# DECISION SUPPORT SYSTEMS IN ONTOLOGY-BASED CONSTRUCTION OF WEB DIRECTORIES

Marko Horvat, Gordan Gledec and Nikola Bogunović
*Faculty of Electrical Engineering and Computing, University of Zagreb*
*Unska 3, HR-10000 Zagreb, Croatia*

Keywords: Ontology, Ontology Alignment, Artificial Intelligence, Semantic Web, Web directory.

Abstract: The paper proposes an ontology-based mechanism for fully automated development of a Web directory's structure using the Semantic web as an underlying and integrating principle. Maintenance of a Web directory is a time and resource wise consuming task. Moreover, there is always a realistic risk of the structure becoming unbalanced, uneven and difficult to use to all except for a few users proficient in a particular Web directory. By using ontologies to describe semantics of Web resources and Web directory's categories, and through the use of ontology mapping, it is possible to construct algorithms that can build or rearrange the structure of a Web directory. Such applications are immediately helpful but also can be useful in the more general problem of ontology sorting.

## 1 INTRODUCTION

In the last eight years we are witnessing rapid development of the Semantic web and the related spectrum of technologies. The initial paper by Tim Berners-Lee (Berners-Lee et al., 2001) introduced the notion of universally described semantics of information and services on the Web. The vision of a Web as a shared common medium for data, information and knowledge exchange, and collaboration, fostered a wealth of development and research. The Semantic web brought the power of managed expressivity provided by ontologies to the World Wide Web (WWW). Today the research in Semantic web application is not largely focused on the problem of ontologically-based Web directories. So far only a handful or papers have been published on the topic of combining ontologies and Web directories (Choi, 2001; Kavalec and Svátek, 2002; Mladenić, 1998). However, importance of Web directories and their commonplaceness makes them appealing for research.

The remainder of the paper is organized as follows; the next chapter formally defines the categories and the structure of Web directories. Web directories construction scenarios are presented in the third chapter, while the fourth chapter describes an algorithm for their ontology-based construction. Conclusion with outlook for future work is presented at the end of the paper.

## 2 CATEGORIES AND THE STRUCTURE OF WEB DIRECTORIES

A Web directory is a structured and hierarchically arranged collection of links to other web sites. Web directories are divided into categories and subcategories with a single top category, often called the root category, or just the root. Each category can have a provisional number of subcategories, with each subcategory further subsuming any number of other subcategories, and so on. Furthermore, every category has a unique name and an accompanying Uniform Resource Locator (URL), and can also carry other associated information. Each category of a Web directory contains a set of links to various sites on the WWW, and a set of links to other categories within the web directory. This basic trait is the most important feature of a Web directory.

Associations between categories are arbitrary, but there must be at least one path between any pair of categories. Disjoint sets of categories are prohibited,



Horvat M., Gledec G. and Bogunovic N. (2009).
DECISION SUPPORT SYSTEMS IN ONTOLOGY-BASED CONSTRUCTION OF WEB DIRECTORIES.
In *Proceedings of the 4th International Conference on Software and Data Technologies*, pages 281-286
DOI: 10.5220/0002243102810286

as well as parallel links and self-loops. Each category must have links to all its children, but can also have links to other categories in the Web directory which are semantically similar, or otherwise analogous to the category (cross-links, related links).

We will formally designate with ***C*** the set of all categories in a Web directory; ***R*** will be the set of all Web resources in a Web directory. One category with unique identification number $n$ is denoted $c_n$. Category has its own characteristic URL *url* and member level $l$, where $l$ is a natural number smaller than or equal to the depth of a Web directory $L$ (Figure 1). The category $c_n$ must be a member of ***C***. $C_n$ is a subset of ***C*** that belongs to the category $c_n$, and $R_n$ the subset of ***R*** with Web resources that belong to category $c_n$. We formally describe categories and structure of Web directories in Uschold, 2003.

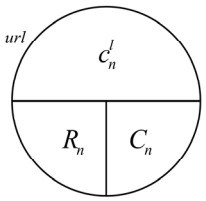


Figure 1: Schematic representation of a single category.

Mathematically speaking, Web directories are simple rooted graphs (Sedgewick, 2001). Sometimes the position of links within a category's Web page is prioritized, and in that case we are talking about ordered and rooted simple graphs. The structure of a Web category cannot be described as a tree because more than one path can connect any of its two categories: apart from paths which connect parent/child categories, they can be associated with *ad hoc* cross-links as in Figure 2.

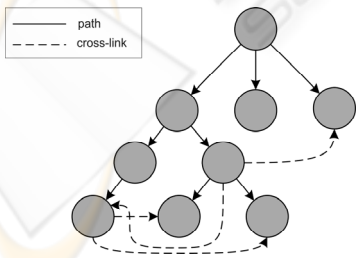


Figure 2: Realistic Web directory with possible multiple paths between two categories.

Although the categorization of a Web directory should be defined by a standard and unchanging policy this is frequently not the case. Web directories often allow site owners to directly submit their site for inclusion, even suggest an appropriate category for the site, and have editors review submissions. The editors must approve the submission and decide in which category to put the link in. However, rules that influence the editors' decision are not completely objective and are thus difficult to implement unambiguously. Sometimes a site will fall in two or even more categories, or require a new category. Defining a new category is very sensitive task because it has to adequately represent a number of sites, avoid interfering with domains of other categories, and at the same time the width and depth of the entire directory's structure has to be balanced. A Web directory with elaborate structure at one end and sparse and shallow at the other is confusing for users and difficult to find quality information in. Furthermore, after several sites have been added to a directory it may become apparent that an entirely new categorization could better represent the directory's content. In this case a part of directory's structure or even all of its levels have to be rearranged which is again time and labor consuming task.

Therefore, recognizing the challenges implied by the Web directory construction, and as well as their overall importance, the paper's authors are motivated to design and develop a decision support system – a computer-based intelligent agent – that can support decision-making in this construction process.

# 3 CONSTRUCTION SCENARIOS

The process of building Web directories has three actors:

1. Web directory system (WDS)
2. Web directory administrator (WDA)
3. Administrator of a Web site listed in the Web directory (WSA)

Ontology-based building process contains the same three actors and represents a subset of the general building process. This process includes three main tasks, or actions, that have to be performed by actors in order to construct a Web directory:

1. Semantics identification task (SIT)
2. Semantics assignment task (SAT)
3. Web directory addition task (WDAT)

*Semantics identification task* is a process that recommends which ontology class, or classes, should be instantiated and assigned to a given Web resource. *Semantics assignment task* is a process that follows semantics identification, and actually assigns a set of ontology classes to a resource. Classes that are recommended and assigned don't necessarily have to be identical. If an actor has made an error and recommended the wrong class, the actor performing assignment can overrule his recommendation. Finally, when a set of classes has been assigned to a Web resource, it has to be added to a directory. *Web directory addition task* decides exactly where in a directory's structure the new resource will be placed. This is a complicated task because it can involve creation of an entirely new category, reshuffling and updating existing categories (both horizontally and vertically within the directory's structure), or simply adding the resource to an existing category. The order of these tasks and their mutual interaction is described in the following UML activity diagram (Figure 3).

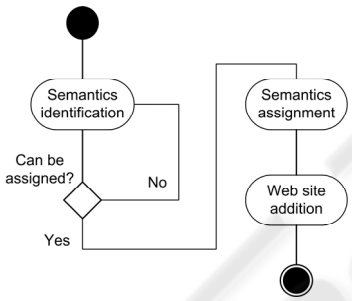


Figure 3: Main tasks in ontology-based building of Web directories.

Construction process scenarios can be divided in two groups:

1. Dominantly autonomous scenario (AUTO)
2. Man-In-The-Loop dominant scenario (MIL)

Each scenario has several possible variations or sub-scenarios. Scenarios are distinguished by the level of human participation. Sub-scenarios describe the roles of the actors involved.

Utilization of human intelligence in majority of tasks is presumed in MIL scenario, while in AUTO scenario the Web directory computer system performs more tasks than human actors. In an ideal AUTO scenario the computer executes all tasks independently. Table 1 depicts all scenarios and their variations with respective grades of positivity.

Table 1: Allocation of actors and task in ontology-based construction of a Web directory.[1]

| Tasks | Roles | | |
|---|---|---|---|
| **MIL scenario** | WSA | WDA | WDS |
| SIT | ++ | + | n/a |
| SAT | + | +++ | ++ |
| WDAT | - | +++ | +++ |
| **AUTO scenario** | WSA | WDA | WDS |
| SIT | + | + | ++ |
| SAT | - | +++ | ++ |
| WDAT | - | + | +++ |

By following the highest grades in each scenario it is possible to determine the best actor for each task. Sequences of the best choices for each task are shown in UML diagrams in Figure 4 and Figure 5. Data in the table, temporally structured in the diagrams, reflects the "Best Practice" experience gathered during 15 years of administrating the Croatian Web directory (http://www.hr/) (Gledec et al., 1999).

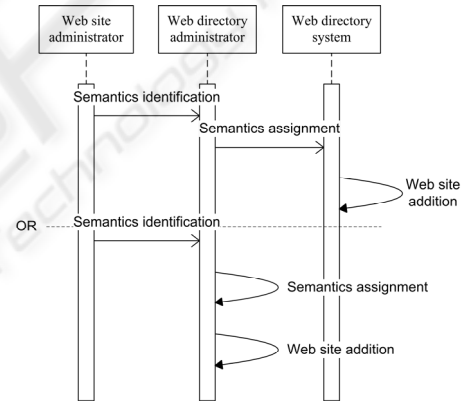


Figure 4: UML sequence diagram with the selection of best actors in the MIL scenario.

As can be seen in MIL scenario, WSA is the best actor to perform SIT, and WDA for SAT. In this scenario SIT is intentionally performed only by a human actor. WDAT can be executed equally good by WDA or WDS, but it would be wrong to leave this task to WSA. The reasoning behind allocation of actors in this scenario is that WSA is the least dependable actor and its contribution is the most likely to be subjective and erroneous. The task will be most successfully performed by WDA, but it would be inefficient and wrong to give all tasks only

[1] Sub-scenario grades: +++: the most acceptable, ++ favorable, + positive, –negative/unfavorable scenario, n/a not applicable.

to WDA. After all, one of the principal goals of the proposed system is to alleviate the burden of Web directory administration from the amenable personnel, and not to leave them with an equally difficult job. The best option is to allocate SIT to WSA and to leave the final decision about semantics to WDA who is the most knowledgeable and dependable actor of the three.

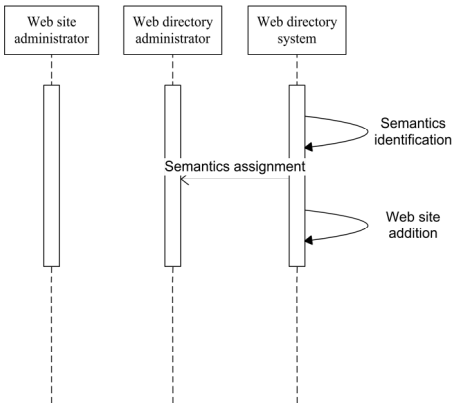


Figure 5: UML sequence diagram with the selection of best actors in the AUTO scenario.

Much the same reasoning is reflected in the AUTO scenario; however the importance of WDS in this scenario is emphasized. Thus, WDS is the optimal choice for executing SIT and WDAT. Again, WDA will perform the final assignment of ontologies to resources (i.e. SAT) to reduce possible errors to a minimum. In this scenario it was determined that it would be negative to let WSA to execute SAT and WDAT since WDA or WDS can perform a better job at this tasks. In this scenario WSA and WDA are equally suitable to execute SIT. If SAT is also given to WDS then the Web directory building system is fully automated and autonomous.

# 4 ONTOLOGICALLY BASED CONSTRUCTION

If it is possible to assign ontology to a Web resource and execute semantics identification and semantics assignment tasks as outlined in the previous chapter, it is also possible to define an ontology-based algorithm for automated construction of a Web directory structure. Such algorithm performs all tasks outlined in Figure 2. The algorithm's input are links to Web resources that are being added to the Web directory, and output is schema of the directory. Schema can be represented in a number of ways, e.g. as a markup language, or additionally the algorithm can use the schema to automatically build the directory by writing and storing necessary static and dynamic Web files like HTML, JavaScript, PHP, etc.

In order to be able to define the described algorithm we will assume that we have at a disposition function *sem* that takes a resource $r_i \in \boldsymbol{R}$ and from its semantic content builds an ontology $o_i \in \boldsymbol{O}$ where $\boldsymbol{R}$ and $\boldsymbol{O}$ are sets of all resources and ontologies, respectively.

$$sem : \boldsymbol{R} \rightarrow \boldsymbol{O} \quad (1)$$

The function *sem* builds an ontology from a resource, i.e. it performs semantics identification and semantics assignment tasks by creating a solid representation of an abstract property. This property can be described as informal and explicit on the semantic continuum scale (Uschold, 2003) and its technical realization is strictly formal. Operations of the function *sem* can be performed by a computer system or a domain expert, in which case we talk about automatic or manual ontology construction, respectively. The function *sem* is described in detail (Horvat et al., 2009).

The basis for the algorithm construction process is definition of category $c_i$ and its set of ontologies $O_i$ as a unified pair ($c_i$, $O_i$). In acquiring $O_i$ the algorithm uses the function *sem* and treats $c_i$ as a Web resource. The input is a set of Web resources $\boldsymbol{R}$ and the algorithm picks one resource $r_i$ at the time, translates in into an ontology $o_{NEW}$ and calculates distance between $o_{NEW}$ and every ontology in the Web directory $\boldsymbol{O}$ looking for the closest. Categories are compared using their member ontologies. At each moment *wd* has *n* categories and a new category has index *n+1*.

Pseudocode for ontology-based construction of Web directories

```
add (c1, ∅)
FOR each ri in R (i = 1,…,m)
create ontology sem: ri → ONEW
 IF K(C) = 1 THEN
  create category (cNEW, ONEW)
  add ri in cNEW
  add cNEW in wd as C(n+1)(1+1)
 ELSE
  find the closest category (C(n)(1), On) to
ONEW
```

```
d = dist(O_NEW, O_n)
CASE OF d
 > mindist_V:
  create category (c_NEW, O_NEW)
  add r_i in c_NEW
  add c_NEW in wd as C_{n+1}^{l+1}
 > mindist_H:
  create category (c_NEW, O_NEW)
  add r_i in c_NEW
  add c_NEW in wd as C_{n+1}^{l}
 OTHERS:
  add r_i in C_n^l
 END CASE
END CASE
END LOOP
```

The most significant aspect of the algorithm is reliance on ontologies and ontology aligning methods in order to measure similarity between ontologies and determine their mutual distance. The similarity measure $sim : \boldsymbol{C}^2 \rightarrow [0,1]$ between the two categories $c_1, c_2 \in \boldsymbol{C}$ and a distance function $dist(c_1, c_2) = 1 / sim(c_1, c_2)$ is defined elsewhere as in (Staab and Studer, 2004; Ehrig and Sure, 2004). The algorithm uses two constants in a predefined metric; *minimal horizontal semantic distance* ($mindist_H$) and *minimal vertical semantic distance* ($mindist_V$) as thresholds in the category addition process. When a new category $c_j$ is being added and category $c_i$ already exists in *wd* if $dist(c_i,c_j) > mindist_H$ then the algorithm will add $c_j$ as a new category of *wd*. Likewise, if $dist(c_i,c_j) > mindist_V$ then $c_j$ will be added in a new level of the directory *wd*, below $c_i$. If $dist(c_i,c_j) <= mindist_H$ AND $dist(c_i,c_j) <= mindist_V$ the algorithm will merge semantics of $c_j$ and $c_i$ incrementing initial ontology of $c_i$. Therefore, the thresholds are used in deciding whether it is necessary to add a new category in the directory's structure or to use an existing category. Also, the thresholds indicate where to add a new category: in the same level next to an existing category or below it.

The algorithm has two main branches. The first branch recognizes one special case when cardinal number K of all categories *C* in *wd* is 1, and the second branch processes three cases with cardinality of categories greater than 1. If $K(C) = 1$ then $l = 1$ and only the root category has been added to *wd*. In this case it is not necessary to calculate the distance between ontologies and a new category can be immediately constructed. If $K(C) > 1$ there are more categories, not just the root, and links to Web resources are assigned to the semantically closest categories. New categories are created if needed.

The single root node does not have a set of links to Web resources ($R_1 = \emptyset$) and it is assigned to an empty ontology ($c_1$, Ø), however the algorithm can be modified so it allows predefinition of main topics in a Web portal or Web directory according to the desired administrating policies.

The proposed algorithm is simple because it represents the direct and the most obvious implementation of an ontological principle in Web directory construction. Categories cannot be mutually prioritized, and the end structure is completely dependent on the order of links to Web resources which are the algorithm's input. Furthermore, there is no back-tracking or iterative optimization. For these reasons the algorithm may also be called basic or elementary, since all other ontology-based algorithms should provide better results. It could be used as an etalon for comparison of different algorithms for construction of Web directories.

Execution of this algorithm can be assigned to different roles in MIL and AUTO scenarios (see Table 1). For example, a part of the algorithm, like SAT can be given to human experts (WDS or WDA) and other tasks – SIT and especially WDAT – can be executed by an intelligent agent (WDS). Different assignment will yield diverse results and this presents an interesting topic for further study and experiments.

# 5 CONCLUSIONS

The primary goal of this paper was to envision and demonstrate a method for automatic construction and maintenance of Web directories content and structure.

We would advise caution in joining Web directories with ontologies and the Semantic web paradigms. Structures of Web directories are often biased and influenced by the contributors of resources. Administration of a large directory is an overwhelming task prone to errors. Therefore, it may be better to construct ontologies from smaller directories or from directories with rigid administrative policies. The former type of directories is more numerous than the latter, but they will also offer less information and in a more specialized area.

In the future work we would like to expand the initial system and build a hard general ontology which would efficiently encompass smaller ontologies of individual categories and provide a unitary base for ontology matching throughout the Web directory. Furthermore, we would like to test the upgraded system in real-life situations and use it regularly as a decision support expert system in maintenance of a large Web directory. In the near future we are planning to validate the system and evaluate its features by implementing it within the Croatian Web directory.

## REFERENCES

Berners Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 284, 5, 34—43 (2001)

Choi, B.: Making Sense of Search Results by Automatic Web-page Classifications. Proc. of WebNet 2001 -- World Conference on the WWW and Internet, 184—186 (2001)

Kavalec, M., Svátek, V.: Information Extraction and Ontology Learning Guided by Web Directory. In: ECAI Workshop on NLP and ML for ontology engineering, Lyon, (2002)

Mladenić, D.: Turning Yahoo into an Automatic Web Page Classifier. In: Proc. 13th European Conference on Artificial Intelligence, ECAI'98, 473—474 (1998)

Uschold, M.: Where are the Semantics in the Semantic Web? AI Magazine, 24, 3, 25—36 (2003)

Horvat, M., Gledec, G., Bogunović, N.: Assessing Semantic Quality of Web Directories Structure. Submitted for publication in ICCCI 2009, Wroclaw, Poland (2009)

Sedgewick, R.: Algorithms in C++, Third Edition. Addison-Wesley (2001)

Gledec, G.; Jurić, J.; Matijašević, M, Mikuc, M.: WWW.HR - Experiences with Web-server development and maintenance. Proc.of the 18th International Conference on Computers in Telecommunications, 22nd International Convention MIPRO'99, 93—96 (1999)

Staab, S., Studer, R.: Handbook on Ontologies. Handbooks in Information Systems, Springer (2004)

Ehrig, M., Sure, Y.: Ontology Mapping – An Integrated Approach. In: Proceedings of the First European Semantic Web Symposium, 3053, 76—91 (2004)