A Technique for Information Retrieval from Microformatted Websites*

J. Guadalupe Ramos[1], Josep Silva[2], Gustavo Arroyo[2], and Juan C. Solorio[1]

[2] DSIC, Universidad Politécnica de Valencia
Camino de Vera s/n, E-46022 Valencia, Spain.
{jsilva,garroyo}@dsic.upv.es
[1] Instituto Tecnológico de La Piedad
Av. Tecnológico 2000, La Piedad, Mich., México. CP 59300
{guadalupe@dsic.upv.es,juancsol@hotmail.com}

Abstract. In this work, we introduce a new method for information extraction from the semantic web. The fundamental idea is to model the semantic information contained in the microformats of a set of web pages, by using a data structure called *semantic network*. Then, we introduce a novel technique for information extraction from semantic networks. In particular, the technique allows us to extract a portion—a *slice*—of the semantic network with respect to some criterion of interest. The slice obtained represents relevant information retrieved from the semantic network and thus from the semantic web. Our approach can be used to design novel tools for information retrieval and presentation, and for information filtering that was distributed along the semantic web.

1 Introduction

The Semantic Web is considered an evolving extension of the World Wide Web in which the semantics of information and services on the web is made explicit by adding metadata. Metadata provides the web contents with descriptions, meaning and inter-relations. The Semantic Web is envisioned as a universal medium for data, information, and knowledge exchange.

Recently, a new initiative has emerged that looks for attaching semantic data to web pages by using simple extensions of the standard tags currently used for web formatting in (X)HTML[1], these extensions are called *microformats* [1, 2]. A microformat is basically an open standard formatting code that specifies a set of attribute descriptors to be used with a set of typical tags.

Example 1. Consider the XHTML of the left that introduces information of a common personal card:

* This work has been partially supported by the Spanish *Ministerio de Ciencia e Innovación* under grant TIN2008-06622-C03-02, by the *Generalitat Valenciana* under grant ACOMP/2009/017, by the *Universidad Politécnica de Valencia* (Programs PAID-05-08 and PAID-06-08) and by the Mexican *Dirección General de Educación Superior Tecnológica* (Programs *CICT 2008* and *CICT 2009*).

[1] XHTML is a sound selection because it enforces a well-structured format.

```
<h2>Directory</h2>
<p> Vicente Ramos <br>
    Software Development <br>
    118, Atmosphere St. <br>
    La Piedad, México <br>
    59300 <br>
    +52 352 52 68499 <br>
</p>
<h4>His Company</h4>
<a href="page2.html">
   Company Page </a>
```

```
<h2>Directory</h2>
<div class="vcard">
  <span class="fn">Vicente Ramos</span>
  <div class="org">Software Development </div>
  <div class="adr">
    <div class="street-address">Atmosphere 118</div>
    <span class="locality">La Piedad, México</span>
    <span class="postal-code">59300</span>
  </div>
  <div class="tel">+52 352 52 68499</div>
    <h4>His Company</h4>
  <a class="url" href="page2.html">Company Page </a>
</div>
```

Now, observe the code on the right which shows the same information but using the standard `hCard` microformat [3], which is useful for representing data about people, companies, organizations, and places. The `class` property qualifies each type of attribute which is defined by the `hCard` microformat. The code starts with the required main class `vcard` and classifies the information with a set of classes which are auto-explicative: `fn` describes name information, `adr` defines address details and so on.

In this paper we propose the use of *semantic networks* which is a convenient simple model for representing semantic data; and we define a slicing technique for this formalism in order to analyze and filter the semantic web.

2 From the semantic web to the semantic network

The concept of *semantic network* is fairly old, and it is a common structure for knowledge representation, which is useful in modern problems of artificial intelligence. A semantic network is a directed graph consisting of nodes which represent *concepts* and edges which represent *semantic relations* between the concepts [4, 5].

In order to represent semantic information in a semantic network we consider the microformats, i.e., classes as convenient entities for modeling, and then, for indexing or referencing. If we focus on the relations between classes we identify two kinds of relations, namely[2]:

strong relations that are the relations which come from hypertext links between pages or sections of a page by using anchors.

weak relations that can be *embedding relationships*, for classes that embeds other classes or *semantic relationships* among classes of the same type, for instance, between two `vcard`.

Example 2. Consider the semantic network depicted in Figure 1 (the grey parts of the figure do not belong to the semantic network and thus they can be ignored for the time being). It is composed of two webpages ($P1$ and $P2$), and $P1$ represents the microformatted code of Example 1.

[2] In this paper, without loss of generality, we only consider weak relations (i.e., only semantic relations), thus we analyze semantic networks without taking into account the labels associated to the edges.

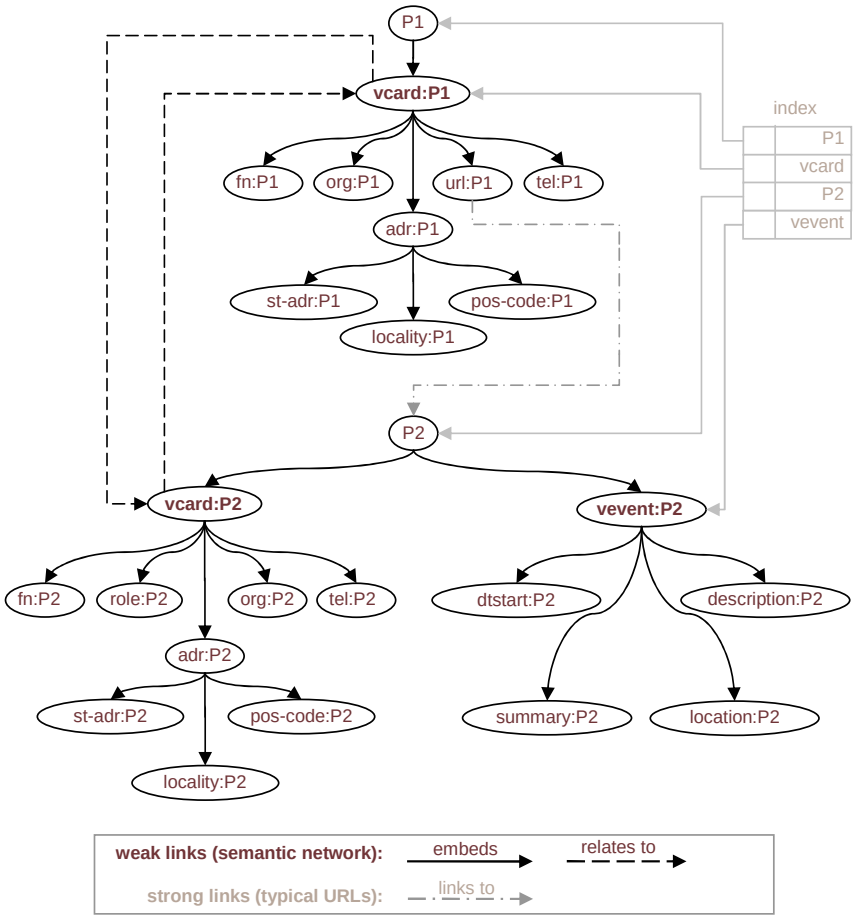


Fig. 1. Example of semantic network.

In the figure, the nodes of the first page are labeled with $P1$ and the nodes of the second page are labeled with $P2$. Thus, nodes (i.e., concepts) are unique. We observe three kinds of edges: The `locality` class from Example 1 is embedded in the `adr` class. Thus, there is an embedding relationship from node `adr` to node `locality`. Furthermore, `vcard` in $P1$ and `vcard` in $P2$ are linked by a semantic relationship. Besides, there is one strong hyperlink to $P2$ generated by the microformatted tag `<a class="url" href="page2.html">`. Observe that the graph only contains semantic information and their relations; and it omits content or formatting information such as the ** labels. Observe that we add to the graph two additional concepts, $P1$ and $P2$, which refer to web pages. This is very useful in practice in order to make explicit the embedding relation between microformats and their web page containers.

3 A technique for information retrieval

We introduce first some preliminary definitions.

Definition 1 (semantic network). *A directed graph is an ordered pair* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *where* $\mathcal{V}$ *is a finite set of vertices or nodes, and* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ *is a set of ordered pairs* $(v \rightarrow v')$ *with* $v, v' \in \mathcal{V}$ *called edges. A semantic network is a directed graph* $\mathcal{S} = (\mathcal{V}, \mathcal{E})$ *in which nodes have been labeled with names of web pages and microformatting classes of these pages.*

As an example of semantic network consider the directed graph in Figure 1 (omitting the grey parts) where nodes are the set of microformatted classes provided by two semantic web pages.

A semantic network is a profuse mesh of information. For this reason, we extend the semantic network with an *index* which acts as an interface between the semantic network and the potential interacting systems. The index contains the subset of concepts that are relevant (or also visible) from outside the semantic net. It is possible to define more than one index for different systems and or applications. Each element of the index contains a key concept and a pointer to its associated node. Artificial concepts such as webpages (See $P1$ and $P2$ in Figure 1) can also be indexed. This is very useful in practice because it is common to retrieve the embedded (microformatted) classes of each semantic web page.

Let $\mathcal{K}$ be a set of concepts represented in the semantic network $\mathcal{S} = (\mathcal{V}, \mathcal{E})$. Then, $rnode : (\mathcal{S}, k) \rightarrow \mathcal{V}$ where $k \in \mathcal{K}$ (for the sake of clarity, in the following we will refer to k as the *key concept*) is a mapping from concepts to nodes; i.e., given a semantic network $\mathcal{S}$ and a key concept k, then $rnode(\mathcal{S}, k)$ returns the node $v \in \mathcal{V}$ associated to k.

Definition 2 (semantic index). *Given a semantic network* $\mathcal{S} = (\mathcal{V}, \mathcal{E})$ *and an alphabet of concepts* $\mathcal{K}$*, a semantic index* $\mathcal{I}$ *for* $\mathcal{S}$ *and* $\mathcal{K}$ *is any set* $\mathcal{I} = \{(k, p) \mid k \in \mathcal{K}$ *and* p *is a mapping from* k *to* $rnode(\mathcal{S}, k)\}$

We can now extend semantic networks by properly including a semantic index. We call this kind of semantic network *indexed semantic network* (IS).

Definition 3 (indexed semantic network). *An indexed semantic network* IS *is a triple* $IS = (\mathcal{V}, \mathcal{E}, \mathcal{I})$*, such that* $\mathcal{I}$ *is a semantic index for the semantic network* $\mathcal{S} = (\mathcal{V}, \mathcal{E})$.

Now, each semantic index allows us to visit the semantic network from a well defined collection of entrance points which are provided by the *rnode* function.

Example 3. An IS with a set of nodes $\mathcal{V} = \{a, b, c, d, e, f, g\}$ is shown in Figure 2 (a). For the time being the reader can ignore the use of colors black and grey and consider the graph as a whole. There is a semantic index with two key concepts a and c pointing out to their respective nodes in the semantic network.

Similarly, the semantic network of Figure 1 has been converted to an IS by defining the index with four entries $P1$ (page1.html), $P2$ (page2.html), *vcard* and *vevent* and by removing the strong links. Thus, for instance, *vcard* entry points to the cycle of *vcard* nodes.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and two nodes $v_1, v_n \in \mathcal{V}$, if there is a sequence $v_1, v_2, \ldots, v_n$ of nodes in $\mathcal{G}$ where $(v_i, v_{i+1}) \in \mathcal{E}$ for $1 \leq i \leq n-1$, then we say that there is a *path* from v_1 to v_n in $\mathcal{G}$. Given $u, v \in \mathcal{V}$ we say that the node v is *reachable* from u if there is a path from u to v.

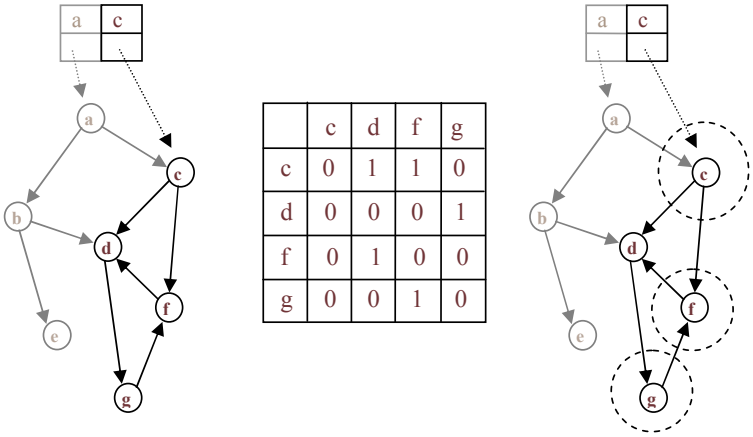
Definition 4 (semantic sub-net). *Let $IS = (\mathcal{V}, \mathcal{E}, \mathcal{I})$ be an indexed semantic network. Then, a semantic sub-net of IS with respect to concept k, with $(k, p) \in \mathcal{I}$ for some p, is $\mathcal{S}_k = (\mathcal{V}', \mathcal{E}')$ such that $\mathcal{V}' = \{rnode((\mathcal{V}, \mathcal{E}), k)\} \cup \{v | v \in \mathcal{V}$ and v is reachable from $rnode((\mathcal{V}, \mathcal{E}), k)\}$ and $\mathcal{E}' = \{(u, v) | (u, v) \in \mathcal{E}$ and $u \in \mathcal{V}'\}$.*

Example 4. Figure 2 (a) shows in black color the semantic sub-net extracted from the whole IS with respect to concept c.

Definition 5 (semantic relationship). *Given a semantic network $\mathcal{S} = (\mathcal{V}, \mathcal{E})$ and a node $v \in \mathcal{V}$, the semantic relationships of v are the edges $\{v \rightarrow v' \in \mathcal{E}\}$. We say that a concept v is semantically related to a concept u if there exists a semantic relationship $(u \rightarrow v)$.*

The semantic relations in our semantic networks are unidirectional. The semantics associated to the edges of a semantic network is not transitive because edges can have different meanings. Therefore, the semantic relation of Definition 5 is neither transitive.

Given a node n in a semantic network, we often use the term *semantically reachable* to denote the set of nodes which are reachable from n through semantic relationships. Clearly, semantic reachability is a transitive relation.



	c	d	f	g
c	0	1	1	0
d	0	0	0	1
f	0	1	0	0
g	0	0	1	0

Fig. 2. a) A semantic sub-net. b) The sub-net's adjacency matrix. c) A backward slice.

3.1 Semantic sub-net slicing

In this section we present a procedure that allows us to extract a portion of a semantic sub-net according to some criterion. The procedure uses an *adjacency matrix* to represent the semantic sub-net.

The adjacency matrix m of a directed graph $\mathcal{G}$ with n nodes is the $n \times n$ matrix where the non-diagonal entry m_{ij} contains 1 if there is an edge such that $m_i \rightarrow m_j$.[3]

Example 5. Consider the semantic sub-net in Figure 2 (a). Node c has two directed edges, one to node d and other to node f. Thus, in the entry m_{cd} and m_{cf} we write 1, and 0 in the other cells.

Now, we are in a position to introduce our slicing based method for information recovering from semantic sub-nets. Firstly, we can select a concept in the index. From this concept we can extract a semantic sub-net as described before. Next, in the resultant semantic sub-net we can select the node of interest. Hence, our slicing criterion consists of a pair formed by a key concept and a node. Formally:

Definition 6 (slicing criterion). *Let $IS = (\mathcal{V}, \mathcal{E}, \mathcal{I})$ be an indexed semantic network. Then a slicing criterion $\mathcal{C}$ for IS is a pair of elements $\langle k, v \rangle$ such that $(k, p) \in \mathcal{I}$ for some p, $v \in \mathcal{V}'$ and $S_k = (\mathcal{V}', \mathcal{E}')$ is the semantic sub-net of IS with respect to concept k.*

Given a semantic sub-net, we can produce two different slices by traversing the sub-net either forwards or backwards from the node pointed out by the slicing criterion. Each slice gives rise to different semantic information.

Example 6. Consider the slicing criterion $\langle c, d \rangle$ for the IS in Figure 2 c). The first level of slicing uses c to extract the semantic sub-net highlighted with black color. Then, the second level of slicing performs a traversal of the semantic sub-net either forwards or backwards from d. In Figure 2 c) the backward slice contains all nodes whereas the forward slice would only contain {d, f, g}.

Example 7. Consider the semantic network in Figure 1 together with the slicing criterion $\langle P1, adr : P1 \rangle$. With $P1$ we can perform the first level of slicing to recover a semantic sub-net which is composed by the nodes $\{P1, vcard : P1, vcard : P2\}$ and all of their descendant (semantically reachable) nodes. Then, from node $adr : P1$ we can go forwards and collect the information related to the address or backwards and collect nodes $vcard : P1$, $P1$ and $vcard : P2$. The backward slicing illustrates that the node $adr : P1$ is semantically reachable from $P1$, $vcard : P1$, and $vcard : P2$, and thus, there are semantic relationships between them. Hence, we extract a slice from the semantic network and, as a consequence, from the semantic web.

We can now formalize the notion of forward/backward slice for semantic sub-nets. In the definition we use $\rightarrow^*$ to denote the reflexive transitive closure of $\rightarrow$.

Definition 7 (forward/backward slice). *Let $IS = (\mathcal{V}, \mathcal{E}, \mathcal{I})$ be an indexed semantic network with $(k, p) \in \mathcal{I}$ for some p. Let $\mathcal{S}_k = (\mathcal{V}', \mathcal{E}')$ be the semantic sub-net of IS with respect to k and $\mathcal{C} = \langle k, node \rangle$ a slicing criterion for IS. Then a slice of IS is $\mathcal{S}' = (\mathcal{V}_1, \mathcal{E}_1)$ such that*

forward $\mathcal{V}_1 = \{node\} \cup \{v | v \in \mathcal{V}' \text{ and } (node \rightarrow^* v) \in \mathcal{E}'\}$

[3] Note that we could write a label associated to the edge in the matrix instead of 1 in order to also consider other relationships between nodes.

Input: An indexed semantic network $IS = (\mathcal{V}, \mathcal{E}, \mathcal{I})$
and a slicing criterion $\mathcal{C} = \langle k, node \rangle$ where $(k, p) \in \mathcal{I}$ for some p
Output: A slice $\mathcal{S}' = (\mathcal{V}', \mathcal{E}')$
Initialization: $\mathcal{V}' := \{node\}, \mathcal{E}' := \{\}, Visited := \{\}$
Begin
Compute $S_k = (\mathcal{V}_k, \mathcal{E}_k)$ a semantic sub-net of IS
whose adjacency matrix is $\mathcal{M}$
Repeat
let $s \in (\mathcal{V}' \setminus Visited)$
let $c := column(s, \mathcal{M})$
For each $s' \in \mathcal{V}_k$ with $r = row(s', \mathcal{M})$ and $\mathcal{M}_{r,c} = 1$
$\mathcal{V}' := \mathcal{V}' \cup \{s'\}$
$\mathcal{E}' := \mathcal{E}' \cup \{(s' \rightarrow s)\}$
$Visited := Visited \cup \{s\}$
Until $\mathcal{V}' = Visited$
End
Return: $(\mathcal{V}', \mathcal{E}')$

Fig. 3. An algorithm for semantic network backward slicing.

backward $\mathcal{V}_1 = \{node\} \cup \{v | v \in \mathcal{V}' \text{ and } (v \rightarrow^* node) \in \mathcal{E}'\}$

and $\mathcal{E}_1 = \{(u \rightarrow v) \mid (u \rightarrow v) \in \mathcal{E}' \text{ with } u, v \in \mathcal{V}_1\}$

The algorithm of Figure 3 shows the complete slicing based method for information extraction from semantic networks. Roughly speaking, given an IS and a slicing criterion, (i) it extracts the associated semantic sub-net, (ii) it computes the sub-net's adjacency matrix, and (iii) it extracts (guided by the adjacency matrix) the nodes and edges that form the final slice.

The algorithm uses two functions $row(s, \mathcal{M})$ and $column(s, \mathcal{M})$ which respectively return the number of row and column of concept s in matrix $\mathcal{M}$. It proceeds as follows: Firstly, the semantic sub-net associated to IS and the adjacency matrix of the sub-net are computed. Then, the matrix is traversed to compute the slice by exploiting the fact that a cell $\mathcal{M}_{i,j}$ with value 1 in the matrix means that the concept in column j is semantically related to the concept in row i. Therefore, edges are traversed backwards by taking a concept in a column and collecting all concepts of the rows that have a 1 in that column.

4 Related work and conclusions

In [6], three prototype hypertext systems were designed and implemented. In the first prototype, an unstructured semantic net is exploited and an authoring tool is provided. The prototype uses a knowledge-based traversal algorithm to facilitate document reorganization. This kind of traversing algorithms is based on typical solutions like depth-first search and breadth-first search. In contrast, our IS allows us to optimize the task of information retrieval.

[7] designed a particular form of a graph to represent questions and answers. These graphs are built according to the question and answer requirements. This is in some way related to our work if we assume that our questions are the slicing criteria and our answers are the computed slices. In our approach, we conserve a general form of semantic network, which is enriched by the index, so, it still permits to represent sub-graphs of knowledge.

To the best of our knowledge this is the first program slicing based approach to extract information from the semantic web. The obtained answers are semantically correct, because since, the information extraction method follows the paths of the source semantic tree, i.e., the original semantic relationships are preserved. Furthermore, semantic relationships contained in sets of microformatted web pages can also be discovered and extracted.

Program slicing has been previously applied to data structures. For instance, Silva [8] used program slicing for information extraction from individual XML documents. He also used a graph-like data structure to represent the documents. However semantic networks are a much more general structure, that could contain many subgraphs, while XML documents are always a tree-like structure. In contrast to this method, our approach can process groups of web pages.

This method could be exploited by tools that feed microformats. Frequently, these tools take all the microformats in the semantic web and store them in their databases in order to perform queries. Our representation improves this behavior by allowing the system to determine what microformats are relevant and what microformats can be discarded. Another potential use is related to automatic information retrieval from websites by summarizing semantic content related to a slicing criterion. Similarly, web search engines could use this method to be able to establish semantic relations between unrelated links.

References

1. Microformats.org. The Official Microformats Site. `http://microformats.org/`, 2009.
2. R. Khare and T. Çelik. Microformats: a Pragmatic Path to the Semantic Web. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 865–866. ACM, 2006.
3. hCard. Simple, Open, Distributed Format for Representing People, Companies, Organizations, and Places. `http://microformats.org/wiki/hcard`, 2009.
4. J. F. Sowa, editor. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann, 1991.
5. J. F. Sowa. Semantic Networks. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*. John Wiley & Sons, 1992.
6. W. Wang and R. Rada. Structured Hypertext with Domain Semantics. *ACM Transactions on Information Systems (TOIS)*, 16(4):372–412, 1998.
7. D. Mollá. Learning of Graph-based Question Answering Rules. In *Proc. HLT/NAACL 2006 Workshop on Graph Algorithms for Natural Language Processing*, pages 37–44, 2006.
8. J. Silva. A Program Slicing Based Method to Filter XML/DTD Documents. In *SOFSEM (1)*, pages 771–782, 2007.