A SEMANTIC WEB BASED APPROACH TO MULTIMEDIA RETRIEVAL

Gorka Marcos[1], Tim Smithers[1], Iván Jiménez[1], Jorge Posada[1], André Stork[2†], Massimiliano Pianciamore[3], Rui Castro[4], Sergio Di Marca[6], Marco Mauri[3], Paolo Selvini[3], Neyir Sevilmis[2], Bruno Thelen[5], and Vincenzo Zechino[6]

[1] VICOMTech, Donostia / San Sebastián
[2] Fraunhofer Institut für Graphische Datenverarbeitung, Darmstadt
[3] CEFRIEL, Milan
[4] Centro de Computação Gráfica, Guimarães
[5] Schenck Pegasus GmbH, Darmstadt
[6] Italdesign – Giugiaro SpA, Moncalieri (Torino)
†Author to whom all correspondence should be sent to at ist-wide@igd.fhg.de

ABSTRACT

Multimedia information retrieval is vital for, among many other fields, design and engineering processes. Studies show that members of product development teams spend more and more time for information gathering and have less and less time for the creative aspects of their work.

Here the combination of the still emerging Semantic Web [1] technologies and information sources with content-based indexed multimedia offers new possibilities.

This paper presents one of the first approaches to tackle the retrieval of indexed multimedia keeping the semantics all over the steps of the process. The designed architecture and the implemented syste allows to take advantage of the information sources with different levels of semantic richness, which we believe will be part of the Semantic Web in the near future, making the "knowledge about the indexed multimedia information" available to the user and the computer

Furthermore, this approach has been developed in the context of real scenarios provided by two companies from the Car Design and Engineering industry, and the conclusions of this work will be also summarized.

1. INTRODUCTION

Nowadays, the management of the larges amounts of multimedia content generated everyday in most of the companies has become not only an urgent need but also a success feature. Commendable and interesting efforts are being made in the field of content based information retrieval, developing and implementing algorithms and techniques that allow automatic o semiautomatic content indexing [2]. Companies are aiming at integrating these management systems in their workflow. However, the implementation process in different enterprises implies new difficulties due to the intrinsic peculiarities of each company.

Therefore, this paper summaries the results of an IST project funded by the European Commission named WIDE (IST-2001-34417) [3], which was encouraged by two companies of the area of Car Design and Engineering ItalDesign-Giugiaro (a company that designs cars from the scratch till the prototype phase) and Schenck-Pegasus GmbH (a company focused on the car engine testing and developing).

The main motivation of these companies is the minimization of the amount of time invested by their engineers and designers in searching and retrieving appropriate multimedia information, which includes a wide set of types of content, as for example, pictures, sketches and high-resolution photos of cars, patents about engine components, 3D models about the cars, regulation about emission standards and so on [4].

The knowledge engineers (KE) of both companies, responsible for guiding and supporting their colleagues finding the required information, estimate that some of the engineers invest up to 80% of their time looking for the appropriate information.

The system developed within the WIDE project is based on a rich combination of semantic techniques allowing the user browsing and retrieving information from several multimedia information sources. It uses not only of the semantics of the indexed content of some information sources, but also from a Domain Ontology [5] that has been modeled in one of the layers that compose the architecture of WIDE in order to:

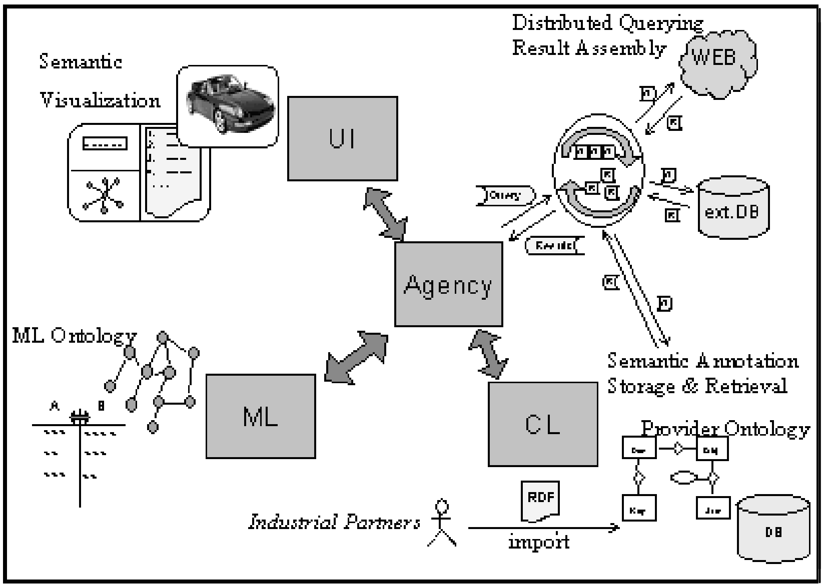

Figure 1: WIDE Architecture

- Allow more accurate and efficient retrieval mechanisms. Although the system is able to query different types of information sources (i.e. unstructured as Google, relational as some of the proprietary Information Sources of the involved companies and repositories aware of the semantics of their content), the retrieval process can be refined when the content is indexed.
- Implement graphical facilities for content and domain browsing and result visualization, allowing the use of the known conceptualization of the domain.
- Handle different kinds of users, such as engineers and designers from both companies working on the same content, who have different terminology and needs depending on their profile.
- Provide the user with query development facilities. The system allows the user to browse a conceptual and intuitive representation of the domain and to use this representation to define a query.

This paper is structured as follows: first of all, the WIDE architecture is described, explaining briefly the role of its components.

Section 3 explains the search paradigm that has been designed and developed in the project in order to keep the semantics all over the search process, putting special attention on the domain and results browsing steps and on the results extraction from the content based indexed multimedia repositories. Thus, this section outlines our semantic web based approach for multimedia retrieval.

Finally some of the results of the evaluation sessions are presented and the conclusions based on the work carried out in the project are summarized

2. WIDE ARCHITECTURE

The aim of this section is to describe the way the WIDE system has been designed to support more effective knowledge-sharing and semantically enhanced multimedia information retrieval. The following sections outline the mechanisms that have been developed.. Figure 1 shows the main components of the WIDE Semantic based Information System, and how they are related.

2.1. User Interface (UI)

The User Interface provides a graphical front end to the user. This interface is used to provide the user with information about the domain of the search (see Figure 2), to support incremental development of a user query, and to present the obtained results.

2.2. Meta Level (ML)

This subsystem of WIDE supports the design and subsequent semantic processing of a user query into so called system queries. Thus, the ML uses a domain ontology written in OWL [6] including more than 1000 concepts representing the main concepts and relationships that shape the domain and the knowledge about different user terminologies and tasks carried out within the organizations (engine adaptation, looking for creative inspiration, etc). The different types of knowledge according to the profile of the users are used

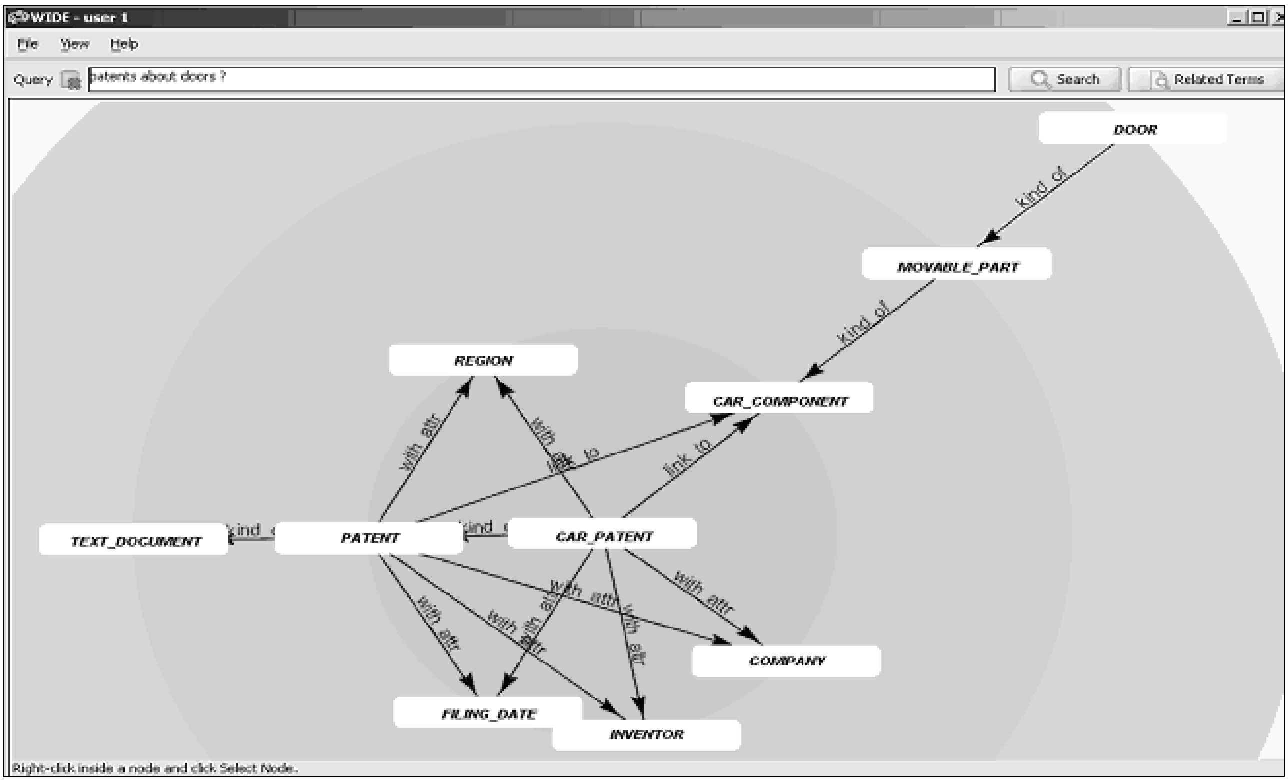

Figure 2 WIDE User Interface

to produce the queries that are then transferred to the Agency (section 2.3).

The returned results and the user queries undergo a similar semantic processing. Both of them are semantically processed in order to associate them with the appropriate concepts of the domain ontology and are finally displayed in a meaningful graph structure by the UI.

2.3. Agency

The Agency subsystem identifies and locates information sources in the Content Level (section 2.4) to which the system queries can be sent to produce effective returns. The Agency also provides the gateway of the system to the Web, which is also considered part of the Content Level. Web sites and Web search engines are treated as weakly structured information sources.

Besides the planning and execution of queries, the Agency is also responsible for collecting and transforming the results of the heterogeneous information sources into a common result format on which the ML performs semantic processing and reasoning.

2.4. Content Level (CL)

The CL includes different information sources (ASAM/ODS [7], semantic information sources built using RDF [8], relational databases and the web), which vary in their semantic richness. The main purpose of the CL is to answer precisely the system queries in a quick way.

Although the combination of the Agency and the CL is able to handle different Information Sources, the Semantic Information Sources are enhanced and its role in order to provide more effective retrieval from multimedia repositories where the content of each "media unit" has been determined in advance.

But how will these Semantic Information sources look like in the forthcoming Semantic Web? From a data-centric view, for us a Semantic Information Source contains the following layers: Schema, annotation (Meta Data) and Content (as shown in Figure 3).

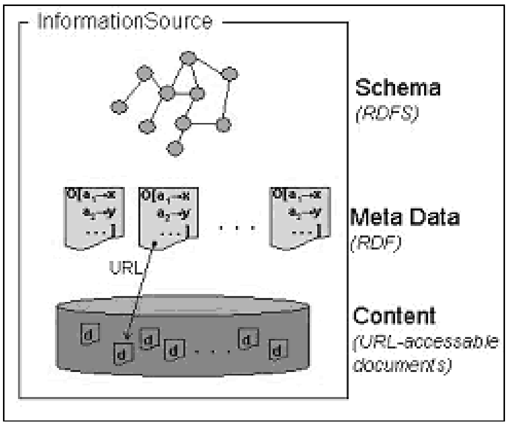

Figure 3: Data-centric view to a Semantic Information Source

On the top-most layer, a conceptual data schema describes the content that is stored by abstract entities and their relations. This schema has been implemented as the provider ontology of the Semantic Information Source. In the middle layer, the abstract entities are instantiated in interlinked metadata annotation objects that, in turn, refer to the actual content items on the bottom layer.

The content include different types of multi-media documents such as pictures, text, and 3D models. The user is interested in retrieving these content items (instances). All the metadata on the schema and annotation level are used to semantically describe the content and allow for precise and accurate retrieval. The metadata objects in the middle layer appear as instances of classes while the content items appear as references to URL-accessible stores or lower-level database access components. It should me mentioned that both the annotation and the relations between the content and the middle layer are done manually by the Knowledge Engineer.

Moreover, the Semantic Information Source does not contain full domain ontology nor does it represent a knowledge base from our point of view. Instead, the Semantic Information Source models the 'aboutness' of documents/information contained in the lowest level. The schema only contains what is needed to appropriately describe the types of documents/information contained in the lowest level and about which real world concept they are talking about.

From a functional point of view, a Semantic Information Source should be able to process a query posed in a standard format, RQL within the WIDE project [9]. In this context processing comprises, mapping it in a syntactic and semantic way.

Furthermore, it provides the results of the query in a standard form, preferable enriched by semantic information/context for further semantic processing, e.g. reasoning on the results for filtering and ranking them before showing them to the user. We are using the W3C suggestion for RDF result representation returning not only the results but also structured context information [10].

3. WIDE: A SEMANTIC-WEB BASED APPROACH TO MULTIMEDIA RETRIEVAL

In this section, the classical model of information retrieval is summarized, and after that the WIDE model of multimedia information retrieval is presented. It has been developed to better support effective and efficient information retrieval and re-use by users who do not have strong prior knowledge of the organization and structure of the information sources they need to access. These improvements are mainly based on the existence of semantic knowledge not only about the domain but also about the content.

The presentation of the WIDE model is made not only from a theoretical point of view, but also outlining some of the most relevant internal processes carried out by the system.

3.1. The Classical Model of Information Retrieval

The classical, or Google model of information retrieval is composed of a linear series of five basic steps (see Figure 4).

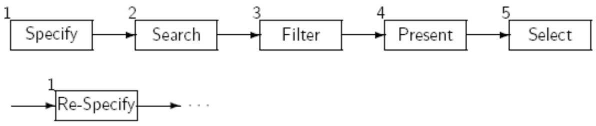

Figure 4: Classical Model of Information Retrieval

In the first step, the user specifies using some input language or Natural language-like search specification. Google, for example, offers Natural Language-like input, together with a more sophisticated specification interface: the Advanced Search option. No matter which input option is used, the important point to note here is that what the user inputs forms a search query.

The second step is to perform the actual search using the specified query. The way this is actually done, and what it involves differs between systems, and can be more or less efficient. Nonetheless, all information systems must perform this step in some way or other.

The third step is to filter the results, and/or remove some of the returned search results. This filtering is done by scoring the returned results with respect to the specified search query, and then using the score of each result to decide in which order it should be presented to the user, and/or if it should be presented at all. (Not all information systems perform this step, but most Web-based information systems do perform some form of filtering).

Step four involves presenting the filtered results to the user. This can be as simple as presenting an ordered list, to something more sophisticated, such as some form of graphical (graph-based or abstract) presentation. The details of how this presentation step is performed depend mainly on what kind of search results are returned, which in turn depends upon what sources are searched.

In the final step, the user selects those results that he or she decides are useful from the presented search results. The outcome of this selection step depends upon several factors: how the original search was specified (in step 1); how the search was performed (step 2); how the returned search results were filtered (step 3); on how the filtered results are presented; and on the selection criteria used and applied by the user. Changes in anyone of these steps can have a direct impact on the outcome of step 5.

One outcome of step 5 can be that the user decides that none or not enough of the presented search results are useful. This can, in turn, lead to a new search being specified, and so to a repeat of the same linear sequence of steps. It is important to note here that each repetition of this sequence of five basic steps in only related to any previous sequences in the mind of user: the information system does not use any information generated during one sequence to help or influence any subsequence repetitions; it uses no memory of any previous searches.

This classical model of information search can thus be understood as a linear single-shot repeatable process.

3.2. A Critique of the Classical Model

The Classical Model of information search outlined above is so widely used in all sorts of information systems, and so widely experienced by all sorts of users, that it might seem hard to identify any difficulties or problems with it. However, there are a number of important criticisms that can be made of this model, and by implication, all information systems that implement it.

On the one hand, this model does not adequately distinguish between the needs of a user and what a user must specify to get it, or to try to get it. Usually, users know well what they want or need, but they may not know how to specify a good search query, even in Natural Language terms. The failure lays on trying to distinguish between what a user wants, and how the user needs to ask for it. In particular, specifying what is required to satisfy some stated need usually requires knowledge of what is available. In other words, specifying a good search does not just depend on knowing the need; but on knowing what to search for, and how it is organized.

The knowledge about what there is to search for is often gained by repeated executions of the Classical Model. Analyzing what is retrieved from the first attempt is used not so much to select useful results, as to find out what is there to be search over.

A second important criticism of the Classical Model of information search is that any knowledge generated during the process of formulation a query (step 1) is not used later on in the sequence, to influence the filtering (step 3) and presenting (step 4) of the search results, or to select the results (step 5). This loss or non-use of the knowledge about the need of the user generated during the specification step- knowledge results in a serious loss of effectiveness and efficiency in the information search process as a whole. Knowledge of how the search specification is built from the user needs usually reflects important aspects of the way the user currently understands his or her information needs.

Moreover, each specification, search, filter, presentation, and selection sequence is treated independently from any previous attempts. There is no possibility of using the knowledge and experience gained as a result of any previous searching in order to improve the information search, or help to work out what needs to be searched for in the first place.

Finally, this Classical Model provides an essentially context-free process. There is no proper way in which knowledge of the task context and situation and user profile can be effectively brought to bear on the information search process. Thus, it is left to the user to work out and learn how to use any search specification options offered by a system to effectively encode such aspects.

3.3. The WIDE Model of Information Retrieval: theory and real implementation

In an attempt to address these criticisms of the Classical Model of information retrieval, the WIDE Model of information retrieval treats the general task of information finding as a kind of design task, and not as a kind of search specification and results selection tasks.

Information retrieval is understood as a kind of design task by first recognizing the difference between users stating needs and forming well specified requirements, and then properly supporting the incremental development of a complete and consistent requirements specification (search specification, in this case), and the re-use of the knowledge generated in this (sub) process to effectively support the subsequent steps in the process that concludes in a useful set of search results.

The WIDE model of information retrieval is thus described in the figure 5

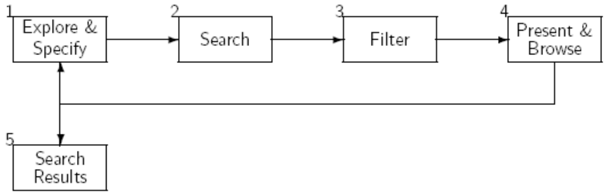

Figure 5: WIDE Model of Information Retrieval

3.3.1. Explore and Specify

The first step is the (sub) process that supports a user in developing a well-formed search specification in the context of his or her information needs. This is described as being exploration, and not just specification, because it is at this step that a user needs to discover what might form possibly good search specifications. Note that in a complex field, as Car Engineering and Design, finding out which kind of information (multimedia) is needed at every moment and how it should be retrieved is not a simple issue.

This often involves discovering relations and related elements that can be usefully included in a search specification, but which sere not initially identified by the user. Effective support of a user in this step requires effective use of knowledge of the domain or domains, user terminology or terminologies, task context and kind of user.

Within the WIDE project, this step is implemented by the combination of the graphical facilities of the UI and the knowledge of the ML has carried out by the user, not only about the domain but also about the different terminologies of the various user communities and the main concepts involved in the task (which has been selected at log-in time).

This is shown in Figure 1, where the system shows the appropriate information about the sub domain of interest of the user, who is trying to find out the relationships among the patents and the doors. The nodes of that graph can be dragged and dropped in the query bar in order to create an appropriate query. According to this, if the user drags the node “doors” and drops it in the query bar close to the word “patents”, the system automatically refines the query to get “patents about doors”.

3.3.2. Search

The second search step is essentially the same as in the Classical Model, and can be implemented by Classical information systems, which form components of a WIDE system.

Concerning the concrete implementation of this search step in the WIDE system, three main (sub) steps are distinguished: semantic analysis of the user query to create a set of queries named system queries, distribution and adaptation of these system queries to the different information, and the results retrieval.

Once the user decides to submit the query, it is forwarded by the Agency to the ML. First of all, the ML uses its knowledge about the different terminologies in order to translate the query into the internal terminology. Then, this query goes through a parser based on a BNF grammar [11], which is used to analyze the structure of the query. The output of this parser is an Abstract Syntax Tree (AST) that includes useful information that is the basis for further semantic analysis.

This information is related to the role of each term of the query. For instance, the term x_1 of the sentence is qualifying the following term of the sentence, the term x_2 is a Boolean operator that joins two pars of the query, the term x_3 is the main concept or target of the query and the rest of the terms define some search constraints and so on.

The ML goes through this tree and infers the Domain ontology in order to find out the existing relationships among the different terms of the tree. As a result of this process, several expansions can be carried out, resulting in a set of system queries.

For example, a user query “Pictures about Jaguars” generates, among others, the following system query: “Pictures of cars with brand = Jaguar”. This is achieved due to the following “asserts” known by the ML: ”Jaguar is not only a kind of animal bur also a kind of car brand”, “in the task carried out by the user (i.e. Product Flyer Development) car is one of the relevant concepts involved” and “every car has a brand”.

The last step before sending the system queries to the Agency is to translate them into RQL language [9], which has been chosen as the query exchange language. Following with the Jaguars example, the RQL query produced would look like follows:

```
SELECT   pt, mc
FROM     {pt:$pt} @p {mc:$mc},{rc1} @w_a1{c1:$c1},
         {rc2} @w_v1 {v1:Literal}
WHERE @p = “has_info_about”AND
        ($p1 = “PICTURE”) AND $mc = “CAR” AND
        mc = rc1 AND @w_a1 = “with_attr” AND
        $c1 = “BRAND” AND c1 = rc2 AND
        @w_v1 = “with_value” AND v1 = “JAGUAR”
```

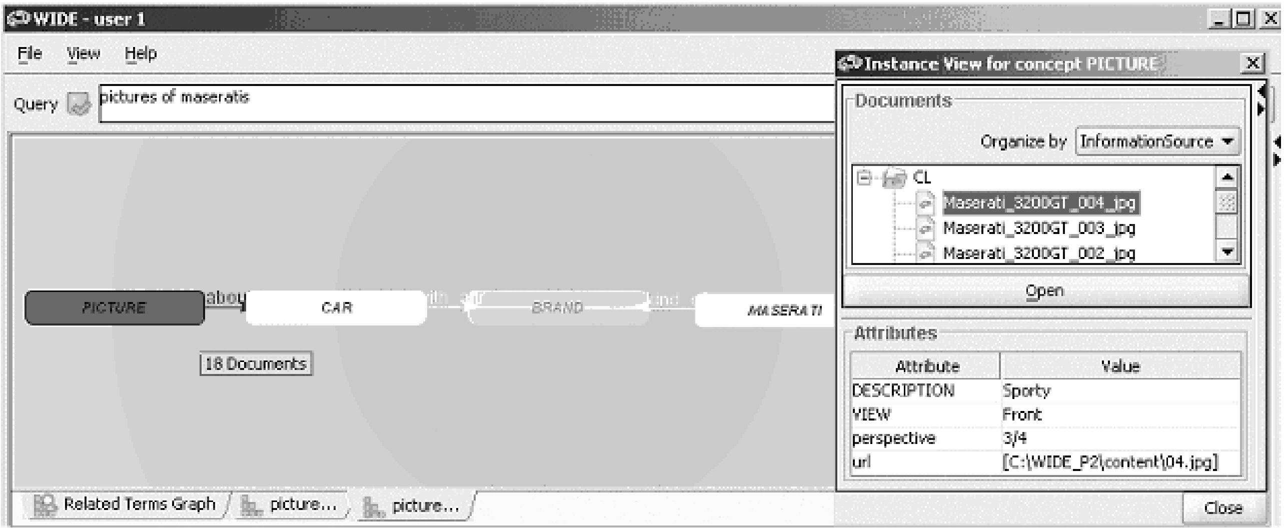

Figure 6: WIDE Result Visualization Interface

As soon as the Agency subsystem receives the queries generated by the ML, they are sent to the various information sources in order to proceed with the search process. The decision about how to distribute the various queries over the available sources is referred to as query execution planning and can be carried out by analyzing the structure of each query. Concerning the structure of a system query expressed in the RQL language, it can easily be seen that the FROM clause can be interpreted as the navigation path inside a proper ontology of concepts bound to one another by means of suitable relationships.

The Agency adopts a two-stage process. It first sends the System Queries to all the information sources in the Content Level. Each information source (or its representative agent named provider agent) determines which part of the full SQ it can try to match. In order to do that, some information sources are able to carry out some terminological mapping (using local dictionaries) and even structural mapping (using rules to define replaceable patterns). With this information, the Agency decides on which (sub) set of information sources to ask. It then collects all the returns, from the various information sources, and returns them to the ML.

The format chosen to exchange the retrieved results among the subsystem is an extension of a proposal discussed at the W3C consortium as a standard for query result formatting [10]. This format allows exchanging not only the result (the multimedia documents) but also the extra knowledge stored in the Information Source.

For instance, an enhanced result could consist of a picture saying that the content of that picture is a car, with a brand that equals Maserati and with a concrete color and so on. Thus, the usage of this format allows taking advantage of the content based indexed multimedia not only in the search step, but also in the following ones.

3.3.3. Filter

The filter step in the WIDE approach is once again similar to step three in the Classical Model, except that more data and knowledge can be brought to bear on the filtering: knowledge gained during step one, together with knowledge about the task carried out by the user and previous search cycles.

Similarly, the way filtered results are presented to the user makes explicit the use of the knowledge generated during the Explore and Specify step and the knowledge extracted from the content based indexed results. In particular, the structure of the relationships in step one, between the need of the user and the search specification, can be used to usefully inform the structure and organization of the presentation.

This is important because the structure of the relationship between the user needs and his/her subsequent search specification is something demanded by the user and his or her subsequent search specifications something that the user is actively engaged in constructing. Thus, the user has a personal and intimate knowledge of this structure. Using this to inform the presentation of the results thus allows the user to re-use a kind of knowledge he or she already has from step one, rather than some more context-free or impersonal or mysterious kind of knowledge, as we seen in Google, for example.

In the WIDE system, the ML is responsible for this task. This task is carried out combining the knowledge of the domain with the results retrieved by the Agency. Thus, the ML analysis the results, trying to find out what they represent, the different relationships among them and raking them according to their relevance. Once again, the role of the ML Domain Ontology together with the knowledge of the user and the task he is carrying out is decisive. The concepts involved in the query developed by the user in the first step of the search are also decisive.

3.3.4. Present and Browse

Step four, in this WIDE model, is called Present and Browse because it is the subsequent browsing of the results presentation that supports further exploration of how the search specification might be useful further developed to better meet his or her information needs. In this way, from the user perspective, step four effectively merges with step one to form what can be understood as a kind of design process. The implementation of this step in the WIDE system is similar to the first step, since the system provides the user with a graph that interrelates the most relevant concepts involved in the query. Besides this, the (multimedia) results are attached to the different nodes of the graph, depending on what they represent.

This graph can be also used to refine the query using the same drag and drop mechanisms described in the first step.

As can be seen in the Figure 6, the nodes of the graph that contain results are colored in a different way. Clicking on one of those nodes, a new window is shown with the list of results and that shows the metadata of each result.

4. USER EVALUATION

At the moment this paper was written, the final evaluation period had not finished. However some preliminary comments together with conclusions from previous evaluations are complied here.

First of all, we summarize some of the featured the user liked:

- The possibility to query different information sources with a unique interface.
- The query expressiveness: they possibility of typing queries in an unstructured (NL alike) way.
- The concept based result-browsing facilities.
- The possibility of browsing the domain of interest and use it to query development.

Secondly some proposals for improvement are highlighted:

- Need of node delete or filtering mechanisms when the size of the graphs increases.
- Improve the time performance for complex queries.

5. CONCLUSIONS

We have presented a system which architecture allows a new information retrieval semantic-centric approach. According to this, the system is able to use the knowledge of the Meta Level and of the information sources with content-based indexed results in order to:

- transmit this knowledge to the user,
- allow him to interact with the domain,
- support him in the query development and
- presenting him a concept-based enhanced visualization of the results.

We have also presented our interpretation of those semantic web information sources and its role in the information retrieval process.

6. BIBLIOGRAPHY

[1] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American*, pp 34–43, May 2001.

[2] “Content-based indexing of multimedia databases”, Knowledge and Data Engineering, IEEE Transactions on, Volume: 9, Issue 6, pages 978-989. Nov.-Dec 1997.

[3] WIDE Project HomePage < www.ist-wide.info >

[4] T. H. Davenport and L. Prusak, “Working Knowledge: How Organizations Manage what they Know,” Cambridge, MA: Harvard Business School Press, 1998.

[5] S. Staab and R. Studer, (Eds.), “Handbook on Ontologies,” Springer Verlag, 2004

[6] OWL Web Ontology Language Overview, W3C Technical Reports and Publications, <http://www.w3.org/TR/owl-features/>.

[7] Association for Standarisation of Automation and Measuring Syetems (ASAM)Open Data Service (ODS), <http://www.asam.net/01 asam-ev 01.php>.

[8] RDF Vocabulary Description Langauge 1.0: RDF Scheme, W3C Technical Reports and Publications, <http://www.w3.org/TR/rdf-schema/>.

[9] The RDF Query Langauge (RQL), FORTH Institute of Computer Science, <http://athena.ics.forth.gr:9090/RDF/RQL/>.

[10] A. Seaborne, Recording Query Results, W3C Discussion document http://www.w3.org/2003/03/rdfqr-tests/recording-query-results.html

[11] Backus-Naur form (BNF), WIKIPEDIA, <http://en.wikipedia.org/wiki/Backus-Naur Form>.

[12] T. Smithers and W. O. Troxell, “Design is intelligent behaviour, but what is formalism?”, AI EIDAM, Vol. 4, Nº 2, pp. 89-98, 1990.