# Knowledge-based access to the Web

Diego Calvanese, Giuseppe De Giacomo, Luca Iocchi, Maurizio Lenzerini, Daniele Nardi
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
*lastname*@dis.uniroma1.it

## 1 Introduction

The idea that knowledge can be used for finding information through the network has been developed in several directions. Most proposals use knowledge representation techniques in order to model the information stored in the network (Web) nodes, and then access the information through ad hoc procedures. On the other hand, the information that we access through the browsers is organized into structures that are coded in terms of the content of each page and of its relationship to other pages. Such structures typically guide humans in the search for a specific information. Our idea is to use this kind of structural information to build knowledge based tools for the access to the network. More specifically, we are working at a system which acquires information on a subject matter by extracting from the pages an explicit representation of the underlying structure, and use it to answer user's requests by directly pointing to the page containing the desired answer. Therefore, our envisioned system, is meant to act as an intelligent browser, which allows the user to reach the requested information by automatically performing navigation steps.

In this abstract we first present an architecture for a knowledge-based approach to the construction of the system with the above described goals. The system is based on a formalization of Web pages in terms of description logics. We then illustrate such a formalization, and discuss the reasoning features that are required in our framework.

## 2 Architecture

There are two main components in our architecture (see Figure 1): *Knowledge Acquisition* and *Navigation*. Knowledge acquisition is the process of building a knowledge base which contains an intensional description of the information of a typical site containing information on a given subject matter. For example, if the subject matter is University Department, the knowledge base contains a description of the typical arrangement of the pages containing the information on this subject, including the link structure. In addition, the knowledge base contains information on how to access actual homepages of University Departments. We do not discuss in further detail the process of knowledge acquisition in this abstract.

The second component of the architecture can be regarded as an intelligent browser, which takes a query from the user, on the subjects known by the system, and answers by presenting him a page which contains the required information. This is achieved by a two step process. The first step consists of finding one or more plans (sequences of links) that lead to a page satisfying the goal associated with the user request. The second step consists in the execution of the plan, which may require the discovery of failures and requests for replanning. The planning component of the system relies on the deductive planning framework developed for Propositional Dynamic Logics (PDLs)[1] [6, 3] and exploits the correspondence between them and description logics [7].

[1] Propositional Dynamic Logics are particular modal logics introduced for reasoning about program schemes [5].

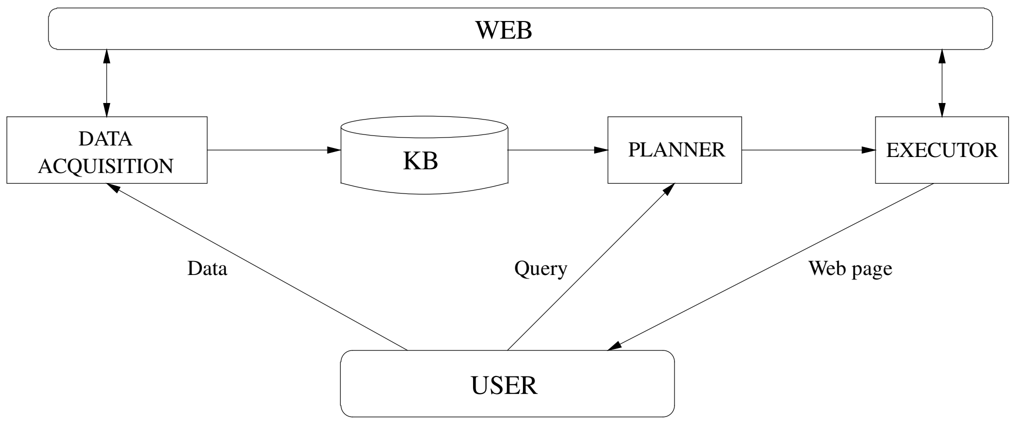


Figure 1: The system architecture

# 3 Representation of Web pages

Our representation of Web pages is based on description logics. Description logics [9] have been introduced with the aim of formalizing terminological systems that have been developed after KL-ONE. Such systems are based on the idea of representing knowledge by building hierarchical structures, sometimes called terminologies, where the descriptions of interest are classified according to their properties. The basic elements of the representation are called *concepts* and *roles*, which denote classes and binary relations, respectively. This general scheme is specialized in our framework according to the following ontology:

- The structure of a page is regarded as a concept description and a specific Web page as a concept instance.
- Each page has static properties, which are represented using (static) roles and refer to any kind of information that can be relevant to both the page structure and its content.
- Each page has dynamic properties, which are also represented using (dynamic) roles, interpreted as actions, in the sense that they allow jumping to other pages.

The novelty of our proposal lies in the interpretation of the network structures available on the Web, directly in terms of the representation structures used by the knowledge representation system in order to provide reasoning facilities.

# 4 Reasoning on Web pages

We use concepts to represent Web pages. Such descriptions model the information about both the static and dynamic aspects of prototypical pages on specific topics. The knowledge base can be viewed as a collection of structural descriptions of Web pages and assertions specifying effects of navigating links. Below we discuss static and dynamic aspects separately.

## 4.1 Reasoning on page content and structure

The static information contained in a page regards both its structure (i.e. its syntax) and its semantic properties. The representation of semantic properties is in the standard style of description logics, and is based on their capability of representing the relationships among the different entities forming the domain of interest. By exploiting the reasoning procedures developed for description logics we can, therefore, reason on the semantic structure of pages. This is necessary both to verify properties of page descriptions, and to check, in the process of executing a plan, that the actual pages encountered satisfy the expected requirements.

Syntactic information on pages can also be exploited during the reasoning process. We illustrate now briefly that we can in fact represent the syntactic structure of Web pages by means of description logics. We will see, however, that in order to capture all structural aspects, we need in general quite expressive logics.

Each Web page is a marked up document obeying the Hypertext Markup Language (HTML) syntax. The grammar generating the HTML language can be specified in the Standard Generalized Markup Language (SGML) [4], the metalanguage which has by now become the de facto standard for defining grammars for textual markup. Additional syntactic restrictions satisfied by HTML documents of a certain type (e.g. the homepages of university departments, which contain a list of links to pages describing research, teaching, etc.) can also be specified in SGML, by refining the grammar of HTML. SGML grammars are commonly referred to as Document Type Definitions (DTDs), and it is well known that they can be formalized by means of Extended Context Free Grammars (ECFGs), in such a way that marked documents that are instances of a DTD are seen as syntax trees of the corresponding grammar (see [8]).

We have addressed the problem of reasoning on DTDs (in order to determine for example if the set of marked documents generated by a DTD is a subset of the one generated by another DTD) by providing a framework where ECFGs formalizing DTDs are in turn formalized in terms of (quite expressive) description logics. This allows us to provide a logical view of DTDs, which is in fact based on the strong similarity that exists between marked documents and Kripke structures in terms of which concept expressions are interpreted. More precisely, a document of type $D$ is seen as a tree-like structure, where nodes represent the terminal and nonterminal symbols of the ECFG associated to $D$, and arcs reflect the hierarchical structure of the document.

Exploiting this similarity, we have designed a polynomial translation $t$ from DTDs to concepts of a special form. We sketch the idea behind the translation. Given a DTD $D$, let $G$ be the corresponding ECFG. We construct the concept $t(D)$ in such a way that it imposes the following conditions on its models:

- The model has the form of a finite tree. The root of the tree represents the start symbol of $G$.
- Each node representing a nonterminal symbol $N$, has a first successor corresponding to the start tag `<N>`, a last successor corresponding to the end tag `</N>`, and a sequence of intermediate successors reflecting the structure of the regular expression in the right hand side of the production defining $N$ (We assume without loss of generality that there is exactly one such production in $G$).

We note that, in order to impose all the above conditions, we actually need to use a very expressive description logic in which we can build arbitrary regular expressions over direct and inverse roles, impose functionality on roles and express the absence of cycles in the model. Reasoning in such a logic is nevertheless decidable and can be done in deterministic exponential time [1]. This allows us to decide various forms of equivalence and inclusion between DTDs, and therefore verify syntactic relationships that exist between different classes of Web pages.

In addition, we stress that, since we use the same formalism to represent both syntactic and semantic structure, it becomes possible to develop reasoning procedures that take both aspects into account, when trying to detect similarities between classes of documents.

## 4.2 Reasoning on Web navigation

Dynamic systems are typically modeled in terms of state evolutions caused by actions. A *state* represents a situation the system can be in, and an *action* causes a state transitions, thus making the system evolve from the current state to the next one. In our case each state is *being in a Web page*, and actions models the transition we get by following a *link* and *loading another page*, thus making us move from the current page to the next one.

At the schema level, the fact that we are in a Web page is represented by the description of such page, i.e. a concept with the characteristic discussed in the previous section. Actions are represented by specific *functional* roles, which are called *action-roles*, and the ability and the effects of performing an action are expressed by means of specific *axioms*. The reasoning service of interest is logical implication. We remark that this setting is exactly that of reasoning about action in PDLs (see [5, 6, 3]). However, we make use of the correspondence between PDLs and description logics, to express our knowledge on both static and dynamic aspects in a unique formalism.

The typical kind of reasoning we are interested in is that aiming at answering the following question (the planning problem): "is there a sequence of actions that, starting from an initial state, leads to a state where a given property (the goal) holds?". Under the assumption of deterministic actions (functional roles), this is captured by the following logical implication:

$$\Gamma \models S \sqsubseteq \exists \alpha^*.G$$

where: (i) $\Gamma$ is the set of axioms representing our knowledge on the Web pages and on navigating the Web; (ii) $S$ is a concept representing the knowledge on the current Web page; (iii) $G$ is a concept representing the goal, which is, in fact, a description of some aspect of the Web page we want to reach; (iv) $\exists\alpha^*.G$ (where $\alpha^*$ stands for any role of the form $R_1 \circ R_2 \circ \ldots \circ R_n$ with $n \geq 0$ and $R_i$ any action-role) expresses the existence of a finite sequence of Web links leading to a Web page where $G$ is satisfied. In the tradition of deductive planning, from a *constructive proof* of the above logical implication we extract an actual sequence of actions (a plan) that leads to the goal. Observe that reasoning about actions in this framework is decidable in deterministic exponential time [5, 3].

We have already applied this method for planning the actions of a mobile robot [2]. In that case we have restricted the language and exploited the use of procedural rules for representing and reasoning on dynamic knowledge in order to obtain an efficient implementation for a running robot.

# 5 Conclusions

We have proposed a general framework for knowledge-based access of information on the Web. Our framework is based on the use of description logics to represent content, syntax and links of Web pages. We have focused our attention on the representation and reasoning aspects, and exploited the correspondence between description and dynamic logic. In particular, we have argued that both reasoning on the syntactic structure and the generation of the sequences of links that lead to a desired goal page, are decidable problems for the chosen formalism. However, in order to build a system based on the above ideas several problems need to be tackled, among which we point out:

- To use practical knowledge representation systems based on description logics, one typically wants to rely on less expressive representation formalism. Consequently, it is necessary to find appropriate restrictions that conveniently simplify the representation.
- To support the knowledge acquisition process, suitable learning methods are of interest.
- To construct the representation for the Web pages, an integration with text processing techniques is required in order to gather semantic information from actual pages.

A prototype implementation is under development.

# References


[1] D. Calvanese, G. De Giacomo, and M. Lenzerini. Structured objects: Modeling and reasoning. In *Proc. of DOOD-95*, LNCS 1013, pages 229–246, 1995.

[2] G. De Giacomo, L. Iocchi, D. Nardi, and R. Rosati. Moving a robot from a theory of actions. In *Proc. of the AAAI'96 Workshop on Reasoning about Actions, Planning and Control: Bridging the Gap*, Portland (USA), 1996.

[3] G. De Giacomo and M. Lenzerini. PDL-based framework for reasoning about actions. In *Proc. of AI*IA-95*, LNAI 992, pages 103–114. Springer-Verlag, 1995.

[4] International Organization for Standardization. *ISO-8879: Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*, October 1986.

[5] D. Kozen and J. Tiuryn. Logics of programs. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science – Formal Models and Semantics*, pages 789–840. Elsevier, 1990.

[6] S. J. Rosenschein. Plan synthesis: A logical perspective. In *Proc. of IJCAI-81*, pages 331–337, 1981.

[7] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of IJCAI-91*, pages 466–471, Sydney, 1991.

[8] D. Wood. Standard generalized markup language: Mathematical and philosophical issues. In *Computer Science Today, Recent Trends and Developments*, LNCS 1000, pages 344–365. Springer-Verlag, 1995.

[9] W. A. Woods and J. G. Schmolze. The KL-ONE family. In F. W. Lehmann, editor, *Semantic Networks in Artificial Intelligence*, pages 133–178. Pergamon Press, 1992.