The Multilingual Thesaurus of LAURIN

Diego Calvanese, Tiziana Catarci, Maurizio Lenzerini, Giuseppe Santucci
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
lastname@dis.uniroma1.it

ABSTRACT

Among the wide range of digital libraries, an interesting, yet quite neglected, subclass is constituted by those exclusively dealing with newspaper clippings. Compared with book-oriented digital libraries, clipping libraries are more difficult to seize, since they are wide and unstructured, and the subjects and content of a clipping are completely heterogeneous. LAURIN is an EU-funded project involving seventeen participants from several countries, including two software companies and a large group of libraries, whose main purpose is to set up a network of digitalized newspaper clipping archives that can be easily accessed through the Internet, for searching and retrieving clippings. The project also provides the libraries with models and methodologies to be used for scanning, digitalizing, storing, indexing, and making accessible newspaper clippings. The core of the LAURIN system is an *integrated multilingual Thesaurus*. In this paper we illustrate how to express the thesaurus in terms of a knowledge base, and how to exploit such a knowledge base to improve the fundamental task of clipping retrieval.

1. INTRODUCTION

During the last ten years Digital Libraries (DLs) have become an important and diffused information technology, with particular attention to book DLs and scientific document collections (see e.g, [16, 2, 8, 17, 14, 7, 15, 13, 18]).

However, there is an important kind of physical library, namely the newspaper clipping collection, which had not got sufficient attention in the digital world (a notable exception is the Historical Newspaper Digital Library project [1], which deals with old clippings).

Compared with book-oriented digital libraries, clipping libraries are more wide and unstructured, since there are not specific standards to collect and classify newspaper clippings. The subjects and content of a clipping are completely heterogeneous: it could be a small article, some photos with some text as caption, or a whole article with diagrams and photos spanning several pages. Additionally, the information associated with clippings may differ from the usual one stored in digital libraries. For example, the author information, which is mandatory in traditional archives may be irrelevant or even missing for clippings. Also, users of clipping archives are typically interested in articles in the original form in which they appeared in the newspaper. As a consequence, not only the full-text of the clippings and the images possibly associated with the text must be stored in the archive, but also a picture of the scanned version of the original newspaper page, and obviously this requirement poses particular challenges with respect to data storage.

Given this diversity, it is extremely difficult to come up with a general clipping catalog system, whereas many specific clippings can be retrieved in a library systematically interested in some subjects or in a library that institutionally collects and catalogs newspapers.

LAURIN (Libraries and Archives Collecting Newspaper Clippings Unified for their Integration into Networks) is an EU-funded Project[1] involving seventeen participants from several countries, including two software companies and a large group of libraries that want to make easily available and give wide visibility to the large cultural heritage they collect and catalog daily. The high number of users/libraries involved in this project gives the opportunity to spread culture and information to a wider public by means of the Internet. LAURIN has two major goals:

- To set up a network of digitalized newspaper clipping archives that can be accessed through the Internet in a centralized fashion, for searching and retrieving clippings.
- To provide a generic model to be used by individual libraries for scanning, digitalizing, storing, and indexing newspaper clippings, and making them accessible via the LAURIN network.

Concerning objective 1, since many users are ill equipped to translate their search requirements into precise queries, and they often prefer to use browsing as retrieval strategy, the LAURIN interface offers, besides traditional keyword based

[1]Telematics Program, Libraries Project LB-5629/A, `http://laurin.uibk.ac.at/`

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SEKE '02, July 15-19, 2002, Ischia, Italy.
Copyright 2002 ACM 1-58113-556-4/02/0700…$5.00.

.

search methods, also the possibility of browsing the clipping collection by argument, organizing the document space in a manner that is readily understood by users. Such activity is supported by the use of an *integrated multilingual Thesaurus*, which plays a central role in the LAURIN system. The user sees a unified search space and therefore s/he can ignore the existence of different information sources, i.e., libraries. However, s/he can also select a library on demand, based on the description of its characteristics, in order to restrict her/his attention to specific topics covered by a certain library only. Requests can be formulated in any of the languages supported by the system (currently English, French, German, Italian, Norwegian, Spanish, and Swedish) and the system provides translations for the purpose of keyword and content based search.

To fulfill the above requirements, the LAURIN system is organized around a *central node*, which is connected via the Internet to a set of *local nodes*, one for each participating library. The digitalized clippings and their full-text (obtained via OCR) are stored in the local nodes, together with a local, possibly personalized, copy of the Thesaurus. The central node contains indexing data about all clippings stored in the local nodes, and a centralized copy of the multilingual Thesaurus with globally validated entries. A constant flow of information from the local nodes to the central node ensures that the latter is up to date.

Concerning objective 2, the integrated Thesaurus system supports librarians in indexing and handling the clippings. This facilitates both the librarians' archiving activity and an improved local access.

The LAURIN project, aiming at producing a highly interactive system, is being carried out by following a rigorous 'user-centered" design methodology [12], so that the envisioned solutions are really based on the user needs and requirements. This kind of approach is particularly appropriate for LAURIN given the large number of libraries involved in the project, playing the double role of end users and test sites. Also, it is worth noting that librarians are "extremely expert" users in their application domain (i.e., libraries and archives, books and journals). For instance, such users have their very precise idea of what a digital library is, and do not accept something different from computer scientists. From a librarian point of view, a digital library is very different from an XML repository! Also, librarians are very familiar with classifications, thesauri and taxonomies. They used to have their own classifications for many years and do expect something "better" from information technology, but very often this does not seem to be the case.

Among other things, librarians involved in LAURIN have stressed the importance of having an indexing system and especially a thesaurus reflecting both their requirements and the needs of people who want to access the clipping archives to retrieve information of interest. Up to now the lack of structured thesauri, supporting a semantic classification of clippings, has prevented final users from accessing the clipping libraries themselves. What usually happens (at least in the many European libraries we have analyzed in LAURIN) is that the user asks a vague query to a ''human interface" (i.e., the librarian) and s/he first tries to refine the query (for instance, enlarging or restricting it) and then searches for the clipping potentially matching the user's interests in the archive. This is obviously a very time-consuming activity and can be carried on only through a physical interaction (or, at least, phone-based interaction) between the librarian and the user. Very difficulty the same pattern could be replicated on the Internet, where, on the other hand, the user could have a remote and universal access to all available digital libraries of clippings.

In order to realize an Internet-based service which aims at replicating at least the efficiency (even if it could never get all other qualities of a human-human interaction) of the librarian-mediated retrieval, the LAURIN project concentrated on two crucial components of the system, namely the multilingual thesaurus and the visual interfaces (both the indexing interface for the librarian and the retrieval interface for the end-user). In this paper we focus on the thesaurus, while the system architecture and the interfaces have been described in [3], and we will just recall them in the following. It is important to note that, even if the present realization of the LAURIN thesaurus is already an achievement with respect to the previous situation, we are still working towards extending the thesaurus with reasoning capabilities, reflecting the reasoning implicitly performed by the librarian when "processing" the end-user request.

The goal of this paper is exactly to illustrate how to express the thesaurus in terms of a knowledge base expressed in a logic-based formalism, and how to exploit such a knowledge base and the associated automated reasoning capabilies to improve the fundamental task of clipping retrieval.

Comparing the LAURIN approach with existing literature, one may note that during the last years, digital library systems have not made many efforts to solve user-interaction problems. Only recently, new projects (e.g. University of Stanford[2] and University of Michigan[3] DL Projects) are developing a more complex model of information-seeking tasks. Display of information, visualization of, and navigation through large information collections, as well as linkages to information manipulation/analysis tools can be identified as key areas for research.

Other recent proposals deal with multi-language access to digital libraries and archives; integration of many different services, where information search is just a subpart of a more complex task; and easy refining of results and revisiting of search process. For example, the expansion and refinement of queries based on lexical relationships between documents, which are automatically extracted from the document collection, is addressed in [5]. A prototype implementation of a general user interface paradigm which is capable of modelling iterative query refinement is described in [10].

Finally, another key issue addressed by the LAURIN project is the distributed nature of the collection of clippings. A distributed query system for preexisting library catalogs and structured databases (storing bibliographic data), based on an ad-hoc query language, has been developed in the HARP

[2] `http://www-diglib.stanford.edu/diglib/`
[3] `http://http2.sils.umich.edu/UMDL/`

project [11].

The paper is organized as follows. Section 2 recalls the overall LAURIN system architecture. Section 3 describes the logic-based formalism that we use for expressing the knowledge base, and illustrates the structure of the knowledge base itself. Finally, Section 4 discusses the features of our approach that allow us to provide automated reasoning support for clipping retrieval.

2. THE LAURIN APPROACH

This section recalls the overall LAURIN architecture, shortly describing its principal components, and summarizes the main system functionalities librarians and end users are provided with.

2.1 System Architecture

The overall LAURIN architecture is displayed in Figure 1. It consists of a set of nodes connected through the Internet: one node for any participant library plus a *central node* collecting data from local nodes and providing the end user with a uniform query environment. The central node hosts a relational database in which summary data coming from the local nodes are stored (i.e., clipping title, date, newspaper, author, etc.). Local nodes are in charge of clipping scanning and indexing; moreover they store all the information about acquired clippings: summary data, full clipping text, and clipping images. LAURIN clippings are strictly related with the LAURIN *Thesaurus* that is stored in the central node and replicated in the local nodes. There is a constant flow of information from the local nodes towards the central node, updating the central database with new clippings and new thesaurus entries. Periodically, the thesaurus administrators validate the proposed thesaurus entries and the central node propagates such validations towards the local nodes. When a user formulates a query, the central node tries to obtain the result using the central data, involving the local nodes only when specific full text based queries are issued or the clipping images are requested. The central node is in charge of collecting the answers coming from local nodes and presenting the final result to the user. The central node contains a Z39.50 [20] interface as well, which allows for acting as a Z39.50 server, exporting all LAURIN summary data. Depending on local hardware/strategical issues, each local node may be directly queried by the end users through a Web interface and/or a Z39.50 interface.

2.2 System Main Functionalities

The LAURIN system provides with different functionalities two classes of users, namely internal and external users.

Internal users are part of the library staff who operate on the system to accomplish the following tasks: (a) to ask queries (in this case they embody the role of external user); (b) to input clippings; and (c) to administer the system. The main task of the internal user is clipping input, that is, scanning, OCR-ring and cataloging of clippings. This activity is performed only on local nodes. Some internal users, playing the role of system administrators, are also allowed to deal with the inner part of the Central Node. In particular, the system provides an interface to periodically validate new Thesaurus entries, coming from the local node clipping classification.

External users are users who access the system, independently from the location of the nodes, to submit a query and, hopefully, get an answer. The most general query is supposed to be formulated as follows: give me all clippings about *something*. The "something" part must be defined in a way that produces valid results (low noise in results), which can be incrementally refined, and must be simple to define by an average user (not extremely expert on the clipping collection or "casual").

2.2.1 Internal Users' Activities

Internal users perform their activities related with indexing and storing clippings only on local nodes through an ad-hoc interface to a sophisticated OCR system.

Several indexing mechanisms, which have been decided in strict cooperation with the librarians, are available. In particular, the Prime Index is the basic information on clipping/article that otherwise will be lost during the clipping process (name of newspaper, page, rubric, date, ...). The Bibliographic Index contains the basic bibliographic information on clipping/article (author, title, subtitle, text type of an article). The Keyword Index is an association of known terms from the Thesaurus with clipping/article which is automatically generated from the article full-text. The Content Index is an association of clipping/article with normalized terms from the Thesaurus resulting from a human content analysis of the clipping/article. The Free Index is an association of clipping/article with subject headings that are not part of the Thesaurus resulting also from the human content analysis. Indeed, it may happen that, while indexing a clipping using the thesaurus concepts, a librarian is not able to find a thesaurus entry satisfying her/his needs. In this case the clipping acquisition module allows for associating the clipping with a *new* (*candidate concept*) that is in the free index. Candidate concepts are locally available for query formulation and are candidates to become new entries in the Thesaurus. The Full-text Index is a computer based retrieving of all normalized terms in the clipping/article (including terms that are not in the Thesaurus), generated and maintained by a full-text information retrieval engine.

The above indices are used in developing different clipping classifications, which are in turn exploited by the search mechanisms the external users are provided with.

Local node Thesaurus Administrators are special internal users, whose main goal is to administrate the local node Thesaurus. They typically update the local node Thesaurus with information associated with new clippings. The Thesaurus is queried and/or browsed to find relevant entries that can be associated with a clipping. Whenever an entry that is already in the Thesaurus needs to be associated with a clipping the association is stored in the local node database and transmitted to the central node together with the clipping data. Candidate entries, i.e., entries coming from the free index that are not in the Thesaurus, are analyzed, inserted in the local Thesaurus, and eventually associated with the clipping. The candidate entries are transmitted to the central node for validation and also kept in the local node

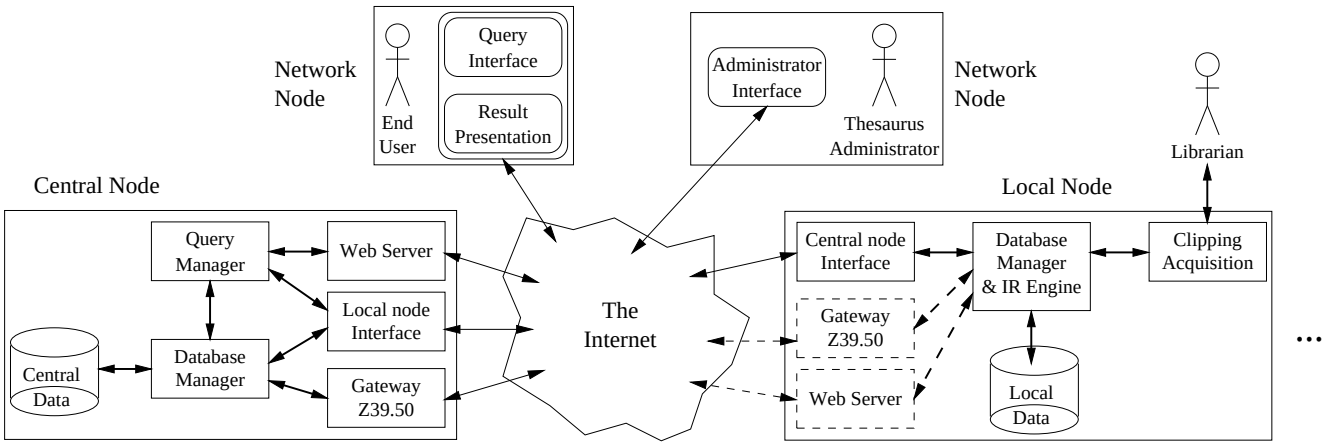

Figure 1: The overall Laurin Architecture

(together with their association with clippings) until validation is performed.

Central node Thesaurus Administrators have two main tasks, namely to build, refine, and modify the Thesaurus and to validate candidate entries. The former is an off-line activity, that alters the Thesaurus content, independently from the activity of local nodes (e.g., correcting errors, adding new terms for existing concepts, etc.). The updates resulting from such an activity are propagated towards local nodes. The latter is part of the routine LAURIN job, and implies the analysis of the candidate entries coming from local nodes, which may be inserted in the Thesaurus, merged with existing entries or even rejected. A full handshake protocol is adopted in this phase to avoid inconsistent clipping classification.

2.2.2 External Users' Activities

External users interact with the central node and the system provides (on demand) a description of the LAURIN consortium and of the involved local nodes, allowing a direct connection to local nodes hosting a Web query interface. If a user wants to ask a query across two or more local nodes (all nodes as an extreme case) s/he interacts only with the central node that acts as a broker with respect to the local nodes. Also, an external user connected to the central node can browse the central Thesaurus to search for associated clippings. Using several kinds of interfaces the user is allowed to formulate a multilingual query in which the Thesaurus plays three different roles:

1. it is a guide to understand the *classification* of the clippings stored in the LAURIN distributed database;
2. if the user has requested a multilingual search it translates the involved terms;
3. if requested by the user, it can be used to modify the scope of a query (e.g., finding not only the clippings containing the word X but also the clippings containing a synonym of X or a more specific term for X). The system provides the user with a Thesaurus browser, allowing for hierarchical navigation among terms. As an example, the user is able to select the location "Rome", either using the alphabetical order of "Rome" within a subset of the geographical Thesaurus data, or following the path "Earth → Europe → Italy → Rome". Every domain is multilingual, that is every concept is translated in the corresponding word in every language involved in the project.

Summarizing, when keywords are used in a query, the Thesaurus is accessed to expand the set of keywords according to the user specified criteria (more general terms, related terms, terms in different languages, etc.). For keyword expansion the Thesaurus of the node to which the user is connected (either central or local) is used. The identifiers of clippings associated with the expanded set of keywords can then be retrieved and presented to the user.

Once the user has formulated a query, the central node query manager answers it as follows. First of all, it analyses the query, splitting it into two parts, one related to the central node database and one related to the local nodes (i.e., the part of the query that refers to the full text of clippings). To solve the former it starts a query against the central database; to compute the latter it selects the local nodes that may possibly contribute to the query (e.g., to look for an Italian clipping at the Uppsala node in Sweden makes no sense) and then sends the query to the selected local nodes. Once each local node has returned a list of clipping IDs the query manager merges such lists with the answer it got by the central database, presenting the final result to the end-user.

When the query has been processed, the user can interactively refine the result. When s/he has reached her/his goal, s/he can ask the system for a summary of the results, containing all the necessary information needed to get the clippings (involved nodes, cost, etc.).

2.2.3 Thesaurus Structure

The LAURIN multilingual thesaurus is presently implemented as part of the overall database comprising several other data sources, namely *clipping data*, *periodical data*, *author data*, *administrative data*.

In the database, thesaurus entries constitute the class Concept, and are related to clippings, languages, categories of entries (i.e., Persons, Institutions, Organisations, Companies, Geographical Locations, Keywords, Events, Actions, Properties, Time Keywords), and especially to other thesaurus entries through various relationships. A limited number of thesaurus entries are chapter headings, they represent the upmost entry points for the thesaurus hierarchy. Several relations are defined between entries. For instance, if X and Y are entries, we may assert that X is-a Y, or X is-part-of Y, or X is-associated-to Y. If Y belongs to the category Geographical Locations, we may state that X is-located-in Y. If both X and Y are geographical locations, we may assert that X is-geographic-parent of Y. Other specific relations can be defined on terms of certain categories, such as persons working in institutions, facts happening at a certain time, etc.

These relationships are exploited by the Thesaurus Browser, which is part of the LAURIN user interface.

3. THE LAURIN KNOWLEDGE BASE

In this section we illustrate our technique for constructing a special knowledge base, called the LAURIN Knowledge Base (LKB), which will be used in query processing in the LAURIN system. The LKB is expressed in a particular logical formalism belonging to the family of Description Logics [6]. We introduce such a formalism in the following subsection.

3.1 The Description Logic $\mathcal{DLR}$

Description Logics[4] were introduced as an attempt to provide a formal ground to Semantic Networks and Frames. In Description Logics, the domain of interest is modeled by means of *concepts* and *relationships*, which denote classes of objects and relations, respectively. Generally speaking, a DL is formed by three basic components. First, a *description language*, which specifies how to construct complex concept and relationship expressions (also called simply concepts and relationships), by starting from a set of atomic symbols and by applying suitable constructors. Second, a *knowledge specification mechanism*, which specifies how to construct a knowledge base, in which properties of concepts and relationships are asserted. Third, a set of *reasoning procedures* provided by the logic.

The set of allowed constructors characterizes the expressive power of the description language. In LAURIN, we use the Description Logic $\mathcal{DLR}$, introduced in [4]. We assume to deal with a finite set of atomic relationships and concepts, denoted by $\mathbf{P}$ and A respectively. We use $\mathbf{R}$ to denote arbitrary relations (of given arity between 2 and n_{max}), and C to denote arbitrary concepts, respectively built according to the following syntax

$$\begin{array}{rcl} \mathbf{R} & ::= & \top_n \mid \mathbf{P} \mid (\$i/n:C) \mid \neg\mathbf{R} \mid \mathbf{R}_1 \sqcap \mathbf{R}_2 \\ C & ::= & \top_1 \mid A \mid \neg C \mid C_1 \sqcap C_2 \mid \exists[\$i]\mathbf{R} \mid (\leqslant k\,[\$i]\mathbf{R}) \end{array}$$

where i and j denote components of relations, i.e. integers between 1 and n_{max}, n denotes the arity of a relation, i.e. an integer between 2 and n_{max}, and k denotes a nonnegative integer.

[4]See `http://dl.kr.org/` for the home page of Description Logics.

$$\begin{array}{rcl} \top_n^{\mathcal{I}} & \subseteq & (\Delta^{\mathcal{I}})^n \\ \mathbf{P}^{\mathcal{I}} & \subseteq & \top_n^{\mathcal{I}} \\ (\neg\mathbf{R})^{\mathcal{I}} & = & \top_n^{\mathcal{I}} \setminus \mathbf{R}^{\mathcal{I}} \\ (\mathbf{R}_1 \sqcap \mathbf{R}_2)^{\mathcal{I}} & = & \mathbf{R}_1^{\mathcal{I}} \cap \mathbf{R}_2^{\mathcal{I}} \\ (i/n:C)^{\mathcal{I}} & = & \{(d_1,\ldots,d_n) \in \top_n^{\mathcal{I}} \mid d_i \in C^{\mathcal{I}}\} \\ \top_1^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\ A^{\mathcal{I}} & \subseteq & \Delta^{\mathcal{I}} \\ (\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\ (C_1 \sqcap C_2)^{\mathcal{I}} & = & C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\ (\exists[\$i]\mathbf{R})^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \exists(d_1,\ldots,d_n) \in \mathbf{R}^{\mathcal{I}}.\, d_i = d\} \\ (\leqslant k\,[\$i]\mathbf{R})^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \\ & & \quad \sharp\{(d_1,\ldots,d_n) \in \mathbf{R}_1^{\mathcal{I}} \mid d_i = d\} \leq k\} \end{array}$$

Figure 2: Semantic rules for $\mathcal{DLR}$ ($\mathbf{P}$, $\mathbf{R}$, $\mathbf{R}_1$, and $\mathbf{R}_2$ have arity n)

We consider only concepts and relationships that are *well-typed*, which means that only relations of the same arity n are combined to form expressions of type $\mathbf{R}_1 \sqcap \mathbf{R}_2$ (which inherit the arity n), and that $i \leq n$ whenever i denotes a component of a relation of arity n.

The semantics of expressions is specified through the notion of interpretation. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is constituted by an *interpretation domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ that assigns to each concept C a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, to each regular expression E a subset $E^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each relation $\mathbf{R}$ of arity n a subset $\mathbf{R}^{\mathcal{I}}$ of $(\Delta^{\mathcal{I}})^n$, such that the conditions in Figure 2 are satisfied. We observe that $\top_1$ denotes the interpretation domain, while $\top_n$, for $n > 1$, does *not* denote the n-Cartesian product of the domain, but only a subset of it, that covers all relations of arity n. It follows, from this property, that the "$\neg$" constructor on relations is used to express difference of relations, rather than complement.

Using (concept and relationship) expressions, knowledge about concepts and relationships, and about the participation of individuals in concepts and relationships can be expressed through the notion of knowledge base. In $\mathcal{DLR}$, a knowledge base is constituted by a finite set of *inclusion assertions* of the form

$$\begin{array}{rcl} \mathbf{R}_1 & \sqsubseteq & \mathbf{R}_2 \\ C_1 & \sqsubseteq & C_2 \end{array}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are of the same arity, and *membership assertions* of the form

$$\mathbf{R}(a_1,\ldots,a_n) \qquad\qquad C(a)$$

where n is the arity of $\mathbf{R}$, and $a, a_1, \ldots, a_n$ are individuals.

An interpretation $\mathcal{I}$ *satisfies* an assertion $\mathbf{R}_1 \sqsubseteq \mathbf{R}_2$ (resp. $C_1 \sqsubseteq C_2$) if $\mathbf{R}_1^{\mathcal{I}} \subseteq \mathbf{R}_2^{\mathcal{I}}$ (resp. $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$). Interpretations can be extended to individuals by assigning to each individual a and element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. An interpretation $\mathcal{I}$ satisfies an assertion $C(a)$, if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and it satisfies an assertion $\mathbf{R}(a_1,\ldots,a_n)$, if $(a_1^{\mathcal{I}},\ldots,a_n^{\mathcal{I}}) \in \mathbf{R}^{\mathcal{I}}$. An interpretation that satisfies all assertions in a knowledge base $\mathcal{S}$ is called a *model* of $\mathcal{S}$.

3.2 Structure of the LKB

Generally speaking, the LKB is a logical representation of all the knowledge that is possessed by the system, and that can be used in query processing. The LKB represents knowledge about several aspects, which we now describe.

Clipping. This is the central aspect of the LKB, and concerns the information and fields related to the clippings, such as the author of the corresponing articles, the periodical to which it belongs, the multimedia objects associated with it, the language of the article, the association with the thesaurus entries, and so on.

Periodicals. The main issue of this LKB portion is to store all information and fields related to a periodical, namely the historical tracking of the newspaper/magazine, the place of publication (city and country), editors, frequency of publication, title, political attitude, supplements and merges of newspapers/magazines.

Authors. Although authors could represent specific thesarus entries, for efficiency reasons they are stored separately. Instead, the country an author belongs to is a thesaurus entry.

Administrative data. Administrative data are maintained mainly for monitoring purposes. We will not discuss it any further.

Thesaurus. This is the aspect concerning all thesaurus entries, and the relationships between the entries and the other concepts of the knowledge base. Lack of space prevents us from describing this part of the knowledge base in detail. We simply provide some examples to illustrate the role of assertions in expressing the LKB.

The following assertion is used to specify the types of the arguments of relation Denote. In particular, $\mathsf{Denote}(c, t, \ell)$ implies that c is a Concept, t is a Term, and ℓ is a Language, with the intuitive meaning that c is denoted by t in language ℓ (Note that some entries are denoted by the same term in every language, e.g., an entry denoting a person):

$$\begin{aligned}\mathsf{Denote} \sqsubseteq\ & (\$1/3 : \mathsf{Concept}) \sqcap \\ & (\$2/3 : \mathsf{Term}) \sqcap \\ & (\$3/3 : \mathsf{Language})\end{aligned}$$

In order to impose that every concept belongs to a category, and that for every concept c there exist at least one language ℓ and a term t in ℓ denoting c, we can use the following assertions:

$$\begin{aligned}\mathsf{Concept} &\sqsubseteq \exists[\$1]\mathsf{Is\text{-}Of\text{-}Category} \\ \mathsf{Concept} &\sqsubseteq \exists[\$1]\mathsf{Denote}\end{aligned}$$

The fact that every clipping is related to at least one concept can be expressed as follows:

$$\mathsf{Clipping} \sqsubseteq \exists[\$1](\mathsf{Related\text{-}to} \sqcap (\$2/2 : \mathsf{Concept}))$$

As we said before, several relations are defined between entries, depending on the category of the entries. The properties of these relations are again modeled in terms of assertions. For example, the assertion

$$\begin{aligned}\exists[\$1]\mathsf{Located\text{-}In} \sqsubseteq\ & \exists[\$1](\mathsf{Is\text{-}Of\text{-}Category} \sqcap \\ & (\$2/3 : \mathsf{Geographical\text{-}Location}))\end{aligned}$$

expresses the property that only concepts of category Geographical-Location participate in the relation Located-In.

3.3 Reasoning on the LKB

The large body of research in Description Logics has produced sophisticated methods for reasoning over Description Logics knowledge bases. Due to space limitatio, we cannot describe such methods in detail. We refer the interested reader to [6]. Here, we simply list the most important reasoning tasks that can be carried out over the LKB, and that will be used in the rest of this paper.

- Check wether a concept C_1 is a subset of another concept C_2 in all the models of the LKB $\mathcal{T}$, denoted $C_1 \sqsubseteq_{\mathcal{T}} C_2$.
- Check wether a concept C_1 is disjoint from another concept C_2 in all the models of the LKB $\mathcal{T}$, denoted $C_1 \otimes_{\mathcal{T}} C_2$.
- Check whether an individual is an instance of a concept in all the models of the LKB $\mathcal{T}$.

The above methods can be directly implemented by making use of existing Description Logic systems, such as the one described in [9].

4. REASONING SUPPORT FOR QUERY PROCESSING

We now describe how to exploit the reasoning techniques associated to the LAURIN Knowledge Base in order to evaluate queries posed to the system.

As a first step, we want to single out a class of concepts that is particularly interesting in our context. Since the LAURIN user is interested in retrieving clippings, we will assume that all the queries ask for a set of clippings satisfying a certain condition. Thus, query formulation reduces to expressing such a condition. Formally, we call *c-concept* any expression of the form

$$\{\ x\ \mid\ \mathsf{Clipping}(x)\ \wedge\ \alpha(x)\ \}$$

where Clipping is the concept representing all the clippings, and α is a $\mathcal{DLR}$ concept. A *query* in LAURIN is simply a c-concept, where α expresses the conditions that the retrieved clippings must satisfy. If c is an instance of the concept Clipping in the LKB $\mathcal{T}$, and $C = \{\ x \mid \mathsf{Clipping}(x)\ \wedge\ \alpha(x)\ \}$ is a c-concept, then we say that c *$\mathcal{T}$-conforms to* C, if c satisfies α in all the models of $\mathcal{T}$. Note that all the reasoning tasks mentioned above can be used in order to reason about conformance to c-concepts.

Nest, we introduce the notion of clipping base, that is used to formalize the way how the systems manages the various clippings retrievable by the users.

DEFINITION 1. *A clipping base $\mathcal{B}$ is a triple $\mathcal{B} = \langle \mathcal{T}, \mathcal{C}, \mathcal{I} \rangle$, where*

- *$\mathcal{T}$ is a LKB,*
- *$\mathcal{C}$ is a set of c-concepts, with the assumption that for each pair $C_1, C_2 \in \mathcal{C}$, it is known whether $C_1 \sqsubseteq_{\mathcal{T}} C_2$, and whether $C_1 \otimes_{\mathcal{T}} C_2$;*
- *$\mathcal{I}$ is a set of clippings, with the assumption that for each $c \in \mathcal{I}$ there is at least one $C \in \mathcal{C}$ such that c $\mathcal{T}$-conforms to C, and for each pair $c \in \mathcal{I}$, $C \in \mathcal{C}$, it is known whether c $\mathcal{T}$-conforms to C.*

The definition makes it clear that, in the clipping base, the clippings are classified in terms of a set of c-concepts, denoted by $\mathcal{C}$. The elements of $\mathcal{C}$ are organized on the basis of the two fundamental relationships holding between c-concepts, namely subsetting and disjointness. Thus, given two c-concepts C_1, C_2 in $\mathcal{C}$, the system knows whether C_1 is a subset of C_2, and whether C_1 is disjoint from C_2. Note that the reasoning techniques mentioned above are crucial for this purpose. Moreover, for each clipping c, and for each c-concept C in $\mathcal{C}$, the system knows whether c $\mathcal{T}$-conforms to C. Again, this is achieved by means of the reasoning techniques.

A query posed to a clipping base is expressed as a c-concept, used to retrieve all clippings that satisfy the definition of the concept. The formal semantics of queries is specified by the following definition.

DEFINITION 2. *The evaluation of a query Q over a clipping base $\mathcal{B} = \langle \mathcal{T}, \mathcal{C}, \mathcal{I} \rangle$ returns as an answer the set $Q(\mathcal{B})$ of all clippings $c \in \mathcal{I}$ such that c $\mathcal{T}$-conforms to Q.*

The way our approach uses the information represented in the clipping base is synthesized by an algorithm for computing the answer $\mathcal{Q}(\mathcal{B})$ to a query Q posed to a clipping base $\mathcal{B} = \langle \mathcal{T}, \mathcal{C}, \mathcal{I} \rangle$. The algorithm exploits the possibility of reasoning over $\mathcal{T}$, and works by maintaining two sets $\mathcal{S}$ and $\mathcal{J}$, of c-concepts and clippings respectively. The algorithm computes a set $\mathcal{A}(\mathcal{B}, Q)$ of clippings by proceeding as follows:

1. Let $\mathcal{S}$ be equal to $\mathcal{D}$, and let $\mathcal{J}$ be equal to $\mathcal{I}$.
2. While $\mathcal{S}$ is not empty, repeatedly select a c-concept C from $\mathcal{S}$ such that there is no $C' \in \mathcal{S}$ with $C \sqsubseteq_{\mathcal{T}} C'$, and do the following:
   (a) If $C \equiv_{\mathcal{T}} Q$, then let $\mathcal{A}(\mathcal{B}, Q)$ be all the clippings c in $\mathcal{I}$ such that c $\mathcal{T}$-conforms to C, and stop.
   (b) If $C \sqsubseteq_{\mathcal{T}} Q$, then
      (b.1) move from $\mathcal{J}$ to $\mathcal{A}(\mathcal{B}, Q)$ all clippings that $\mathcal{T}$-conform to C,
      (b.2) remove from $\mathcal{S}$ every c-concept C' such that $C' \sqsubseteq_{\mathcal{T}} C$,
      (b.3) continue with the next iteration of the while-loop.
   (c) If $Q \sqsubseteq_{\mathcal{T}} C$, then
      (c.1) remove C from $\mathcal{S}$,
      (c.2) for every c-concept C' in $\mathcal{S}$ such that $C' \otimes_{\mathcal{T}} C$, remove C' from $\mathcal{S}$ and remove from $\mathcal{J}$ every clipping that $\mathcal{T}$-conforms to C',
      (c.3) continue with the next iteration of the while-loop.
   (d) If $C \otimes_{\mathcal{T}} Q$, then
      (d.1) remove from $\mathcal{S}$ every c-concept C' such that $C' \sqsubseteq_{\mathcal{T}} C$,
      (d.2) remove from $\mathcal{J}$ every clipping e that $\mathcal{T}$-conforms to C',
      (d.3) continue with the next iteration of the while-loop.
   (e) Otherwise, remove C from $\mathcal{S}$, and continue.
3. Add to $\mathcal{A}(\mathcal{B}, Q)$ every clipping c in $\mathcal{J}$ that $\mathcal{T}$-conforms to Q.

The correctness of the above algorithm can be shown by demonstrating that, if $\mathcal{B} = \langle \mathcal{T}, \mathcal{C}, \mathcal{I} \rangle$ is a clipping base, and Q is a query, then the set $\mathcal{A}(\mathcal{B}, Q)$ computed by the algorithm above is equal to $Q(\mathcal{B})$. This can be done by using the following arguments.

Since Step 3 considers all clippings whose conformance to Q could not be determined by looking only at the concepts in $\mathcal{C}$, it is sufficient to show that Step 2 of the algorithm does not remove from $\mathcal{A}(\mathcal{B}, Q)$ any clipping that contributes to $Q(\mathcal{B})$. Step 2.a is obvious: if C is $\mathcal{T}$-equivalent to Q, then the answer to Q is the set of clippings in $\mathcal{I}$ that $\mathcal{T}$-conform to C. Step 2.b deals with the case where C is a subset of Q. In such a case, the set of clippings conforming to C takes part to the answer to the query. Moreover, since such a set comprises all clippings conforming to the concepts that are $\mathcal{T}$-included in C, these concepts need not to be considered anymore and are discarded. Step 2.c considers the case where Q is $\mathcal{T}$-included in C. Since the clippings satisfying Q are among those that conform to C, the algorithm discards all clippings conforming to some concept that is $\mathcal{T}$-disjoint from C. Step 2.d takes care of the case where Q is $\mathcal{T}$-disjoint from C, and therefore, discards all concepts that are $\mathcal{T}$-included in C, and excludes from the answer all clippings that $\mathcal{T}$-conform to C.

Observe that the above method can be seen as an adaptation of the semantic indexing technique developed in Description Logics [19], where concepts act as semantic indexes on clippings in the clipping base. In this way, they help in improving performance of query evaluation with respect to the brute force approach of evaluating clippings one by one. In other words, reasoning on the LKB allows for a more effective query evaluation process. Obviously, since comparing concepts is costly, the method pays off when the choice of the c-concepts in $\mathcal{C}$ is the right one, depending on the kind of queries that one expects from the users, and when the size of concepts is small (e.g., logarithmic) with respect to the size of the clippings, which is usually the case.

5. REFERENCES

[1] R. B. Allen and J. Schalow. Metadata and data structures for the historical newspaper digital library. In *Proc. of the 8th Int. Conf. on Information and Knowledge Management (CIKM'99)*, pages 147–153, 1999.

[2] D. Calvanese, T. Catarci, V. Curci, E. Melis, A. Rastellini, and G. Santucci. The overall laurin architecture. Technical Report Deliverable Nr. D3.10.2, Laurin Project, Dipartimento di Informatica e Sistemistica and CM Sistemi S.p.A., 1999.

[3] D. Calvanese, T. Catarci, and G. Santucci. Building a digital library of newspaper clippings: The LAURIN project. In *Proc. of the IEEE Forum on Research and Technology Advances in Digital Libraries (ADL 2000)*, pages 15–26, 2000.

[4] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[5] J. W. Cooper and R. J. Byrd. Lexical navigation: Visually prompted query expansion and refinement. In *Proc. of the 2nd ACM Int. Conf. on Digital Libraries (DL'97)*, pages 237–246, 1997.

[6] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. In G. Brewka, editor, *Principles of Knowledge Representation*, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.

[7] J. Frew, M. Freeston, R. B. Kemp, J. Simpson, T. Smith, A. Wells, and Q. Zheng. The Alexandria digital library testbed. *D-Lib Magazine*, July 1996.

[8] T. S. D. L. Group. The stanford digital library project. *Communications of the ACM*, 38:59–60, 1995.

[9] I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.

[10] L. Kovács, A. Micsik, and B. Pataki. AQUA: Query visualization for the NCSTRL digital library. In *Proc. of the 4th ACM Conf. on Digital Libraries (DL'99)*, pages 230–231, 1999.

[11] E.-P. Lim and Y. Lu. HARP: A distributed query system for legacy public libraries and structured databases. *ACM Trans. on Information Systems*, 17(3):291–319, 1999.

[12] D. Norman and S. Draper. *User Centered System Design*. LEA, Hillsdale, N.J., 1986.

[13] J. Ober. The california digital library. *D-Lib Magazine*, Mar. 1999.

[14] V. Ogle and R. Wilensky. Testbed development for the Berkeley digital library project. *D-Lib Magazine*, July 1996.

[15] A. Peterson Bishop. Measuring access, use, and success in digital libraries. *The Journal of Electronic Publishing*, 4(2), 1998.

[16] B. Schatz and H. Chen, editors. *Digital Libraries: Technological Advances and Social Impacts*. Feb. 1999. Special Issue of IEEE Computer.

[17] N. A. Van House, M. H. Butler, V. Ogle, and L. Schiff. User-centered iterative design for digital libraries: The Cypress experience. *D-Lib Magazine*, Feb. 1996.

[18] R. Williams and B. Sears. A high performance active digital library. *Parallel Computing, Special Issue on Metacomputing*, 24(12–13):1791–1806, 1998.

[19] W. A. Woods. Understanding subsumption and taxomony: A framework for progress. In J. F. Sowa, editor, *Principles of Semantic Networks*, pages 45–94. Morgan Kaufmann, Los Altos, 1991.

[20] Z39.50 Maintenance Agency. *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50-1995)*, july 1995.