# Instantiation of relations for semantic annotation

Sylvain Tenier, Yannick Toussaint, Amedeo Napoli, Xavier Polanco

# Instantiation of relations for semantic annotation

S.Tenier[1,2], Y.Toussaint[1], A.Napoli[1], X.Polanco[2]
[1] Laboratoire Lorrain de Recherche en Informatique et ses Applications
BP 239, 54506 Vandoeuvre-lès-Nancy Cedex, France
{tenier, yannick, napoli}@loria.fr
[2] Institut de l'Information Scientifique et Technique
54514 Vandoeuvre-lès-Nancy, France

## Abstract

*This paper presents a methodology for the semantic annotation of web pages with individuals of a domain ontology. While most semantic annotation systems can recognize knowledge units, they usually do not establish explicit relations between them. The method presented identifies the individuals which should be related among the whole set of individuals and codes them as role instances within an OWL ontology. This is done by using a correspondance between the tree structure of a web page and the semantics of the information it contains.*

## 1. Semantic annotation of web pages

This paper presents a method to formalize the content of web pages. This method is presented through the test case of research team pages. Most teams have a web site describing their activities including team members, projects and themes. The aim is to be able to answer questions like "who are the researchers working in this particular thematics". Results from research work in Information Retrieval allow for improved search engines to return a relevant set of pages [13, 15, 4]. However, an expert is required to perform the analysis of the information presented by the actual web pages. This analysis requires to define precisely what is meant by "researcher": is it any person working for a research team, or someone who has research themes and projects? Then a search for all the pieces of information of the page allowing a person to fit the definition is needed.

In order to automatize the analysis, the knowledge units lying in a web page must be formalized within a knowledge representation (KR) language. This is the purpose of the Semantic Web [3], in which web pages can be processed by software agents while remaining readable by domain expert. For this, a formal ontology is provided that represents the concepts (and the relations between the concepts) of the domain considered. The ontology is then populated by a semantic annotation process : given a research team page, the knowledge units it contains are discovered and the ontology is populated with individuals. Each individual is an instance of a concept and can be related to others by an instance of role. The analysis of annotated web pages is then done by querying the ontology. The result of the query consists in a set of individuals satisfying the constraints associated with the query.

However, most semantic annotation systems can recognize concept instances in a web page but they do not establish relations between them [20]. We claim that individuals recognition as well as relation instantiation are both needed for an efficient semantic annotation. Accordingly, this paper deals with the problem of relation recognition (and then role instantiation) within web pages. The method is based on the following assumption : there is a correspondance between the structure of a web page and its semantics. In our approach, classification mechanisms on the domain and range of roles defined in the ontology are exploited to relate existing individuals.

The paper is organized as follows: the next section introduces the correspondance between structure and semantics. Section 3 presents the ontology defined to formalize the domain of research and to guide the annotation process. Section 4 shows how the role instantiation method builds on previous works on semantic annotation. Section 5 develops the inductive role instantiation process. Section 6 evaluates the method on two sets of web pages. Conclusions and perspectives are presented in section 7.

## 2. Semantics of a page structure

When a user loads a web page in a web browser, a collection of units of information is displayed, each of them consisting of a string of characters (image interpretation is out of the scope of this paper). This information is displayed with a certain format to the user. Let's consider the page

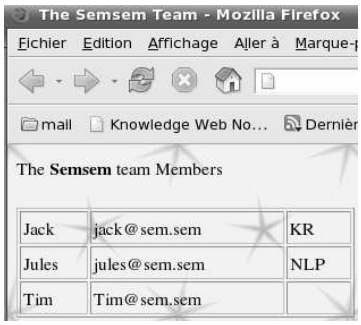


**Figure 1. Web browser display of a page**

```
<html><head><title>Th
e Semsem Team</title>
</head><body>The<b>Se
msem</b>team Members<
table><tr><td>Jack</t
d><td>jack@sem.sem</t
d><td>KR</td></tr><tr
><td>Jules</td><td>ju
les@sem.sem</td><td>N
LP</td></tr><tr><td>T
im</td><td>tim@sem.se
m</td><td></td></tr><
/table></body></html>
```

**Figure 2.** HTML **source code**

presented on Figure 1 as an exemple of a simplified version of a research team web page. This page presents the *Semsem* team which is composed of three members, all of them having an email address and two of them having a research theme. The relations between each piece of information are easily infered by a human thanks to the structuration of the information in a table. For instance, `Jack` is naturally associated to his email `jack@sem.sem` and his theme `KR`. While this understanding process is natural for a human, it actually requires a lot of background knowledge:

**Requirement 1** (identifying concepts)**.** The first need is to associate a concept to strings of the web page. In the example Figure 1, `Jack` is a person, `jack@sem.sem` an email, `KR` a research theme.

**Requirement 2** (interpreting structures)**.** In the case of a table, several interpretations are possible. In the current example, the correct interpretation is that persons are presented in the first column, email addresses in the second and themes in the third. Moreover, there exists a relation between the information presented in the same line. In the example, `Jack` , `jack@sem.sem` and `KR` are related in the same way as `Jules` , `jules@sem.sem` and `NLP`.

**Requirement 3** (relations between concepts)**.** For relations between `Jack`, `jack@sem.sem` and `KR` to be recognized, it is necessary to know which relations are relevant. For instance, we should know that a person can have an email address and a research theme, but that it makes no sense to establish a relation between a theme and an address.

In order for a machine to infer the relations between information units in a web page, such background knowledge must be made available. The first requirement is dealt with in semantic annotation systems as the task of identifying the character strings of a web page that correspond to the concepts of a domain ontology [20]. This identification process results in individuals of an ontology. However, current systems generally do not identify relations between the generated individuals. The automatic identification of those relations needs to satisfy requirements 2 (structure interpretation) and 3 (relation relevance). In a web page, the structure is provided by the HTML implementation of the page. HTML code for Figure 1 is presented Figure 2. This code contains tags that define the presentation and gives a structure to the information (the strings between the tags). Examples of tags used for structuring a web page are *table* or *ul/li*, which present the information respectively as a table or as a list. Solving requirement 2 can be seen as being able to assign a semantics to those tags. Finally, requirement 3 implies the formalization of the relations between the concepts as roles in the ontology. The next section presents the formalization of the concepts and roles of the research domain in an ontology.

## 3. The $\mathcal{O}$ ontology: a formalization of concepts and roles of the research domain

The $\mathcal{O}$ ontology formalizes the research domain and is designed to guide the annotation process. The $\mathcal{O}$ ontology is coded within the Web Ontology Language (OWL), which is the standard language for KR in the Semantic Web. OWL is based on Description Logics, which provides a good tradeoff between efficient reasonning support and sufficient expressive power [1]. $\mathcal{O}$ has been designed using the ontology editor SWOOP [12] which supports consistency checking thanks to its integration of Pellet [18], a JAVA OWL-specific reasoner.


**Figure 3.** $\mathcal{H}$ **hierarchy**

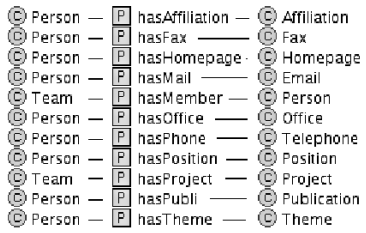


**Figure 4. Roles**

The $\mathcal{O}$ ontology is composed of the following elements:

- primitive concepts, organized in a hierarchy $\mathcal{H}$. Figure 3 shows some concepts from the research domain. Figure 5 displays an excerpt of the OWL code. For the sake of clarity, the hierachy has been simplified, each concept being subsumed by `owl : Thing`. This is acceptable since no subsumption relation between concepts is used in the role instantiation process. A concept can be seen as a class to be instanciated with knowledge units identified in web pages.

```
<owl:Class rdf:about="#Person"/>
<owl:Class rdf:about="#Position"/>
<owl:Class rdf:about="#Project"/>
<owl:Class rdf:about="#Publication"/>
<owl:Class rdf:about="#Team"/>
```

**Figure 5. OWL concept definition**

```
<owl:ObjectProperty rdf:about="#hasMail">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Email"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasTheme">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Theme"/>
</owl:ObjectProperty>
```

**Figure 6. OWL role definition**

- a set of roles. Each role is defined by its domain and range which are concepts of $\mathcal{H}$. Let $C, D \in \mathcal{H}$, then $R(C, D)$ is a role whose domain is $C$ and whose range is $D$. Only one role can be defined between two concepts. Figure 4 displays a set of roles with their domain and range. For instance, `Person` is domain of the `hasEmail` and `hasTheme` roles. Figure 6 shows an excerpt of the OWL code. Each role represents a sufficient but not necessary condition for a relation to exist between two concepts : a person whose telephone number is not given in the web page is still a person.

- a set of individuals $\mathcal{I}$. In OWL, individuals are identified uniquely by a string. Each individual is defined as an instance of concept of $\mathcal{H}$ and can be related to other individuals by role instances. Each role instance defines the relation between an individual and another individual. For example, `Jack` is an individual which is instance of the concept of `Person`. `jack@sem.sem` is an instance of the concept of the `Email` concept. The role instance `hasEmail(Jack, jack@sem.sem)` relates `Jack` to `jack@sem.sem`.

The annotation process of a web page with individuals related by roles of the $\mathcal{O}$ ontology is a based on two steps. In the first step, individuals are generated. This is done by annotating a web page with concepts of the $\mathcal{O}$ ontology. Then, relations between individuals are discovered and role instances are added. Figure 7 presents the resulting OWL encoding of individuals and role instances. The next section

```
<research:Team rdf:about="#html/body/">
  <research:hasMember rdf:resource="#html/body/table/tr[0]/td[0]"/>
  <research:hasMember rdf:resource="#html/body/table/tr[1]/td[0]"/>
  <research:hasMember rdf:resource="#html/body/table/tr[2]/td[0]"/>
  <rdf:type rdf:resource="&owl;Thing"/>
</research:Team>
<research:Person rdf:about="#html/body/table/tr[0]">
  <research:hasMail rdf:resource="#html/body/table/tr[0]/td[1]"/>
  <research:hasTheme rdf:resource="#html/body/table/tr[0]/td[2]"/>
  <rdf:type rdf:resource="&owl;Thing"/>
</research:Person>
```

**Figure 7. Individuals related with roles : the Semsem `Team` and the jack `Person`**

presents the initial individual generation. It is followed by the description of the role instantiation process.

## 4. Initial individuals generation

In this section, the generation of individuals of the $\mathcal{O}$ ontology is presented. The generation process consists in identifying information items in the web page and a-ssociating units to a concept. Several systems have been designed to automatize the task of annotating a web page with concept instances. These systems can be classified in three categories, depending on the level of interaction with the user. Some of these systems are presented below. For a more detailed state of the art on those systems, the reader is refered to [20].

1. **Supervised systems:** these interactive systems are based on interfaces that display simultaneously the ontology and the document to be annotated, and let a user mark different instances of concepts of the ontology. The system generates a file containing the annotations. Amaya [17] and Mangrove [14] are tools for web pages annotation allowing the user to annotate a page in the web browser.

2. **Semi-automatic systems:** these systems are still user-centered but automatize certain tasks. S-CREAM [10] integrates the Information Extraction (IE) module Amilcare [5]: user-generated annotations are taken as input of a machine-learning algorithm. The output suggests annotations when annotating another document. Lixto [9] and SHOE [11] provide wrappers that recognize and annotate regular patterns in a page. COHSE [2] and Magpie [7] match strings in a document with a list of terms associated with concepts of the ontology.

3. **Unsupervised annotation:** the aim of unsupervised systems is to let the user out of the annotating loop, after a bootstrapping step. These systems crawl the Web and exploit the redundancy of information to validate annotations. This approach is constrained by the large amount of information it requires. Amardillo [16] associates IE techniques with a statistical Information Integration method that confirms the validity of the mined information. KnowItAll [8] exploits a specific mesure based on the answers provided by different search engines.

Semi-automatic and unsupervised systems are intended to be less time consuming than supervised systems. However, they require expertise which domain experts might not have and can lead to incorrect annotations. We have therefore designed an interactive system, implemented as an extension of the Firefox web browser. The web page is dis-

played along with the concepts of the ontology. Then a domain expert associates each piece of information with one of the concepts. Each generated individual is named using a unique identifier in order to record its location in the web page for the role instantiation process.

## 5. The role instantiation process

Firstly, the DOM tree representation of a web page is introduced. An example consisting of a web page annotated with individuals related by role instances follows. A heuristic is then proposed for the identification of *minimal subtrees* where individuals can be related. Finally, the association of role instances to individuals in each minimal subtree is presented.

### 5.1. Representation of a web page as a tree

Requirement 2 from section 2 introduces an interpretation of a web page based on its structure. The structure of a page is represented as a tree on which it is possible to set constraints. Such a tree representation is defined by the Document Object Model (DOM) recommandation[1]. Figure 8 displays the DOM tree of the Semsem team web page, generated from the HTML code presented on Figure 2. The DOM tree consists in nodes and arcs where each arc represents a parent/child relation between two nodes. Childless nodes are named *leaves*, as opposed to *internal nodes*. Each internal node is the root of a *subtree*. The tree is ordered, meaning that the order of child nodes is relevant. A label is associated to each node: subtrees are labelled with the name of the HTML tags and leaves by the character strings between the HTML tags. Each node is unambiguously identified by a *point and range location* as defined by the XPointer recommandation[2].

### 5.2. Annotation on a tree representation

The input of the role instantiation process consists of the DOM tree where each relevant leaf has been annotated by a concept instance using the initial concept recognition process presented in section 4. For the sake of simplicity, it is considered that each annotated unit of information corresponds to exactly one leaf of the DOM tree. Cases where a leaf contains several information units are studied further in the evaluation section. Each individual is located using the XPointer notation as the parent node of the recognized leaf.

The output of the whole process is the page annotated by individuals related by role instances. Figure 7 shows an

[1] http://www.w3.org/TR/DOM-Level-2-Core/

[2] http://www.w3.org/TR/xptr-xpointer/#datatypes

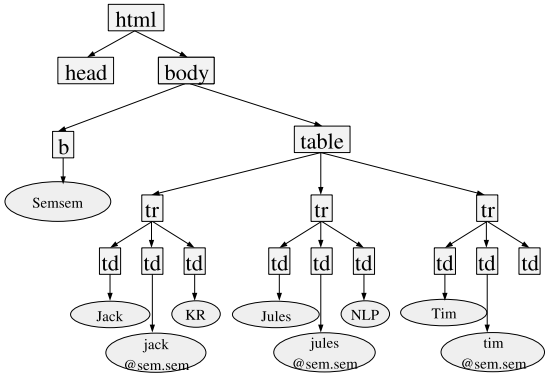


**Figure 8. DOM tree of the web page**

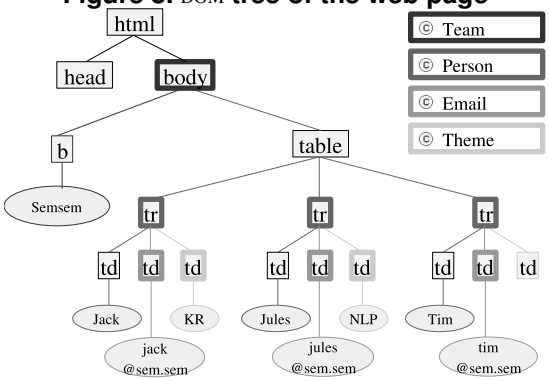


**Figure 9. Tree view of the final annotation**

excerpt of the annotation for the example page. The `body` node is annotated by an individual, instance of the `Team` concept. This individual is related to individuals associated to each `tr` by `hasMember` role instances. Let's detail the first `tr` subtree, annotated by an individual $\mathtt{i}_1$, instance of a `Person` concept. $\mathtt{i}_1$ is related to an individual $\mathtt{i}_2$ associated to the second `td` of the subtree by a `hasEmail` role instance. It is also related to an individual $\mathtt{i}_3$ associated to the third `td` by a `hasTheme` role instance. The annotated DOM tree for the example page is presented on Figure 9. It shows that certain nodes are associated to individuals. Each node is root of a subtree which is used to infer the relations between the individuals it contains. The next section explains the identification process of such subtrees.

### 5.3. Minimal subtree identification

Possible relations between concepts are defined by roles of the ontology. This provides a necessary condition to establish a relation between two individuals : a role must be defined in the $\mathcal{O}$ ontology between the concepts the individuals are instance of. However, this condition is not sufficient. It would be an error to instanciate a role relating all the individuals that the role recognizes. On the web page Figure 1, Jack and Jules are

persons, KR and NLP are themes and the $\mathcal{O}$ ontology defines the `hasTheme(Person, Theme)` role. This role can be correctly instanciated as `hasTheme(Jack, KR)` and `hasTheme(Jules, NLP)`. However, `hasTheme(Jack, NLP)` is incorrect. It is therefore necessary to define a heuristic to limit role instantiation to relevant individuals. This heuristic is based on the following hypothesis:

**Hypothesis** (Boundary to role instantiation)**.** The instantiation of roles between individuals is constrained by the location of the individuals in a tree representation of the web page.

The heuristic identifies sets of individuals to which the role instantiation process must be applied. Each set of individuals is located inside a subtree. For instance, the first `tr` on Figure 8 is the root node of the subtree grouping Jack with his email address and his theme. The heuristic uses a bottom-up approach on the DOM tree, starting from leaves and identifying tree structures that group at least two individuals that are instances of different concepts. Recursively, it identifies tree structures that group at least two tree structures, or one tree structure and an individual from different concepts. Such structures are called minimal subtrees:

**Definition 1** (Minimal Subtree)**.** A subtree is minimal iff it is the smallest subtree that includes two or more constituants. A constituant is either a minimal subtree or an individual. At least two of these constituants must be instance of a different concept.

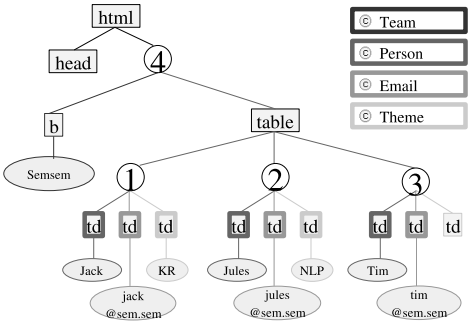


**Figure 10. Minimal subtrees identification**

Figure 10 shows the result of the minimal subtrees identification process. Numbers associated to the root of the subtrees reflect the order of their identification. Each `tr` is identified since the `td` it groups are annotated by a different concept. `body` is identified as the minimal subtree that groups the `b` and `tr` constituants. On the other hand, `table` is not a minimal subtree. As Figure 9 shows, the analysis of each `tr` results in their annotation with an individual instance of the same, `Person` concept. In this case, the requirement of at least two constituants that are instance of a different concept is therefore not validated.

This requirement of at least two different concepts in the subtree implies that the annotation of each minimal subtree must be performed before the identification process is resumed, according to the Structural Proximity Principle:

**Definition 2** (Structural Proximity Principle)**.** The structural proximity principle restricts the search for roles between concepts. Role instances can be computed only between constituants of the same minimal subtree.

### 5.4. Semantic analysis of minimal subtrees

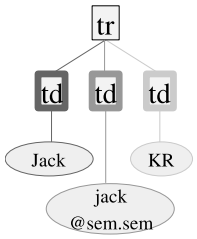


**Figure 11. Minimal subtree $\mathcal{S}$ with three individuals**

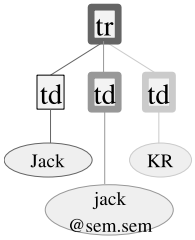


**Figure 12. Annotation with a `Person` instance**

The semantic analysis of a subtree instanciates roles between individuals of the subtree and associates one of the individuals to the root node of the subtree. The analysis of the minimal subtree $\mathcal{S}$ is presented on Figure 11, corresponding to the first minimal subtree identified on Figure 10, is given below. This subtree groups Jack, instance of `Person`, jack@sem.sem, instance of `Email`, and KR, instance of `Theme`.

Firstly, the role instantiation is performed by a lookup into the ontology. For each pair of individuals $(I_C, I_D) \in \mathcal{S}$ that are respectively instances of concepts $C, D \in \mathcal{O}$ with $C \neq D$, the ontology is checked for the existence of $R_{(C,D)}$. If there exists $R(C, D)$, then $R(I_C, I_D)$ is instanciated. In our example, $\texttt{hasEmail}(Jack, jack@sem.sem)$ and $\texttt{hasTheme}(Jack, KR)$ are instanciated. This lookup within the $\mathcal{O}$ ontology is implemented in an extended version of the KnowledgeServer (KS) of the Kasimir [6] project. The KS provides reasoning facilities based on the OWL reasonner Pellet [18].

Once roles are instanciated, the root node of the subtree is associated to one of the individuals. Such an individual must be domain of at least a role identified in the previous step and range of none of them. Formally, $I_C$ annotates the subtree iff

- $\exists I_D \in \mathcal{S}$ such as $\exists R(I_C, I_D)$
- $\forall I_D \in \mathcal{S}, \nexists R_{(I_D, I_C)}$.

The result is the annotation of $\mathcal{S}$ with $I_D$ and the generated role instances. Figure 12 shows the association of the individual Jack to the $\mathcal{S}$ subtree. The OWL encoding is presented Figure 7. The fully annotated tree is shown on Figure 9.

## 6. Evaluation

The aim of this section is to validate, with real world data, the role instantiation methodology. This evaluation is based on web pages describing research activities. Two corpus consisting of web pages presenting members of a research team have been gathered. The first corpus is composed of 22 web pages retrieved from web sites of teams that participate to the Knowledge Web Network of Excellence[3], a project funded by the European Commission $6^{th}$ Framework Programme. The second corpus is made of 23 web pages presenting the members of each team of a french laboratory[4].

The fist evaluation holds on the role instantiation process in minimal subtrees. Since the most related concept is `Person`, this concept is used as the basis of this evaluation. It is checked for each web page whether individuals instances of `Person` are located in at least the same subtree as one of the individuals it should be related to. This is the case for 20 pages out of 22 in the first corpus and 21 out of 23 in the second.

The second evaluation is performed on pages which pass the first evaluation. The percentage of roles that are instanciated is calculated. 107 roles should be instanciated in the first corpus, 111 in the second. The results are presented on Figure 13. The *Subtree* column shows the results when applying the minimal subtree identification and analysis method on the web pages. The other columns present improvements that could be provided by different optimizations. A full explanation is proposed below :

| Role instances | Total | Subtree | Hyperlink | Context | Multi | Failed |
|---|---|---|---|---|---|---|
| Corpus 1 | 111 | 66 | 21 | 2 | 14 | 8 |
| Corpus 1 (%) | 100 | 59,46 | 18,92 | 1,8 | 12,61 | 7,21 |
| Corpus 2 | 107 | 68 | 1 | 16 | 19 | 3 |
| Corpus 2 (%) | 100 | 63,55 | 0,93 | 14,95 | 17,76 | 2,8 |

**Figure 13. Statistics of the structure of information in both web pages corpus**

1. *Subtree*: how many relations are identified by the minimal subtree identification and analysis method? The direct application of the proposed method gives a correct role instantiation rate of 60% of instanciated roles in the first corpus and of 63% in the second.

2. *Link*: how does following links improve the role instantiation rate? This first optimization deals with the case when the information is presented in several pages. In such cases, the main page presents a list of persons. For each person, a link is provided leading to a subpage presenting the related individuals. This situation is dealt with by replacing the node featuring the link by the actual DOM tree of the target page. The minimal subtree identification and analysis are then applied on the expanded DOM tree. This allows to instanciate an additional 19% of the roles in the first corpus.

3. *Context*: are linebreakers a good way to improve results? linebreakers are HTML tags like <br> or <hr> that cause individuals which are related to be located in the previous or the next subtree. To deal with this case, the concerned individuals should be considered by the minimal subtree analysis algorithm as part of the same subtree. This allows a additional 15% of the roles to be instanciated in the second corpus. However, proper identification of such cases is a work in progress and is not currently applied.

4. *Multi*: what must be done when several individuals are located in the same leaf of the DOM tree? On both corpus, this case occurs when individuals are separated by syntactic elements, such as commas, instead of HTML tags. The solution considered is to rebuild the DOM tree so that individuals are in separate leaves. Around 15% of roles instances from both corpus could be taken into account that way.

5. *Failed*: how many role instances cannot be processed? The generation of some role instances is only manageable manually. This case occurs for 7% of the total individuals in the first corpus, 3% in the second.

In conclusion, the proposed method based on minimal subtree identification and analysis is able to generate more than 60% of role instances of the gathered web pages. Implementation of all the planned optimizations would allow the automatic generation of most of the remaining role instances.

## 7. Summary and perspectives

In this paper, we have proposed a role instantiation method for semantic annotation of web pages. This role instantiation method is based on the fundamental hypothesis of the existence of a correspondance between the structure of a web page and its semantics. This hypothesis is justified by the evaluation. The annotation process takes as input a web page and uses an ontology as a resource. The web page is annotated by individuals that are instances of

[3] http://knowledgeweb.semanticweb.org/
[4] http://www.loria.fr/research/equipes/

concepts of the ontology. Then the tree representation of the web page is analysed in order to determine the minimal subtrees in which individuals can be related. For each minimal subtree, a semantic analysis is performed to instanciate roles between individuals. This instantiation depends on the domain and range of roles of the ontology. A role is associated with an individual if the concept of the individual is the domain of a role whose range is instanciated by another individual of the subtree. Finally, the minimal subtree is annotated by an individual. The evaluation shows that the process is able to instanciate 60% of the role instances. Optimizations would allow for an additional 35% more role instances.

A short-term perspective is the integration of all the optimizations. For example, the *Context* optimization requires being able to detect individuals that must be associated to an existing minimal subtree. A further perspective is to deal with an incomplete initial concept annotation. Up to now, we have considered that the DOM tree is fully annotated by concept instances. The aim would be to deal with incompletete annotation. This would allow for the integration of a semi-automatic concept annotation system. Given a web page, several subtrees are annotated with instances of the same concept. The idea is to use approaches like [4, 19] in which structural similarities of XML documents based on Description Logics have been studied. Once a description is generated for a given concept, similar subtrees can be annotated automatically. In the example on Figure 9, the `tr` subtrees are annoted by the `Person` concept by using the semantic analysis. By exploiting the similarity of these subtrees, the system could automatically annotate similar subtrees, like each `tr` as a `Person` with its `hasEmail` and `hasTheme` role instances.

## References


[1] G. Antoniou and F. van Harmelen. Web ontology language: Owl. In *Handbook on Ontologies in Information Systems*, pages 67–92, 2003.

[2] S. Bechhofer, C. Goble, L. Carr, and S. Kampa. COHSE: Semantic Web gives a Better Deal for the Whole Web? In *ISWC International Semantic Web Conference Poster*, Sardinia, June 2002.

[3] T. Berners-Lee. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco, 1999.

[4] D. Calvanese, G. D. Giacomo, and M. Lenzerini. Representing and reasoning on xml documents: A description logic approach. *J. Log. Comput.*, 9(3):295–318, 1999.

[5] F. Ciravegna, A. Dingli, Y. Wilks, and D. Petrelli. Amilcare: adaptive information extraction for document annotation. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 367–368, New York, NY, USA, 2002. ACM Press.

[6] M. d'Aquin, C. Bouthier, S. Brachais, J. Lieber, and A. Napoli. Knowledge editing and maintenance tools for a semantic portal in oncology. *Int. J. Hum.-Comput. Stud.*, 62(5):619–638, 2005.

[7] J. Domingue, M. Dzbor, and E. Motta. Semantic layering with magpie. In *Handbook on Ontologies*, pages 533–554. Springer, 2004.

[8] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall: (preliminary results). In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 100–110, New York, NY, USA, 2004. ACM Press.

[9] G. Gottlob, C. Koch, R. Baumgartner, M. Herzog, and S. Flesca. The lixto data extraction project: back and forth between theory and practice. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–12, New York, NY, USA, 2004. ACM Press.

[10] S. Handschuh, S. Staab, and F. Ciravegna. S-cream-semi-automatic creation of metadata. *Proc. of the European Conference on Knowledge Acquisition and Management*, 2002.

[11] J. Heflin, J. A. Hendler, and S. Luke. Shoe: A blueprint for the semantic web. In *Spinning the Semantic Web*, pages 29–63, 2003.

[12] A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca-Grau, and J. Hendler. Swoop - a web ontology editing browser. *Journal of Web Semantics*, 4(1), 2005.

[13] D. Konopnicki and O. Shmueli. Database-inspired search. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 2–12. VLDB Endowment, 2005.

[14] L. McDowell, O. Etzioni, S. D. Gribble, A. Y. Halevy, H. M. Levy, W. Pentney, D. Verma, and S. Vlasseva. Mangrove: Enticing ordinary people onto the semantic web via instant gratification. In *International Semantic Web Conference*, pages 754–770, 2003.

[15] A. O. Mendelzon, G. A. Mihaila, and T. Milo. Querying the world wide web. In *DIS '96: Proceedings of the fourth international conference on on Parallel and distributed information systems*, pages 80–91, Washington, DC, USA, 1996. IEEE Computer Society.

[16] B. Norton, S. Chapman, and F. Ciravegna. Orchestration of semantic web services for large-scale document annotation. In *ESWC*, pages 649–663, 2005.

[17] V. Quint and I. Vatton. An introduction to amaya. *World Wide Web J.*, 2(2):39–46, 1997.

[18] E. Sirin and B. Parsia. Pellet: An owl dl reasoner. In *Description Logics*, 2004.

[19] D. Toman and G. Weddell. On reasoning about structural equality in xml: a description logic approach. *Theor. Comput. Sci.*, 336(1):181–203, 2005.

[20] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, (4):14–28, 2006.