Verifying the World Wide Web: a Position Statement

Marie-Christine Rousset
L.R.I. U.R.A C.N.R.S
University of Paris-Sud, France
mcr@lri.lri.fr

Abstract

This paper investigates the issues raised by checking potential update anomalies of a web site. It is of course impossible to control the updates for the whole World Wide Web. However, it is conceivable to provide tools for pointing out potential missing updates for a given web site (i.e, a set of Web pages, for instance related to a given matter or to a given institution, which is under the control of a web site manager). The assumption that we make in this paper is that *modeling the semantics of the considered Web site* is possible, while *modeling the semantics of the sources outside of it*, which can be connected to it, is not possible. Therefore, two different update problems have to be distinguished, depending on whether the updates that have to be done inside the web site are triggered by changes occuring inside or outside the web site.

1 Introduction

The World Wide Web(WWW) is a tremendous source of information which can be viewed as a large, heterogeneous and distributed database. The nature of the WWW raises several difficult and new issues for accessing pertinent information, querying different sources, and building and maintaining web sites. Several recent works have addressed problems for improving the search of pertinent information and in particular query interfaces to multiple sources(e.g, [LRO96, CGMH$^+$94]).

The WWW has the potential to give a large access to a large body of knowledge, mainly in the form of documents (e.g web pages written in html). One important problem is that documents have a syntactic structure which is usually loose and, more importantly, which does not necessarily reflect the actual semantics of their content. In particular, the markers used to structure html documents provide a very poor semantics for the content of the documents, though providing some indication for it.

Another important aspect of the nature of the WWW is that information sources are created, added, deleted and updated in a manner which is not controlled or centralized. So far, all this is done more or less manually by people responsible of their own web pages, or by web site managers who control and centralize a whole web site of an institution or a company, but who cannot control how it interacts with outside their web site. As a result, updating is not done in a rigourous or reliable way: a lot of information that is out of date remain accessible, and new information that is expected to be accessible is not. For example, it is frequent to find obsolete information about people and projects related to an institution or a research laboratory. The reason is that people forget to delete information that they put on the web. On the other hand, it is frequent to face with missing information because people forgot to update their pages. For instance, it is frequent not to find the most recent publications of a person on his/her web page.

This paper investigates the problem of checking potential update anomalies of sources. It is of course impossible to control the updates for the whole web. However, it is conceivable to provide tools for pointing out potential missing updates for a given a web site. By a web site, we mean a set of web pages, for instance related to a given matter or to a given institution, which is under the control of a web manager. The assumption that we make is that *modeling the semantics of the considered web site* is possible, while *modeling the semantics of the sources outside of it* is not possible. Therefore, two different update problems have to be distinguished, depending on whether the updates that have to be done inside the web site are triggered by changes occuring inside or outside the web site.

In the first case, the updating problem is similar to the usual update problem in databases: some global semantic constraints can be expressed about the data, which have to remain satisfied at each update. For instance, it can be stated and checked that all the Web pages of the members of a laboratory must have a link to the Web page of the laboratory's director. Languages for expressing integrity constraints have to be designed to fit with web site modeling.

In the second case, the problem is different because we have no control over what happens outside the web site. It is just possible to resort to existing search tools to go and search outside the potential relevant information and sources. Expressing and maintaining global constraints overlapping data inside and outside the web site is not possible. However, it might be the case that an update to an existing source ouside the web site might have to trigger or to suggest an update to be done inside the web site. For instance, the update of a web page containing the reference list of a subject (e.g, Description Logics) might suggest to update the content of the item "recent publications" of the web page of a person inside the web site under consideration.

In the sequel of the paper, we describe the model of the syntactic structure of a web site that we consider, and how we articulate it with a semantic model for describing its schema in a flexible way. We then elaborate on how to deal with the two distinct update problems, concerning the updates that have to be done inside the web site and outside the web site respectively.

2 Syntactic Structure and Semantics of a Web Site

It is important to clearly distinguish the syntactic structure and the semantic content of (a set of) information sources that are available through the network.

On one hand, many information sources are web pages obeying a certain syntactic structure reflecting a certain grammar. In particular, a document written in a HTML format can be seen as a tree-like structure where nodes represent the grammar symbols and edges reflect the hierarchical structure of the document.

On the other hand, the web pages have a semantic content, which interests the users, but whose a very small part shows through markers and associated strings.

Many attempts have been done to model the semantics of a set of sources. The point is to find a good compromise between considering a minimalist semantic model, which is simple to build (and which may even be automatically built from the existing sources) but limited, and building a sophisticated model with a lot of potentiality but too complicated to acquire. In particular, in the context of WWW, it is not conceivable to model the semantics of all the sources. The only reasonable option is to clearly distinguish the sources that can be modeled (e.g, those which compose a given web site), from those which cannot be modeled. In addition, building manually a sophiticated model of even a small subpart of the web (e.g a web site) is a difficult task. Some authors [DGL$^+$96] propose to use methods of automatic knowledge acquisition for facilitating this task. Some simple graph-based models (e.g OEM [YPW95], or labeled trees [ACM97]) have been proposed in database community to deal with the integration or the transformation of heteregenous data.

It is important to note that the semantic models that have to be built over existing web sources must be simple and flexible. In addition, they have to be appropriate to the needs, which are not the same whether the aim is querying existing sources or helping building a site for instance. One major difficulty is how to articulate those semantic models with the syntactic structure of the existing sources that are stored.

Our approach is to start from a minimalist model of the structure and the content of a web site, based on the labeled trees model proposed by [ACM97], and to enrich it by adding a logical layer over it, which is grounded on the semantics available in the labels and the links of the trees.

2.1 The labeled tree model

As a starting point, we use the *labeled tree* representation proposed in [ACM97] to model different formats for semi-structured data (in particular, html documents).

A web page is represented as a labeled tree. A web site is represented as a forest of labeled trees. A labeled tree is a tree with a labeling of the vertexes. Distinguishing the labels from the vertexes is the key to clearly distinguish the syntactic structure from the semantic content which is associated to it.

Following [ACM97], we assume the existence of some infinite sets: (i) *name* of names, (ii) *vertex* of vertexes, (iii) *dom* of data values. The internal vertexes of the trees have labels from *name* which express the type or the class of the complex objects represented by the subtrees rooted by the vertexes. The leaves have labels from $dom \cup vertex$, which represent the values associated to the leaves. Such a value can be a string representing a data value, or a vertex. The only constraint is that if a vertex occurs as a leaf label, it must also occur as a vertex in the forest. It is the way to express that there exists a link between two web pages (see for instance the &2 node in figure 1). With each root of a labeled tree is associated an

URL address.

To illustrate things, we show in figure 1 an example of a model of three web pages contained for instance in the web site of a given university. The first one corresponds to the web page of a research laboratory, whose the name, the head, the parent organizations, the groups and the publications are represented as labels which give an indication for the semantic content of the considered web page. The second one describes the content of the web page of a person related to the university: he/she has a name, a position, an affiliation, teaching activities and research activities. The third one describes a publication record, each publication refering to an author, a title, a conference and a year.

We distinguish two kinds of objects. A *complex object* corresponds to a subtree: for each internal vertex v, the maximal subtree of root v is called the object v. A *basic object* corresponds to a leaf labeled with a string: in that case, we identify the label, the leaf and the basic object.

2.2 The additional logical layer

The idea is to represent the semantic information that is available about the domain through the labels of the trees by *base predicates*. In particular, we consider as many unary base predicates as names used in the labels. Therefore, the extension of each predicate p is made of the objects corresponding to the subtrees rooted by a node labeled with p. For instance, in our example illustrated in figure 1, the extension of the unary predicate $Group$ is made of the objects that are the trees rooted &14 and &15 respectively. In the same way, the extension of the predicate $Name$ is made of the string objects "lri", "iasi" and "rousset".

In addition, we consider two binary base predicates that represent semantic links between objects, which are reflected by the syntactic structure of a web site. We make the assumption that a link between two labeled trees expresses a semantic relation between the two corresponding objects (recall that one of these objects has a leaf labeled with the root of the labeled tree representing the other object). This semantic connection is represented by the binary base predicate *ConnectedTo*. We distinguish another kind of semantic relation between objects that are in the same labeled tree. We denote *PartOf* the binary predicate expressing that an object appears as a subtree of another object. For instance, the fact $ConnectedTo(\&1, \&2)$ expresses the fact that in our web site the object &1 (representing a laboratory) is related to the object &2 (representing a person who is the head of one of the groups of the laboratory). The fact $PartOf(\&251, \&2)$ expresses the fact that the object &251 (representing a reference record) is a sub-component of the object &2 (representing a person).

Finally, we need to relate objects that represent the same real world entity. To state such a correspondence we use the binary predicate $Same$. For instance, in figure1, the object &2512 might represent the same publication as the object &31. It will be represented by the fact $Same(\&2512, \&31)$. It is a semantic information. That does not mean necessarily that they have exactly the same label and the same structure. In this paper, to simplify the discussion, we suppose that this binary predicate $Same$ is a base predicate. Note however that [ACM97] considers a similar predicate (noted is), which is defined by correspondence rules (involving correspondence literals and tree terms).

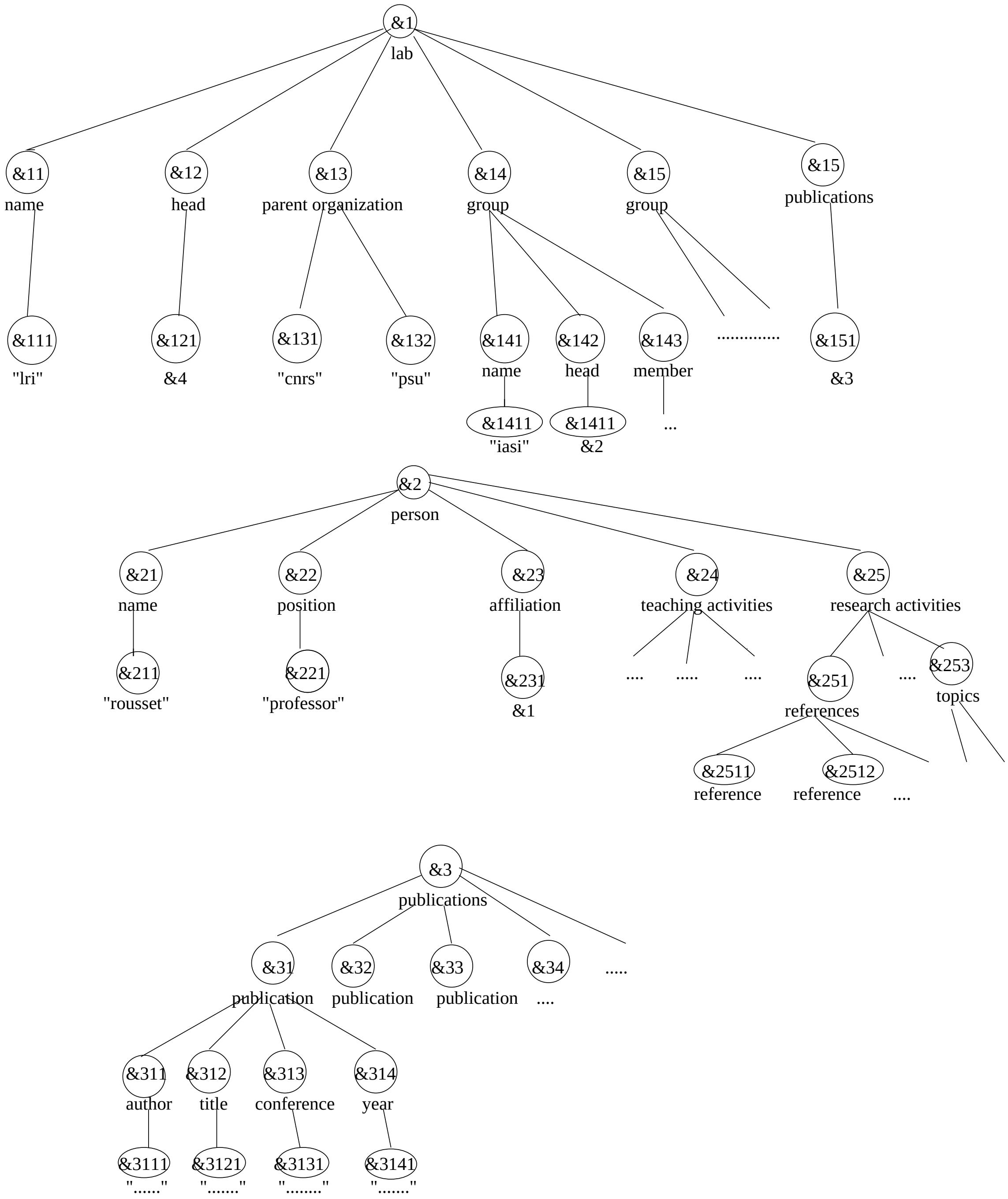

Figure 1: An example of labeled trees

The point is that the extension of the (unary and binary) base predicates that we consider can be easily and automatically extracted from labeled trees. They represent the minimal semantic model that can be automatically acquired from existing web pages in a web site.

Clearly, more sophisticated semantic models are often needed. In that case, we advocate for defining them in a flexible way, which enables an easy and clear articulation with the syntactic structure supporting the minimal semantic model. For doing so, we propose to define additional predicates from the existing base predicates, which enrich the semantics of the domain of interest. The declaration of those additional predicates can be seen as defining *virtual logical views* over the base predicates whose the extension is stored in the web site.

Example 1: In our example, it can be useful to define the notion of *researcher* from the basic and existing notion of *person*. A researcher is a person whose the web page mentions an item about research activities. Also, we may want to semantically relate a person to a lab by defining the predicate $IsMember$. A person is a member of a lab if the web page describing the lab contains an object that is connected to the web page of that person, refering it as a member (of a group in a lab) or as a head (of the lab or of a group in the lab). In the same way, we can relate a reference to his author by defining a binary predicate $IsAuthor$, which connects objects that are persons to objects that are parts of publications objects.

It can be expressed through the following logical rules, where the literals appearing in the antecedent are base predicates.

$$
\begin{array}{l}
Person(X), ResearchActivities(Y), PartOf(Y, X) \Rightarrow Researcher(X) \\
Person(X), Lab(Y), Head(Z), PartOf(Z, Y), ConnectedTo(Z, X) \Rightarrow IsMember(X, Y) \\
Person(X), Lab(Y), Member(Z), PartOf(Z, Y), ConnectedTo(Z, X) \Rightarrow IsMember(X, Y) \\
Person(X), Publication(Y), PartOf(Z, Y), Author(Z), Name(Z), PartOf(Z, X) \\
\quad \Rightarrow IsAuthor(X, Y) \\
Person(X), Publication(Y), Same(Y, U), IsAuthor(X, Y) \Rightarrow IsAuthor(X, U)
\end{array}
$$

The point is that the view predicates can be easily and declaratively defined by a web site manager who knows what he/she needs to express on the domain related to the web site he/she controls, depending on his/her purposes. Once this view layer has been defined, it can be used for querying or for reasoning on the web site.

In particular, for the task of maintaining a web site, expressing some global (integrity or dependency) constraints over the content and the structure of the web pages in the concerned web site can be needed. Constraints can be easily specified by logical formulas over view as well as base predicates.

In the next section, we first elaborate on expressing and dealing with constraints to maintain a web site. We then show some directions to control the effects on the given web site of updates that comes from outside. The problem there is that the web site manager has no control over the web pages outside his/her web site and no specific knowledge over their content.

3 Two Different Update Problems Raised by WWW

Expressing and checking global constraints that have to be satisfied by data is a classical database problem. In the setting of WWW, it requires to have a certain knowledge and a certain control on the data that are stored in web pages. According to our assumption, it can only be done at the level of a web site. The first issue is to choose a language to express the constraints that are needed. A second issue, which is of course related to the first one, is to check that the constraints which have been specified remain satisfied at each update of the data. Those issues are investigated in section 3.1

However, controling the updates that might be necessary because of possible updates that occurred outside the web site raises new issues that were not considered so far in database. We investigate them in section 3.2.

3.1 Expressing and Dealing with Constraints Over a Web Site

First of all, it is important to note that the database verification problem is quite different from the knowledge base verification problem, even if in both settings, similar constraints are considered. In the database context, the difficulty is to check that a huge amount of data satisfy constraints. The point is to reduce the data that have to be matched againt the constraints: at each update, only the data that might be concerned by the update have to be considered and checked w.r.t the constraints. In contrast, in the knowledge base setting, data are not stored and the verification problem concerns the formulas (usually rules) in the knowledge base. The point is to check that the data which can be logically entailed from those formulas satisfy some output constraints if they satisfy some input contraints. In[LR96], the connection between the knowledge base verification problem and the query containment problem has been established.

In this section, the verification problem that we consider is a database verification problem: the data are the information stored and structured in web pages composing a given web site.

In that setting, different kinds of constraints have to be considered. Constraints about the content of web pages have to be distinguished from constraints about the structure of web pages. For example, saying that publications contained in the web pages of members of a lab have to be found too in the web page about the publications of the lab is a constraint about the content of web pages. In contrast, saying that a lab web page has to be structured according to a given list of items (e.g, name, head, parent organizations, groups, publications), and sub-items (a group is structured in terms of a name, a head, and members) is a constraint on its syntactic organization.

Another kind of constraints which is needed concerns the links connecting web pages in a web site. For instance, it can be stated as a constraint that every web page of a member of a laboratory must have a link to the web page of the laboratory's head.

In this paper, we restrict ourselves to expressing and dealing with semantic constraints, i.e, constraints that deal with the content of web pages or the existing of links between web pages. Recall that the content of web pages is reflected by the unary base predicates and the binary base predicate *PartOf*, and the links between the pages is reflected by the binary base predicate *ConnectedTo*.

3.1.1 Expressing constraints

The constraints that we consider are dependencies between conjunctive queries. More precisely, a constraint is a formula of the form:

$$\forall\bar{X}[\exists\bar{Y}Q_1(\bar{X},\bar{Y}) \Rightarrow \exists\bar{Z}Q_2(\bar{X},\bar{Z})]$$

where $Q_1(\bar{X},\bar{Y})$ and $Q_2(\bar{X},\bar{Z})$ are conjunctions of base or view literals. When there is no ambiguity, we omit the variables quantifications.

Note that such formulas can just express dependencies between positive updates (i.e, if a fact is added another fact may have to be added).

Example 2: The first following constraint C_1 expresses that a person who is member of a laboratory must have in his/her web page an item describing his/her research activities. The second constraint C_2 specifies that there must exist a link between a web page of a member of a laboratory and the web page of the head of this laboratory. Finally, the third constraint C_3 says that the publications associated with a laboratory must have authors that are members of the laboratory.

$$\begin{aligned}
&C_1 : IsMember(X,Y), Lab(Y) \Rightarrow \exists Z[ResearchActivities(Z), PartOf(Z,X)]\\
&C_2 : IsMember(X,Y), Lab(Y), Head(Z), PartOf(Z,Y) \Rightarrow ConnectedTo(X,Z)\\
&C_3 : Lab(X), Publication(Z), ConnectedTo(X,Z)\\
&\qquad \Rightarrow \exists V[IsMember(V,X), IsAuthor(V,U), Same(Z,U)]
\end{aligned}$$

3.1.2 Checking constraints

In theory, checking that a constraint $\forall\bar{X}[Q_1(\bar{X},\bar{Y}) \Rightarrow Q_2(\bar{X},\bar{Z})]$ is satisfied against a set of data can be done in a two-steps querying algorithm:
(1) building from the data the answers (i.e, tuples $(\bar{a},\bar{b})$) of the conjunctive query $Q_1(\bar{X},\bar{Y})$ which is the antecedent of the constraint,
(2) then building from the data the answers of each conjunctive query $Q_2(\bar{a},\bar{Z})$.

In practice, according to the possible huge amount of stored data, it is crucial to be guided by updates in order to focus on the data and on the constraints which have a chance to be affected by the data that are updated.

In our setting, an update on the data concerns base predicates. Let B the base predicate which is involved in the concerned update.

A simple way to focus on the constraints that might be concerned by an update is to take advantage of the dependency graph underlying the definition of views predicates and the antecedents of the constraints. Given a set of rules defining a set of view predicates, we can define a dependency graph, whose nodes are the predicates appearing in the rules. There is an arc from the node of predicate Q to the node of predicate P if Q appears in the antecedent of a rule whose consequent predicate is P.

Let C a constraint, if B appears in the antecedent of C, or if there is a path in the dependency graph from B to one of the view predicates appearing in the antecedent of C, then the constraint C is kept and the conjunctive query which is antecedent will be evaluated, otherwise, it is discarded from the evaluation.

For instance, in our example, if the list of publications of the LRI laboratory is updated, only the last constraint C_3 has to be evaluated.

A simple way to focus on the data that are updated is to instantiate the literals of predicate B with the tuple $\bar{a}$ corresponding to the update $B(\bar{a})$, and to propagate this instantiation in the whole antecedents of the constraints that are concerned.

3.2 Controlling Updates Coming from Outside the Web Site

The problem here is that we have no control on the web pages outside the web site, and no way of modeling them. Consequently, we cannot consider semantic constraints as before. We can just use existing search tools that are available on the network (e.g Altavista) in order to go and search outside our web site some documents which might contain recent information that might be relevant to certain sources inside our web site.

The main drawback of the current existing search tools is that they allow very poor and only syntactic queries (mostly key words). As a result, if the query is a too precise key word, the search can fail and no document is returned. On the other hand, if the query is a too general and vague key word, several thousands documents might be returned, which match with the query (i.e, which contains the specified key word). Therefore, it is crucial to find ways to focus the search.

The approach that we suggest is to use *production rules* (also called *active rules* in the database context) as evocation rules whose activation is controlled by the web site, and whose action is to trigger a search query using for instance Altavista. Focussing the search of pertinent information can be done in two ways: first, by constraining the triggering conditions of the evocation rules, second by making the queries of the action part as precise as possible.

For doing so, we consider productions rules whose the condition part is a conjunctive query over base and view predicates modeling the web site, and the action is a call to any existing search engine, associated with a syntactic query. Note however that all the existing search engines do not offer the same services. It is important to call search engine supporting sophisticated syntactic queries allowing "or" and "and" key words combinations together with the specification of the last date of updating of the documents and possibly partial URL address specification.

Such production rules are related to web pages (or some of the objects that they contain) and their evaluation is triggered at given periods of time (e.g, every six months for controlling updates of publication record objects).

Example 3: Let suppose that we want to specify evocation rules associated with the possible updates to be done to the reference record of a researcher. Recall that the notion of being a researcher has been defined as a view predicate over the base predicate person. One possible search that can be triggered using the tool Altavista is to search documents containing the string "conference program", the string corresponding to research topics of the researcher, and the string corresponding to the name of the researcher. In addition, we can be interested only by documents whose the last update is in 1997. Finally, we can restrict the search to URL's containing "aaai.org". It is specified by the following production rule.

If	$Researcher(X), Topic(Y), PartOf(Y,X), Name(Z), PartOf(Z,X)$	
Then	Altavista	[key words = Z & "conference program" & Y, date $\geq$ 1997, partial URL = aaai.org]

Of course, as things evolve, it might be the case that the web site manager acquires partial knowledge about the sites of interest outside his/her web site. In that case, he/she can add this knowledge as as many new view definitions. For instance, knowledge about conferences, the URL where they can be usually found, as well as their domains and the way those domains can be related to research topics of people inside the web site can be easily expressed through the definition of new view predicates about conferences. They can then be used in the condition part of evocation rules.

Dealing with a set of evocation rules requires first to evaluate their condition part, second to use a strategy to choose which evocation rule has to be triggered among possibly several evocation rules whose conditions are satisfied. This problem is a specific instance of the general and well-known problem of choice strategy for a set of production rules. The point here is that since the evocation rules are related to objects and that their evaluation is triggered by external events (e.g, at a given period of time), the problem of their choice is much less crucial than for a usual expert system rule base.

Once evocation rules have been triggered and have returned documents that are likely to bring new information which might require an update inside the web site, the issue is to decide whether an update (and which update) has to be done. In order to decide or to suggest an update, document analysis techniques can be used in order to closely compare the syntactic content of the web pages that have been returned and of the web pages in the web site from which evocation rules have been triggered.

4 Conclusion and Perspectives

In this paper, we investigated the issues raised by checking potential update anomalies of web sites. The strong but reasonable assumption that we make is that modeling the semantics of a given web site is possible, while modeling the semantics of the sources outside of it is not feasible. As a consequence, two different update problems have to be considered. The first one is similar to the usual update problem in databases: languages for expressing integrity constraints have to be designed to fit with Web site modeling. The second one is different because we have no control over what happens outside the Web site. It is just possible to resort to existing search tools to go and search outside the potential relevant information and sources.

In this paper, we proposed a language based on logical rules to define a semantic model as well as semantic global constraints over a set of web pages composing a web site. We proposed an extension of this language with production rules in order to drive the use of existing search tools to search outside the web site pertinent information for updating it.

In order to ground the logical layer that we propose to add, on the semantic information that can be extracted from the structure of web pages, we distinguish base predicates from the others. The extension of base predicates can be easily extracted from the structure, the markers and the links existing in and between web pages.

We pointed out the main issues for evaluating the constraints and the production rules that we considered.

This work is preliminary and has to be pursued in several directions in order to build a real system on the above ideas and specifications. First, the language that we consider can be refined and extended. In particular, it is necessary to express constraints that deal with the deletion of information. Second, optimization algorithms have to be designed to deal with the specific and numerous data that are stored through the network as documents. Finally, they have to be connected to existing tools like search engines and also to tools for document analysis.

References

[ACM97] Serge Abiteboul, Sophie Cluet, and Tova Milo. Correspondence and translation for heterogeneous data. In *Proceedings of the International Conference on Database Theory: ICDT-97*, 1997.

[CGMH+94] Sudarshan Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom. The TSIMMIS project: Integration of heterogenous information sources. In proceedings of IPSJ, Tokyo, Japan, October 1994.

[DGL+96] D.Calvanese, G.De Giacomo, L.Iocchi, M. Lenzerini, and D. Nardi. Knowledge-based access to the web. In *Proceedings of AI*IA-96 conference*, 1996.

[LR96] Alon Y. Levy and Marie-Christine Rousset. Verification of knowledge bases using containment checking. In *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence*, 1996.

[LRO96] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Query answering algorithms for information agents. In *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence*, pages 40–47, 1996.

[YPW95] H. Garcia-Molina Y. Papakonstantinou and J. Widom. Object exchange across heterogeneous information sources. In *Proceedings of the International Conference on Data Engineering*, 1995.