ontology bootstrapping process described here is based on the advantage that a Web service can be separated into two types of descriptions: i) the Web Service Description Language (WSDL) describing "how" the service should be used and ii) a textual description of the Web service in free text describing "what" the service does. This advantage allows bootstrapping the ontology based on WSDL and verifying the process based on the Web service free text descriptor.

The ontology bootstrapping process is based on analyzing a Web service using three different methods, where each method represents a different perspective of viewing the Web service. As a result, the process provides a more accurate definition of the ontology and yields better results. In particular, the Term Frequency/Inverse Document Frequency (TF/IDF) method analyzes the Web service from an internal point of view, i.e., what concept in the text best describes the WSDL document content. The Web Context Extraction method describes the WSDL document from an external point of view, i.e., what most common concept represents the answers to the Web search queries based on the WSDL content. Finally, the Free Text Description Verification method is used to resolve inconsistencies with the current ontology. An ontology evolution is performed when all three analysis methods agree on the identification of a new concept or a relation change between the ontology concepts. The relation between two concepts is defined using the descriptors related to both concepts.

This approach can assist in ontology construction and reduce the maintenance effort substantially. The approach facilitates automatic building of an ontology that can assist in expanding, classifying, and retrieving relevant services, without the prior training required by previously developed approaches.

5.1 The bootstrapping ontology model

The bootstrapping ontology model proposed in (Segev & Sheng, 2011) is based on the continuous analysis of WSDL documents and employs an ontology model based on concepts and relationships (Gruber, 1993). The innovation of this proposed bootstrapping model centers on i) the combination of the use of two different extraction methods, TF/IDF and Web based concept generation, and ii) the verification of the results using a Free Text Description Verification method by analyzing the external service descriptor. These three methods are utilized to demonstrate the feasibility of the model. It should be noted that other more complex methods, from the field of Machine Learning (ML) and Information Retrieval (IR), can also be used to implement the model. However, the use of the methods in a straightforward manner emphasizes that many methods can be "plugged in" and that the results are attributed to the model's process of combination and verification. (Segev & Sheng, 2011) integrated these three specific methods since each method presents a unique advantage - internal perspective of the Web service by the TF/IDF, external perspective of the Web service by the Web Context Extraction, and a comparison to a free text description, a manual evaluation of the results, for verification purposes.

The overall bootstrapping ontology process is described in Figure 6. There are four main steps in the process. The token extraction step extracts tokens representing relevant information from a WSDL document. This step extracts all the name labels, parses the tokens, and performs initial filtering.

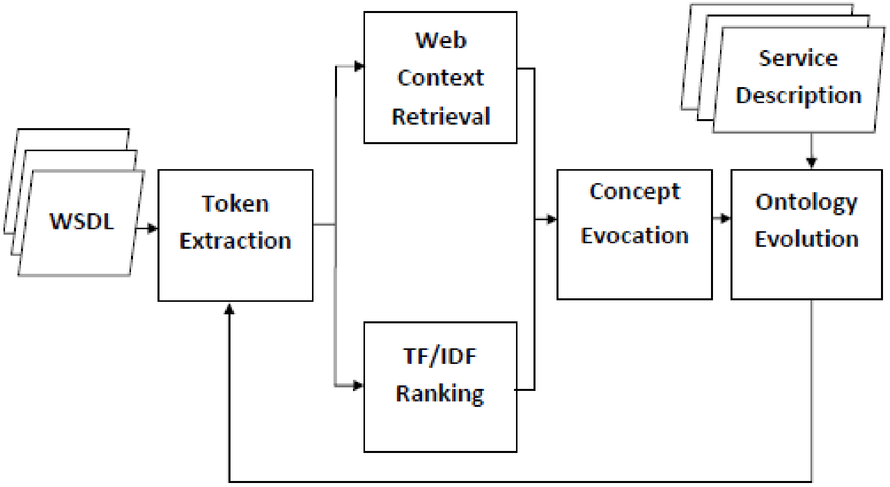
The second step analyzes in parallel the extracted WSDL tokens using two methods. In particular, TF/IDF analyzes the most common terms appearing in each Web service document and appearing less frequently in other documents. Web Context Extraction uses the sets of tokens as a query to a search engine, clusters the results according to textual descriptors, and classifies which set of descriptors identifies the context of the Web service.

The concept evocation step identifies the descriptors which appear in both the TF/IDF method and the Web context method. These descriptors identify possible concept names that could be utilized by the ontology evolution. The context descriptors also assist in the convergence process of the relations between concepts.

Finally, the ontology evolution step expands the ontology as required according to the newly identified concepts and modifies the relations between them. The external Web service textual descriptor serves as a moderator if there is a conflict between the current ontology and a new concept. Such conflicts may derive from the need to more accurately specify the concept or to define concept relations. New concepts can be checked against the free text descriptors to verify the correct interpretation of the concept. The relations are defined as an ongoing process according to the most common context descriptors between the concepts. After the ontology evolution, the whole process continues to the next WSDL with the evolved ontology concepts and relations. It should be noted that the processing order of WSDL documents is arbitrary.

The main contributions of this work are as follows:

- On a conceptual level, an ontology bootstrapping model, a model for automatically creating the concepts and relations “from scratch”, is introduced.
- On an algorithmic level, an implementation of the model in the Web service domain is provided, using integration of two methods for implementing the ontology construction and a Free Text Description Verification method for validation using a different source of information.



(Segev & Sheng, Bootstrapping Ontologies for Web Services, IEEE Transactions on Services Computing, 2011) (© 2011 IEEE)

Figure 6 - Web Service Ontology Bootstrapping Process

6 Applications

6.1 Medical diagnostic assistance

This section presents a Web-based technique of integrating context recognition and computer vision and demonstrates how this method can be implemented. Usually document analysis focuses on the text part of a document, but (Segev, Leshno, & Zviran, 2007b) proposes an idea of text understanding by understanding image first, since image can constitute a rich source of information. This idea is based on the assumption that the accuracy of computer vision is high enough to provide a useful hint for context recognition, since an inaccurate computer vision system might also mislead the overall context recognition.

The integration method (Segev, Leshno, & Zviran, 2007b) yields improved results in comparison to the separate use of context recognition or TF/IDF methods. Additionally, use of state-of-the-art as opposed to simple computer vision algorithms can improve the results.

The main advantage of the model for the integration of computer vision into context recognition is its use of the Web as a knowledge base for data extraction. The information provided by the computer vision model complements and augments the context recognition process by reducing the number of incorrect diagnoses.

To analyze information consisting of both text and images a model of the integration of both methods is described in Figure 7. The input is separated into text and image. The next step implements a context recognition model for textual analysis and a computer vision model for image analysis. Then the vision is integrated into context, yielding conceptual output. For example, in the field of medicine, the model input can be a medical case study and the model output is a list of words that represent major symptoms or possible diagnoses and these words are checked against the solutions in the medical case studies.

The main advantage of both the Web context method and the integrated computer vision and Web context over the TF/IDF is the ability to identify a symptom or cause of death which does not appear in the text itself. While the latter has to work within the limits of the original case study text, the context analysis method goes out to the Web, using it as an external judge and returning keywords that are deemed relevant, although they were not originally specified in the case description.

The advantage of the integrated computer vision and Web context model compared to the Web context model can be seen in the reduction of the false positive results. Although the Web context by itself in most cases returns the correct results, the ranking of the result is not always high in the result list. The computer vision results allow the identification of which context results should receive higher ranking and consequently the model identifies the correct diagnosis or relevant symptom.

The model achieved high results in both identifying diagnoses that include the identification of the correct diagnosis and identifying symptoms for correct diagnosis. A possible implementation of the model could include a decision support system for a physician analyzing a case. Alternatively, an implementation of the model could be used as a second opinion tool for the patient or his family. Since the model in most cases supplies a list of diagnoses, including the correct diagnosis, a physician would be able to receive an extended list and rule out the incorrect diagnoses.

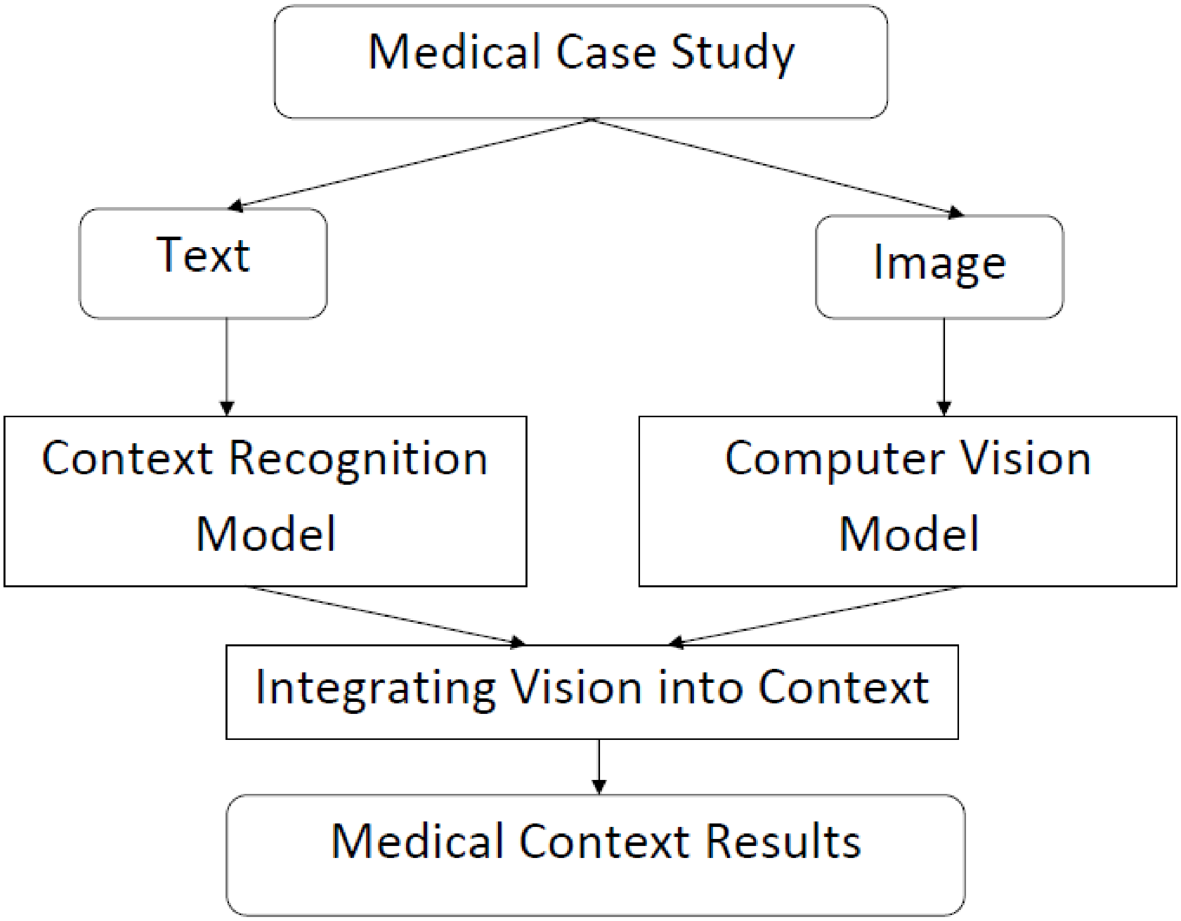


Figure 7 - Medical Diagnostic Assistance Method Outline

6.2 Multilingual Decision Support Systems

Experiences in developing information systems have shown it to be a long and expensive process. Therefore, once a generic information system has been developed, it is the aim of the developer to make it as portable as possible and the aim of users to deploy it with minimum effort. In some cases, such deployment requires the change of language, which affects the user interface as well as the internal decision making processes. This section focuses on applications in which a language transfer serves as a main obstacle in adapting an information system to user needs.

As a case in point, consider eGovernment applications in the European Union. The EU puts effort into homogenizing its governance procedures to allow easy interoperability. Yet it does so without committing to a single language. On the contrary, the EU values the preservation of local culture (including language). In such applications, the development of an information system that is monolingual will result in low portability and high deployment costs and therefore multilingual information systems seem to be more appropriate.

Recent advances in information system development suggest the use of ontologies as a main knowledge management tool. Ontologies model the domain of discourse and may be used for routing data, controlling the workflow of activities, assisting in semantic annotation of both data and queries, etc. To take advantage of these recent advances, an ontology-based model for multilingual knowledge management in information systems was proposed in (Segev & Gal, 2008). The mechanism is based on a single ontology, whose concepts can have multiple representations (i.e., concept names) in various languages. While such solutions already exist (e.g., in Protégé), it is argued that they are insufficient. On the one hand, a single global ontology is preferred over local ontologies when it comes to interoperability. On the other hand, mere translation of ontological concepts from one language to another is insufficient to fully represent differences that may arise from the change of language. Such differences may result in concept ambiguity and generally in under-specification of semantic meaning (Gal & Segev, 2006).

To compensate for ontology under-specification, multilingual ontologies can be supported with a lightweight mechanism, dubbed context. Contexts serve in the literature to represent local views of a domain, as opposed to the global view of an ontology (Gruber, 1993). While the specific representation of contexts vary, one may envision a context, as an example, to be represented by a set of words, possibly associated with weights, reflecting some notion of importance. Contexts, in this solution, are associated with ontological concepts and specified in multiple languages. Therefore, contexts aim at conveying the local interpretation of ontological concepts, thus assisting in the resolution of cross-language and local interpretation ambiguities.

To summarize, the main contributions are as follows:

- The knowledge management model is based on the relationships between ontologies and contexts, thus supporting effective portability and deployment of multilingual information systems.
- The high degree of flexibility this model provides is translated into procedures for the deployment and querying of a multilingual information system.
- The feasibility of the model is demonstrated using an implementation and deployment in the context of a European eGovernment project.

6.2.1 Ontologies, contexts, and multilingual knowledge management

Now a model for multilingual knowledge management using ontologies and context is described. A common definition of an ontology considers it to be “a specification of a conceptualization” (Gruber, 1993), where conceptualization is an abstract view of the world represented as a set of objects. An *ontology* O = (V,E) is a directed graph, with nodes representing concepts (vocabulary or things (Bunge, 1977), (Bunge, 1979)) associated with certain semantics and relationships (Russell & Norving, 2003). For example, in eGovernment a concept can be *Public Service* with a relation *includes* to a concept *Activity of Public Administration* and a relation *responsibility* to a concept *Local Spatial Management Strategic Plan*.

Each descriptor c can be considered to be a different point of view of some concept $v \in V$. A descriptor set then defines different perspectives and their relevant weight, which identifies the importance of each perspective. For example, an ontology concept *Local Spatial Management Strategic Plan* can be represented by descriptors such as: $\langle \mathrm{Immovables}, 40 \rangle, \langle \mathrm{Building},\ 25 \rangle, \langle \mathrm{Infrastructure}, 20 \rangle$, etc. It can now be assumed that each descriptor set represents a different language and then a context is a multilingual representation of a concept.

The model associates an ontology concept with a name and a context. A multiple-name support mechanism is extended and multiple-context support is proposed in a similar fashion. A concept is associated with multiple contexts. (Segev & Gal, 2007a) defined a context algebra that is closed under the union operator and therefore multiple contexts are in themselves a context, each in a different language. Figure 8 provides a schematic illustration of the model for multilingual knowledge management. Four ontology concepts are displayed: *Public Service, Citizen, Activity of Public*, and *Local Spatial*. Each one has concept names also in French, German, and Polish. For the Local Spatial concept, a set of contexts represents the local perspective of the concepts in both English and Polish.

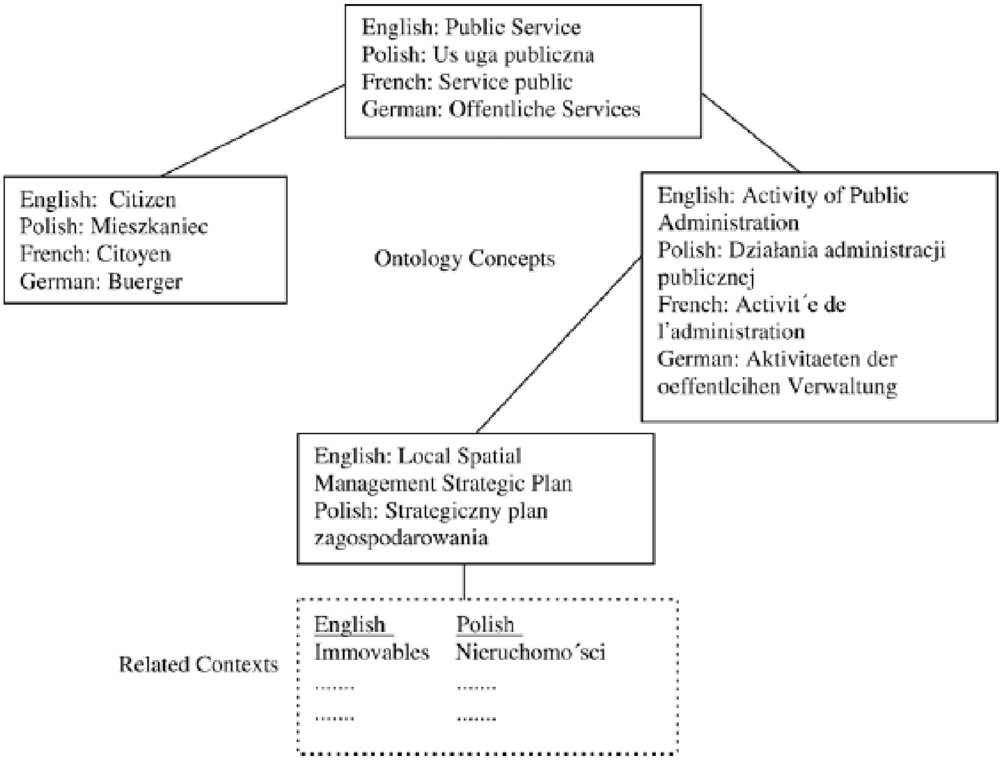

English: Public Service
Polish: Us uga publiczna
French: Service public
German: Offentliche Services
English: Citizen
Polish: Mieszkaniec
French: Citoyen
German: Buerger
Ontology Concepts
English: Activity of Public Administration
Polish: Działania administracji publicznej
French: Activit´e de l'administration
German: Aktivitaeten der oeffentlcihen Verwaltung
English: Local Spatial Management Strategic Plan
Polish: Strategiczny plan zagospodarowania
Related Contexts
English
Polish
Immovables
Nieruchomo´sci

Figure 8 – Multilingual Ontology Example

6.3 Multilingual crisis knowledge representation

6.3.1 Crisis ontology

In the quest to identify frameworks, concepts, and models for crisis ontologies the term 'Open Ontology' was addressed in (Di Maio, 2007). 'Open Ontology' refers to a given set of agreed terms, in terms of conceptualization and semantic formalization, that has been developed based on public consultation and that embodies, represents, and synthesizes all available valid knowledge thought to pertain to a given domain and necessary to fulfill a given functional requirement.

The Sphere handbook (Sphere Project, 2004) is designed for use in disaster response and may also be useful in disaster preparedness and humanitarian advocacy. It is applicable in a range of situations where relief is required, including natural disasters and armed conflict. It is designed for use in both slow- and rapid-onset situations, rural and urban environments, developing and developed countries, anywhere in the world. The emphasis throughout is on meeting the urgent survival needs of people affected by disaster, while asserting their basic human right to life with dignity.

Analysis of the Sphere handbook index, displayed in Figure 9, indicates that it meets many requirements of Open Ontology. Thus, the current index can be defined as an Index Ontology. Generic top level requirements for an Open Ontology according to (Di Maio, 2007) include:

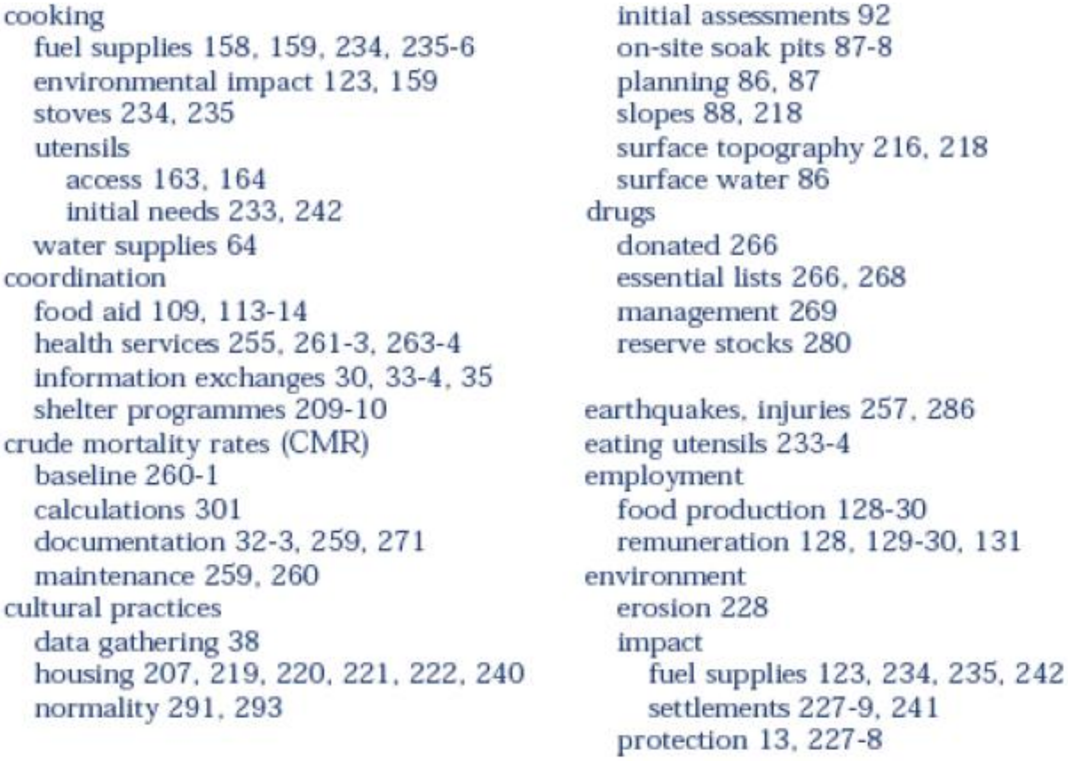
- Declaring what high level knowledge (upper level ontology) it references. The Index Ontology primary concepts can be identified by the outer level keywords in the index. These keywords serve as a high level framework defining the primary topics of the Crisis Ontology.
- The ontology allows reasoning / inference based on the index. For example, according to Figure 9 the concept *fuel supplies* is related to the class of *cooking* and also related to the concept *impact*, which is related to the concept *environment*. It is also related to the concept *vulnerable groups*. The relational index structure supplies the initial structure of the Index Ontology.
- Natural language queries can be supported by simple string matching of words from the query against the Index Ontology concepts. The request to receive relevant information appearing in Figure 10, which shows a blog entry posted by a New Orleans resident, displays an example of a textual natural language query which could be analyzed using the Index Ontology. Simple string matching between the text and the Index Ontology can identify relevant topics such as: *food/water/medicine* and *personal hygiene*, which appear in the Index. The relevant page numbers of the index topics can supply

immediate relevant information delivered in response to the query in any of the above topics. These could include a short description and possible values required to maintain minimal standards in areas such as *personal hygiene*. A simple Web interface could support an online connection between the blog and the Index Ontology, allowing immediate response.

- Use of the Index Ontology supplies an easy-to-understand mechanism with which most users are familiar. The skills required to utilize the ontology are minimal and can be implemented by any ontology tool, such as Protégé (Noy and Musen, 2000) or Topic Maps Ontopia (Pepper, 1999).
- The 'high level knowledge' represented by the Index Ontology can easily be linked to classes representing required actions such as: status updates, email notification of current crisis situation, resources required for the survivors, and critical locations where immediate intervention is required. The current ontology representation already includes values that can be represented as properties such as measuring acute malnutrition in children under five years and other age groups.
- The implementation of the ontology is independent of any ontology language. It can be implemented in any currently used ontology language such as OWL/DARPA Agent Markup Language (DAML) and due to its simplicity can be implemented by alternative ontology languages such as Topics, Associations, and Occurrences (TAO) of topic maps.
- The adoption of an Index Ontology allows a flexible approach to ontology creation and adoption. As the following section describes, the ontology can be

expanded using additional Index Ontologies or alternatively direct links to information on the Web.

- Finally the basic ontology and the knowledge it represents are already defined in multiple languages, allowing multiple viewpoints of similar information in multiple languages. Furthermore, it allows information in multiple languages to be directed to identical ontology concepts.

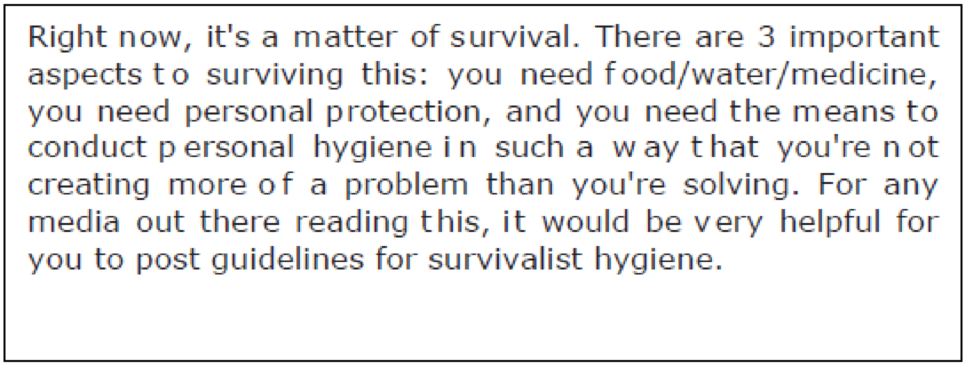


- cooking
  - fuel supplies 158, 159, 234, 235-6
  - environmental impact 123, 159
  - stoves 234, 235
  - utensils
    - access 163, 164
    - initial needs 233, 242
  - water supplies 64
- coordination
  - food aid 109, 113-14
  - health services 255, 261-3, 263-4
  - information exchanges 30, 33-4, 35
  - shelter programmes 209-10
- crude mortality rates (CMR)
  - baseline 260-1
  - calculations 301
  - documentation 32-3, 259, 271
  - maintenance 259, 260
- cultural practices
  - data gathering 38
  - housing 207, 219, 220, 221, 222, 240
  - normality 291, 293

  - initial assessments 92
  - on-site soak pits 87-8
  - planning 86, 87
  - slopes 88, 218
  - surface topography 216, 218
  - surface water 86
- drugs
  - donated 266
  - essential lists 266, 268
  - management 269
  - reserve stocks 280

- earthquakes, injuries 257, 286
- eating utensils 233-4
- employment
  - food production 128-30
  - remuneration 128, 129-30, 131
- environment
  - erosion 228
  - impact
    - fuel supplies 123, 234, 235, 242
    - settlements 227-9, 241
  - protection 13, 227-8

Figure 9 - Index of Humanitarian Charter and Minimum Standards



Right now, it's a matter of survival. There are 3 important aspects to surviving this: you need food/water/medicine, you need personal protection, and you need the means to conduct personal hygiene in such a way that you're not creating more of a problem than you're solving. For any media out there reading this, it would be very helpful for you to post guidelines for survivalist hygiene.

Figure 10 – Sample Blog Posting during Katrina Crisis – August 20th, 2005

6.3.2 Ontology design

This section presents the ontology design process. The first section shows how concepts are extracted from predefined research presented in a book or on-line documentation to construct the ontology layout. The following section displays how to extract the concept relations. Next, the section depicts how the ontology can be expanded and similar documents based on similar concepts can be added to the ontology. The last section shows how the ontology can function in a multilingual environment.

6.3.3 Extracting the ontology layout

Based on the Sphere Handbook index (Sphere Project, 2004), an initial ontology can be constructed using existing hierarchical and semantic relations. Furthermore, data linking to additional information can be stored as class properties. Figure 11 displays a sample of the Index Ontology created from the Sphere Handbook index (Figure 9). The class defined as *cooking* is defined as a super-class of four subclasses: *fuel supplies, environmental impact, water supplies*, and *stoves*. However, *fuel supplies* is a subclass of two additional classes: *vulnerable groups* and *impact*. Similarly, *water supplies* is a subclass of both *cooking* and *vulnerable groups*. The properties of the class *personal hygiene* can match the class with additional information regarding hygiene in the Sphere Handbook, such as full description pages or relevant values. Additionally, external information extracted from other resources can be matched with the extracted Index Ontology.

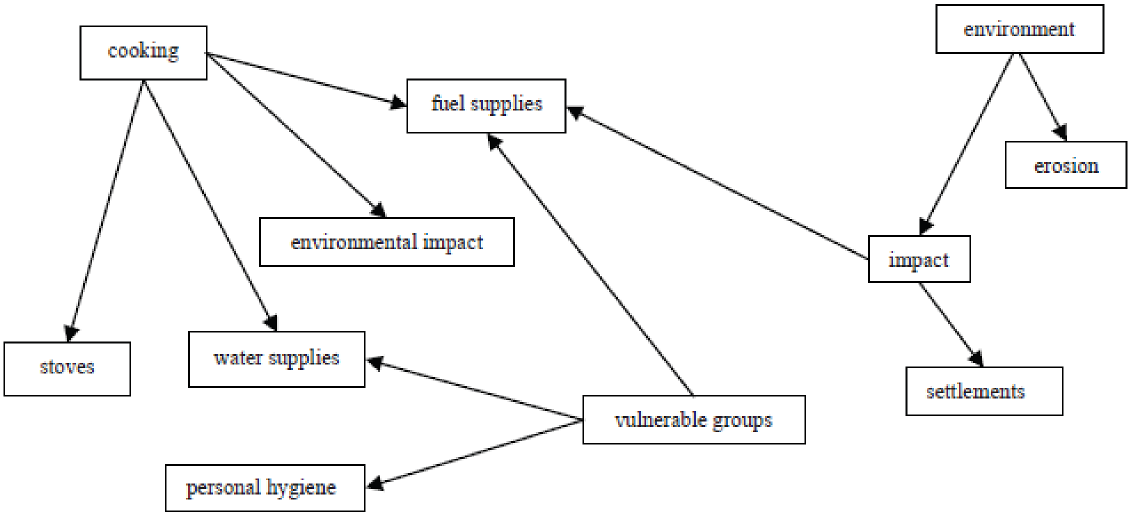


Figure 11 – A Sample of the Extracted Index Ontology

6.3.4 Extracting the concept relations

The ontology concept relations can be extracted in a similar technique, using the book index. The binary relation is defined as the chapter title shared by each of two concepts. For example, in the Sphere Handbook, for each two concepts appearing in the Index Ontology, the chapter title which connects the two can be defined as the relation.

Figure 12 displays an example of the relations of the cooking concept with another four concepts. In the example it can be seen that the relation of *tools and equipment and lighting* describes both *cooking* and *fuel supply* and *cooking* and *stoves*. The relation that can be automatically extracted in this case supplies an appropriate description.

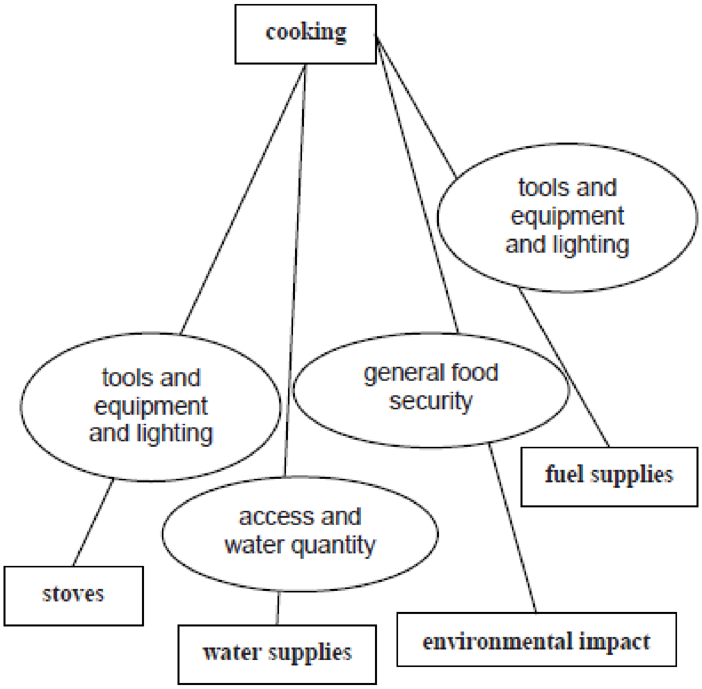
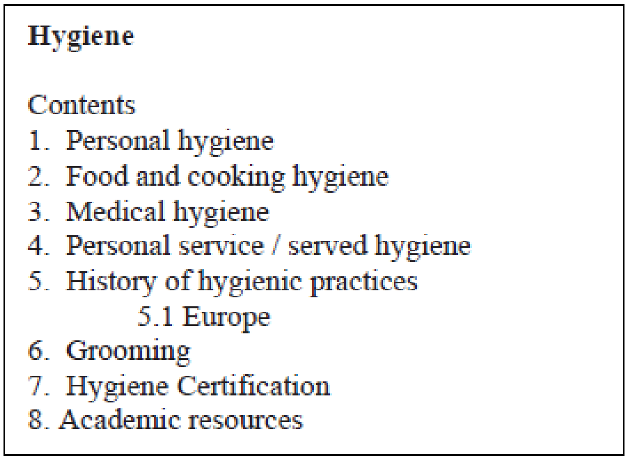

cooking
tools and equipment and lighting
tools and equipment and lighting
general food security
fuel supplies
access and water quantity
stoves
water supplies
environmental impact

Figure 12 – Ontology Concept Relations Based on Document Sections

6.3.5 Expanding the ontology

The ontology can be expanded using external information from other resources such as additional data based on books or websites. For example, the Wikipedia web site for hygiene includes index information which could be added to the current Index Ontology using similar class definitions. Figure 13 displays index information from the Wikipedia hygiene index that can be used as concepts for possible ontology expansion. Notice that the concept *personal hygiene* is a subclass of *hygiene* according to this definition. Figure 14 displays the ontology expansion based on the Wikipedia hygiene entry. Alternatively, additional index books considered fundamental in the field can be added to the ontology. For example, the Merck Manual of Medical Information (Beers, 2003) index can be used for medical class expansion.

There are multiple approaches to merging ontologies such as the Formal Concept Analysis described in (Stumme & Maedche, 2001). Possible merging operations for the ontology engineer are presented in (Noy & Klein, 2003). Furthermore, (Segev & Gal, 2007b) proposed using (machine generated) contexts as a mechanism for quantifying relationships among concepts. Using this model has an advantage since it provides the ontology administrator with an explicit numeric estimation of the extent to which a modification "makes sense." The present research adopts the method of expanding the ontology based on context mechanism.



Hygiene

Contents

1. Personal hygiene
2. Food and cooking hygiene
3. Medical hygiene
4. Personal service / served hygiene
5. History of hygienic practices
    5.1 Europe
6. Grooming
7. Hygiene Certification
8. Academic resources

Figure 13 – Possible Concepts Expansion Based on Wikipedia Indexing

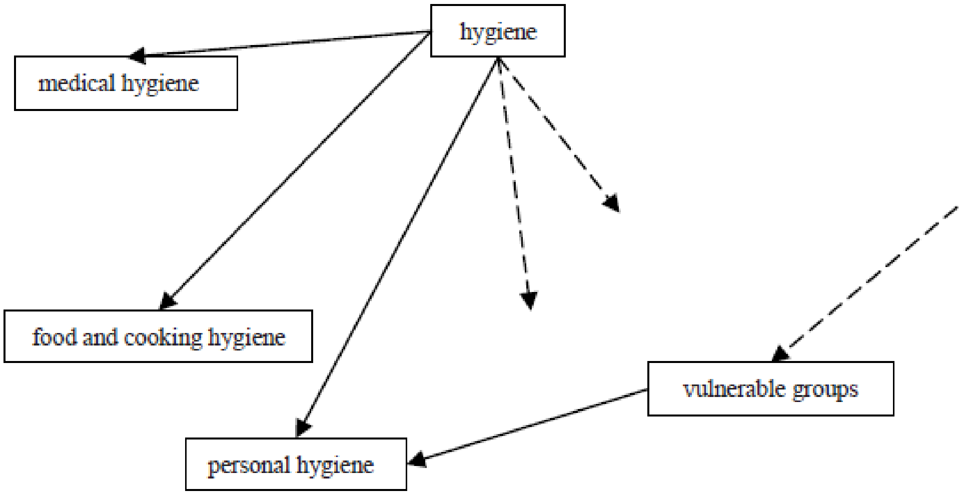


Figure 14 – Ontology Expansion Based on Wikipedia

6.3.6 Multilingualism in crisis management

As aforementioned, an ontology-based model for multilingual knowledge management in information systems has been proposed in (Segev & Gal, 2008). The unique feature was a lightweight mechanism, dubbed context, which is associated with ontological concepts and specified in multiple languages. The contexts were used to assist in resolving cross-language and local variation ambiguities. The technique (described in Section 7.1) can be adopted to build an ontology where each concept can be represented in multiple languages.

The technique presented here is different from the previous model since it requires the ability to create and modify the ontology in real-time as the crisis arises and continues to evolve. This requirement necessitates having a basic predefined multilingual ontology while allowing the expansion of the ontology according to the crisis circumstances and the addition of other languages within the crisis time limitations. The technique can be adopted to build an ontology where each concept can be represented in multiple languages and can be expanded for use in crises, such as the Boxing Day Tsunami.

The Sphere handbook (Sphere Project, 2004) is designed for use in disaster response and was translated into 37 languages. Thus it supplies a top level ontology that can be used concurrently in multiple languages. Since each high level Index Ontology concept is represented in multiple languages, there is faster ontology adaptation in crisis situations. A sample of a multilingual ontology in English, French (F), Tamil (T), and Sinhala (S) is presented in Figure 15.

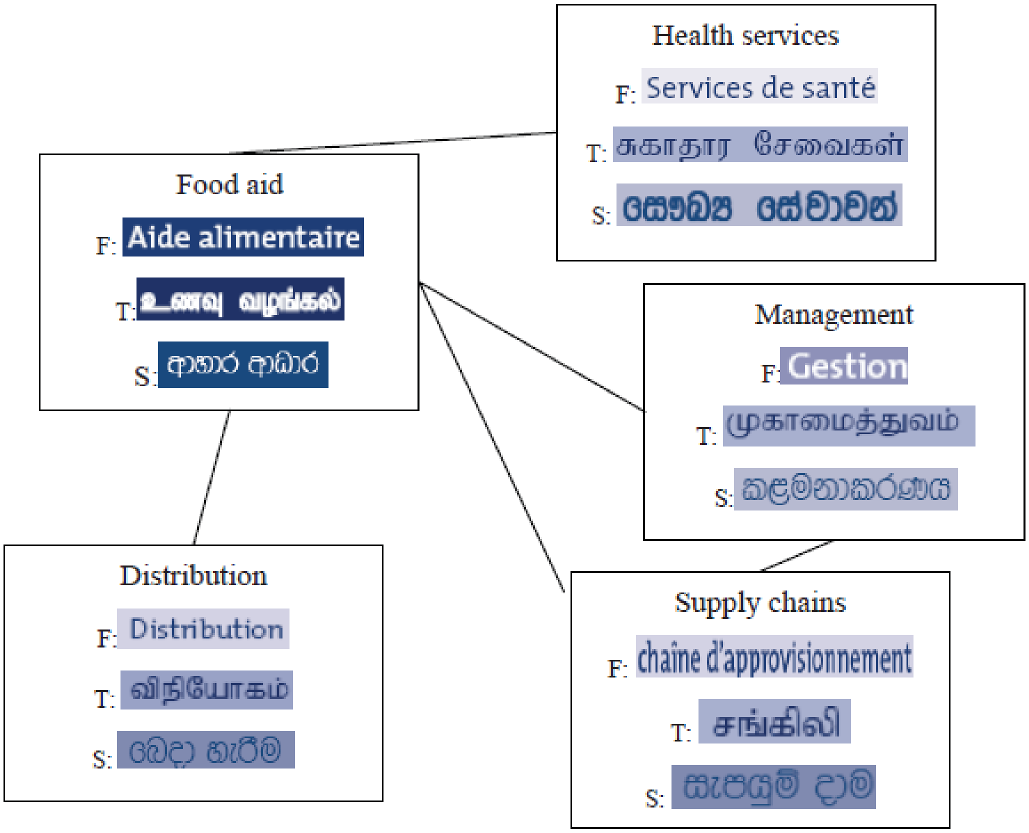


Figure 15 – A Sample of the Extracted Multilingual Ontology

7 Conclusion

This chapter has presented a review of knowledge management that uses context and ontology. The knowledge analysis was initially based on extracting the relevant context. Next the ontology was provided as a outline for representing the framework for organizational knowledge. Mapping from context to ontology is a tool for linking knowledge for representation and extraction. The topic of the matching and composition of Web services was described and bootstrapping ontologies for Web services was discussed. Knowledge management applications were presented in the fields of medical analysis, multilingual decision support systems, and crisis response systems.

References

Aitchison, J., Gilchrist, A., & Bawden, D. (1997). *Thesaurus construction and use: A practical manual* (3rd ed.). London: Aslib.

Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M. T., Sheth, A., & Verma, K. (2005). WSDL-S Web Service Semantics, *W3C Candidate Recommendation*, http://www.w3.org/Submission/WSDL-S/.

Ankolekar, A., Martin, D., Zeng, Z., Hobbs, J., Sycara, K., Burstein, B., Paolucci, M., Lassila, O., Mcilraith, S., Narayanan, S., & Payne, P. (2001). DAML-S: Semantic Markup for Web Services, *Proceedings of International Semantic Web Workshop (SWWS '01)*.

Arens, Y., Knoblock, C. A., & Shen, W. (1996). Query reformulation for dynamic information integration. In G. Wiederhold (Ed.), *Intelligent integration of information* (pp. 11–42). Boston: Kluwer.

Assadi, H. (1998). Construction of a regional ontology from text and its use within a documentary system. *Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS-98)*. Amsterdam: IOS.

Bechhofer, S., Harmelen, F. van, Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., & Stein, L. (2004). OWL Web Ontology Language Reference, *W3C, W3C Candidate Recommendation*, http://www.w3.org/TR/owl-ref/.

Beers, M., ed. (2003). *The Merck Manual of Medical Information*, Merck Research Laboratories, second edition.

Belhajjame, K., Embury, S. M., Paton, N. W., Stevens, R., & Goble, C. A. (2008). Automatic Annotation of Web Services Based on Workflow Definitions, *ACM Transactions Web*, 2(2), 1-34.

Borgida, A., & Brachman, R. J. (1993). Loading data into description reasoners. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 217–226. ACM Press, New York 3.

Bowers, S. & Ludäscher, B. (2005). Towards Automatic Generation of Semantic Types in Scientific Workflows, Proceedings International Workshop Scalable Semantic Web Knowledge Base Systems (SSWS ’05), 207- 216.

Bunge, M. (1977). *Treatise on Basic Philosophy: vol. 3: Ontology I: The Furniture of the World*. D. Reidel Publishing Co., Inc., New York.

Bunge, M. (1979). *Treatise on Basic Philosophy: vol. 4: Ontology II: A World of Systems*. D. Reidel Publishing Co., Inc., New York.

Buvac, S. (1996). Resolving lexical ambiguity using a formal theory of context, semantic ambiguity and underspecification. *CLSI lecture notes* (pp. 1–24).

Castano, S., Espinosa, S., Ferrara, A., Karkaletsis, V., Kaya, A., Melzer, S., Moller, R., Montanelli, S., & Petasis, G. (2007). Ontology Dynamics with Multimedia Information: The BOEMIE Evolution Methodology, *Proceedings of the International Workshop on Ontology Dynamics (IWOD’07), held with the 4th European Semantic Web Conference (ESWC’07)*, Innsbruck, Austria.

Christensen, E. , Curbera, F., Meredith, G., & Weerawarana, S. (2001). WSDL Web Services Description Language, *W3C Candidate Recommendation*, http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

Chung, C. Y., Lieu, R., Liu, J., Luk, A., Mao, J., & Raghavan, P. (2002). Thematic mapping from unstructured documents to taxonomies. *Proceedings of the 11th International Conference on Information and Knowledge Management* (CIKM).

Davulcu, H., Vadrevu, S., & Nagarajan, S. (2003). Ontominer: Bootstrapping and populating ontologies from domain specific websites. *Proceedings of the First International Workshop on Semantic Web and Databases*.

Dey, A. K., (2000). Providing Architectural Support for Building Context- Aware Applications,” PhD thesis, *Georgia Inst. of Technology*.

Di Maio, P. (2007). An Open Ontology for Open Source Emergency Response System, *Open Source Research Community*.

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2002). Learning to map between ontologies on the semantic web. *Proceedings of the Eleventh International Conference on World Wide Web,* Honolulu, Hawaii, USA, pp. 662–673. ACM Press, New York.

Dong, X., Halevy, A. Madhavan, J., Nemes, E., & Zhang, J. (2004). Similarity Search for Web Services, Proc. *Proceedings International Very Large Data Bases*, 372-383.

Donini, F. M., Lenzerini, M., Nardi, D., & Schaerf, A. (1996). Reasoning in description logic. Brewka, G. (ed.) *Principles on Knowledge Representation, Studies in Logic, Languages and Information*, pp. 193–238. CSLI Publications.

Dumais, S., & Chen, H. (2000). Hierarchical classification of web content. *Proceedings of SIGIR, 23rd ACM International Conference on Research and Development in Information Retrieval*, Athens (pp. 256–263).

Duo, Z. , Juan-Zi, L., & Bin, X. (2005). Web Service Annotation Using Ontology Mapping, *Proceedings IEEE International Workshop Service-Oriented System Eng. (SOSE '05)*, 243-250.

Ehrig, M., Staab, S., & Sure, Y. (2005). Bootstrapping Ontology Alignment Methods with APFEL, *Proceedings of 4th International Semantic Web Conference (ISWC'05)*, Galway, Ireland.

Erman, L., Hayes-Roth, F., Lesser, V., & Reddy, D. R. (1980). The hearsay II speech understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2), 213–253.

Gal, A. & Segev, A. (2006). Putting things in context: dynamic eGovernment re-engineering using ontologies and context, *Proceedings of the 2006 WWW Workshop on E-Government: Barriers and Opportunities*.

Gal, A. (1999). Semantic interoperability in information services: Experiencing with CoopWARE. *SIGMOD Record*, 28(1), 68–75.

Gal, A., Anaby-Tavor, A., Trombetta, A., & Montesi, D. (2005). A framework for modeling and evaluating automatic semantic reconciliation. *VLDB Journal* 14(1), 50–67.

Gal, A.,Modica, G., Jamil, H.M., & Eyal, A. (2005). Automatic ontology matching using application semantics. *AI Magazine*, 26(1).

Gruber, T. R. (1993). A Translation Approach to Portable Ontologies, *Knowledge Acquisition*, 5(2), 199–220.

Guha, R. V. (1991). *Contexts: A formalization and some applications*. Doctoral dissertation, Stanford University, Stanford, CT, USA (STAN-CS-91-1399-Thesis).

Hayes-Roth, B. (1985). A blackboard architecture for control. *Artificial Intelligence*, 26, 251–321.

Heß, A., Johnston, E., & Kushmerick, N. (2004). ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services, *Proceedings International Semantic Web Conference*, 320-334.

Kahng, J., & McLeod, D. (1996). Dynamic classification ontologies for discovery in cooperative federated databases. *Proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, Brussels, Belgium (pp. 26–35). Belgium.

Kashyap, V., Dalal, S., & Behrens, C. (2001). Professional services automation: A knowledge management approach using LSI and domain specific ontologies. *Proceedings of the 14th International FLAIRS Conference (Florida AI Research Symposium), Special track on AI and Knowledge Management*.

Kelley, J. (1969). *General Topology*. American Book Company.

Kifer, M., Lausen, G., & Wu, J. (1995). Logical foundation of object-oriented and frame-based languages. *Journal of the ACM* 42.

Kim, D., Lee, S., Shim, J., Chun, J., Lee, Z., & Park, H. (2005). Practical Ontology Systems for Enterprise Application, *Proceedings of 10th Asian Computing Science Conference (ASIAN'05)*, Kunming, China.

Klusch, M., Fries, B., Khalid, M., & Sycara, K. (2005). OWLS-MX: Hybrid Semantic Web Service Retrieval, *Proceedings First International AAAI Fall Symposium Agents and the Semantic Web*.

Lesser, V., Fennell, R., Erman, L., & Reddy, D. R. (1975). Organization of the Hearsay II speech understanding system. *IEEE Transactions on Human Factors in Electronics*, ASSP-23, 11–24.

Madhavan, J., Bernstein, P. A., Domingos, P., & Halevy, A. Y. (2002). Representing and reasoning about mappings between domain models. *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, 80–86.

Madhavan, J., Bernstein, P.A., & Rahm, E. (2001). Generic schema matching with Cupid. *Proceedings of the International conference on very Large Data Bases (VLDB)*, 49–58, Rome, Italy.

Maedche, A. & Staab, S. (2001) Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16.

McCarthy, J. (1987). Generality in artificial intelligence. *Communication of ACM*, 30, 1030–1035.

McCarthy, J., & Buvac, S. (1997). *Formalizing context, computing natural language*, 13–50. Stanford: Stanford University.

McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). An environment for merging and testing large ontologies. *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*.

Melnik, S. (ed.) (2004). *Generic Model Management: Concepts and Algorithms*. Springer, Heidelberg.

Mena, E., Kashyap, V., Illarramendi, A., & Sheth, A. P. (2000). Imprecise answers in distributed environments: Estimation of information loss for multi-ontology based query processing. *International Journal of Cooperative Information Systems*, 9(4), 403–425.

Modica, G., Gal, A., & Jamil, H. M. (2001). The use of machine-generated ontologies in dynamic information seeking. *Proceedings of the Sixth International Conference on Cooperative Information Systems (CoopIS 2001)*, Trento.

Mooers, C. (1972). *Encyclopedia of Library and Information Science*. Marcel Dekker, vol. 7, ch. Descriptors, 31–45.

Motro, A., & Rakov, I. (1998). Estimating the quality of databases. *Lecture Notes in Computer Science*, 1495, 298.

Moulton, A., Madnick, S. E., & Siegel, M. (1998). Context mediation on Wall Street. *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems (CoopIS'98)*, 271–279. New York: IEEE-CS.

Mrissa, M., Ghedira, C., Benslimane, D., Maamar, Z., Rosenberg, F., & Dustdar, S. (2007). A Context-Based Mediation Approach to Compose Semantic Web Services, ACM Transactions Internet Technology, 8(1), p. 4.

Noy, F. N. & Musen, M. A. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, 450–455, Austin, TX.

Noy, N. F., & Klein, M., (2004). Ontology Evolution: Not the Same as Schema Evolution, *Knowledge and Information Systems*, 6(4), 428–440.

Oh, S. C. (2006). Effective Web-Service Composition in Diverse and Large-Scale Service Networks," *PhD dissertation, University Park*.

Oldham, N., Thomas, C., Sheth, A. P., & Verma, K. (2004). Meteor-s Web Service Annotation Framework with Machine Learning Classification, *Proceedings International Workshop Semantic Web Services and Web Process Composition (SWSWPC '04)*, 137-146.

Ouksel, A. M., & Naiman, C. F. (1994). Coordinating context building in heterogeneous information systems. *Journal of Intelligent Information Systems*, 3(2), 151–183.

Paolucci, M., Kawamura, T., Payne, T., & Sycara, K. (2002). Semantic Matching of Web Services Capabilities, *Proceedings of International Semantic Web Conference*.

Papatheodorou, C., Vassiliou, A., & Simon, B. (2002). Discovery of ontologies for learning resources using word-based clustering. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2002)*, Denver, CO (pp. 1523–1528).

Patil, A., Oundhakar, S., Sheth, A., & Verma, K. (2004). Meteor-s Web Service Annotation Framework, *Proceedings 13th International Conference World Wide Web (WWW '04)*, 553-562.

Pepper, S. (1999). Navigating Haystacks, Discovering Needles, *Markup Languages: Theory and Practice*, 1(4), MIT Press.

Platzer, C., & Dustdar, S. (2005). A Vector Space Search Engine for Web Services, *Proceeding of Third European Conference of Web Services (ECOWS '05)*.

Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). London: Butterworths.

Russell, S. & Norving, P. (2003). *Artificial Intelligence: A Modern Approach*, 2nd edition, Prentice Hall, Upper Saddle River, New Jersey.

Sacco, G. (2000). Dynamic taxonomies: A model for large information bases. *IEEE Transactions Knowledge Data Engineering*, 12(2), 468–479.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Schuyler, P. L., Hole, W. T., & Tuttle, M. S. (1993). The UMLS (Unified Medical Language System) metathesaurus: Representing different views of biomedical concepts. *Bulletin of the Medical Library Association*, 81, 217–222.

Sciore, E., Siegel, M., & Rosenthal, A. (1994). Using Semantic Values to Facilitate Interoperability among Heterogeneous Information Systems, ACM Transactions Database Systems, 19(2), 254-290.

Segev, A., & Gal, A. (2007a). Putting Things in Context: A Topological Approach to Mapping Contexts to Ontologies, *Journal of Data Semantics*, 9, 113-140.

Segev, A., & Gal, A. (2007b). Puzzling It Out: Supporting Ontology Evolution with Applications to eGovernment, *Proceedings of IJCAI Workshop on Modeling and Representation in Computational Semantics*.

Segev, A., & Gal, A. (2008). Enhancing portability with multilingual ontology-based knowledge management, *Decision Support Systems*, 45(3).

Segev, A., & Sheng, Q. Z. (2011). Bootstrapping Ontologies for Web Services, *IEEE Transactions on Services Computing*, In Print.

Segev, A., & Toch, E. (2009). Context-Based Matching and Ranking of Web Services for Composition, *IEEE Transactions on Services Computing*, 2(3), 210-222.

Segev, A., Leshno, M., & Zviran, M. (2007a). Context Recognition Using Internet as a Knowledge Base, *Journal of Intelligent Information Systems*, 29(3), 305–327.

Segev, A., Leshno, M., & Zviran, M. (2007b). Internet as a knowledge base for medical diagnostic assistance. Expert Systems with Applications, 33(1), 251-255.

Smith, H., & Poulter, K. (1999). Share the ontology in XML-based trading architectures. *Communications of the ACM*, 42(3), 110–111.

Soergel, D. (1985). *Organizing information: Principles of data base and retrieval systems*. Orlando: Academic.

Sphere Project (2004). Humanitarian Charter and Minimum Standards in Disaster Response*, The Sphere Project*, Geneva.

Spyns, P., Meersman, R., & Jarrar, M. (2002) Data modelling versus ontology engineering. *ACM SIGMOD Record*, 31(4).

Stumme, G., & Maedche, A. (2001). Ontology Merging for Federated Ontologies on the Semantic Web. *Proceedings of the International Workshop for Foundations of Models for Information Integration*. Viterbo, Italy.

Toch, E., Gal, A., & Dori, D. (2005). Automatically Grounding Semantically-Enriched Conceptual Models to Concrete Web Services, *ER*, Delcambre, L., Kop, L., Mayr, H., Mylopoulos, J., & Pastor, O. , eds., 304-319, Springer.

Turney, P. (2002). *Mining the web for lexical knowledge to improve keyphrase extraction: Learning from labeled and unlabeled data*. (Tech. Rep. No. ERB-1096; NRC #44947). Washington, DC: National Research Council, Institute for Information Technology.

Valdes-Perez, R. E., & Pereira, F. (2000). Concise, intelligible, and approximate profiling of multiple classes. *International Journal of Human Computer Studies*, 53, 411–436.

Vickery, B. C. (1966). *Faceted classification schemes*. Graduate School of Library Service, Rutgers, the State University, New Brunswick, NJ.

Vouros, G. A., Dimitrokallis, F., & Kotis, K. (2008). Look Ma, No Hands: Supporting the Semantic Discovery of Services without Ontologies, *Proceedings International Workshop Service Matchmaking and Resource Retrieval in the Semantic Web (SMRR)*.

Williams, T., Lowrance, J., Hanson, A., & Riseman, E. (1977). Model-building in the VISIONS system. *Proceedings of IJCAI-77*, Cambridge, MA (pp. 644–645).

Zhang, G., Troy, A., & Bourgoin, K. (2006). Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors, *Proceedings of 14th International Conference on Conceptual Structures (ICCS’06)*, Aalborg University, Denmark.