# Interaction and navigation for a document database : a concrete case study

Isabelle Berrien, François Laburthe and Jean-David Ruvini

e-lab BOUYGUES SA
1, avenue Eugène Freyssinet,
78061 Saint Quentin en Yvelines, FRANCE
Mail :iberrien@bouygues.com
Tél : +33 1 30 60 53 66  Fax : +33 1 30 60 22 15

## Summary

In this article, we present the application Wishbone, an innovative search engine to explore complex document database. Dedicated to the intranet document site of our society, we present in this paper how we managed to improve the request process by avoiding the "all or nothing" syndrome for a too fuzzy or too demanding request respectively, offering also the maximum of smoothness in the way the user formulates the query. We particularly focused our attention on the interaction between a web database and its users, either during a consultation or an interrogation process. This web context can be found in multiple environments like an intranet, a FAQs site, self-assistance, sale catalogues or document data bases. First we analyse document consultation sites in regard to quality criteria. In a second step, we describe our approach to furnish a software response to the problematic of settling, management and interaction of an online document database. All this work was accomplished using as example an enterprise document database with various documents like contracts, business notes and bug reports.

**Keywords : document database, web mining, research engine, knowledge management**

# I. Introduction

Surfing in a content site appears to be one of the major Internet's problematics encountered these days with search engines. Therefore it has become one of the striking research axes about the World Wide Web, which not only has to face increasing amounts of pages but also is supposed to render the information they contain inside easy to read and understand. It is a huge problem since the multiple factors which are linked to it seem quite complex: different information sources, multiple publishers, site architecture, request systems… We therefore asked to ourselves the following question:

**How can we help an internaut these days while searching information or investigating a data base using the internet?**

Part of our research topics is concerned by this problem and this paper presented here discloses our first results.

In order to both answer the question and propose a concrete solution, we chose to investigate our research program in an enterprise intranet context. The data sources we got in hand could be therefore better identified. We can as a consequence narrow the field of our work to "Intranet Mining" (*vs.* "Web Mining"), which presents the following advantages. :

- **Intranet** : the problematics encountered in an intranet are quite identical to those of the world wide web but in a restricted context.
- **Data are easy to access** and smaller. It is a context well appropriate to conceive, develop and validate components which should be *in fine* exploited in the World Wide Web.

This article in made of three main parts. First we describe the problematic of document site we got interested in. In the second part, we explain in detail the issue we bring to it by studying the difficulties faced by the **C2S** enterprise about its document database and the last part presents our future work.

# II. Document database consultation problematics

In order to clearly understand the problems encountered with the management of a document database and orientate our investigation properly, we first identify the main difficulties the user has to face when she connects to such a site. We briefly point out the main criteria which make a document site appealing, then we focus our attention to the complexity of such sites.

## 2.1. Quality measurement of a site

Rating a site quality notion is quite fuzzy and quite complex (aesthetic aspect, ergonomics content quality).

Nevertheless, our bad experience of frustrated internaut when not finding what we want makes us feel that a web site reaches its goal[1] when a user gets easily the advice or the information she's searching for.

This could be measured according to the following criteria:

Objective :

- percentage of times a user gets an answer
- number of clicks required to access the answer

Subjective/qualitative :

- easiness for the user to understand the database logic (does she get lost pretty early in the process or not ?)
- answer pertinence according to the user's expectation

We have now in hands some quantifiers to estimate the quality of a documentary database access. This will be our evaluation protocol for every site concerned.

## 2.2 Complexity factors

An information site must implement several different components like multiple choice forms with depending on the context, preceding selections, surfing access to side. This implies a continuous user interface during the process taking place between the site and the user. This process successively switches from questions, choices, fields of values presentation on one side and on the other side answers, selection,

[1] http://www.auditweb.net/

criteria. This also requires having a good capability of managing the process logic while the interface keeps on changing its content (apparition and disappearance of criteria, values changes…).

Le Grand has identified in her PhD. thesis three major difficulties:

- **Organization level of the documents:** in the user mind, the desired information takes usually part of a structured frame. Invoking a rich document organization does not refer directly to the general descriptors of the documents, but rather to the way the documents are related to each other. We are talking about specific descriptors depending on the document type which allow the information to be classified harmoniously in a hierarchy. The purpose here is to offer an intuitive access to the information.
- **The volume of information feed back:** quantity does not guarantee quality. A system that has plenty of data is not necessarily a good service. On the contrary, it often procures the user an uncomfortable feeling because of incapacity for her to get an overall view of the environment.
- **Richness of the documents content:** this time, we're really talking about the document itself. The more the document is annotated, classified, the more the final interaction is pertinent and smooth. This allows, moreover it implies, that the research data (*e.g.* documents descriptors) should be rich, multidimensional, giving a user the possibility to navigate or express her request with maximum of smoothness and comfort. Richness of content concerns the data production phases.

From those three points, we deduce that the route between weakly structured, heterogeneous data sources and its consultation on line can be long and complex. We can qualify this as an "information cycle", which equilibrium is fragile. Next we detail our solutions.

# III. WISHBONE : a tool for the publishing and the consultation of documents.

The solution we propose, the Wishbone application [Berrien and Laburthe, 2003] is cut in three parts :

- **1. A research and exploration service** capable of displaying overall views of solutions if there are too many, or re-direct the user in case of no response. In summary, we prevent the user to fall into the "all-or-nothing" scenario which often forces her to click back during the browsing process (*e.g.* each time the request does not imply a small amount of responses).
- **2. A research model detached from the real data storing**. To each real document is associated a descriptor object. These objects are organized following a rich model and detailed for numerous possible types of documents. It has many advantages like a higher flexibility (for it allows data to be stored in a business model different from the research model used for the query process), and also enriched investigation techniques. The model permits to define a service structured like a product configuration component, completely disjoined from the document nature (classes/types/options/containers/associated objects...).
- **3. A help to the data production.** First of all, we must have in hand rich and complete data, even after a lazy publication process. We also need to prevent any storage of absurd information. Thinking about perfect automatic data storage does more look like a utopia, but it still should be feasible to narrow the amount of errors generated during the process. This could be done with the help of assistance mechanisms for document publishing.

Wishbone has been settled in an intranet frame dedicated to the management of documents published by a computer branch society of BOUYGUES Company (C2S) which covers all problematics mentioned upside. The **C2S** company is a hundred people staff society[2], with a flux of thousands documents published per year. These documents concern a large part of the company patrimony (checking flow of contracts, product description documents, marketing papers, technical points, internal procedures…).

In the following section, we describe each of the 3 preceding points.

[2] http://www.c2s.fr/

### 3.1. Research and exploration service: using views or relaxations to browse

As it was upper mentioned, a important criterion for a successful document database site is to be able to generate interrogation and response screens well adapted to the user (external or professional). We must keep, whatever the degree of advancement of the research process is, a friendly and clear interface whereas it could easily turn into a deep complex one if the information volume gets too big. Our key rule was to accompany the internaut by giving her views of documents groups instead of a huge list.

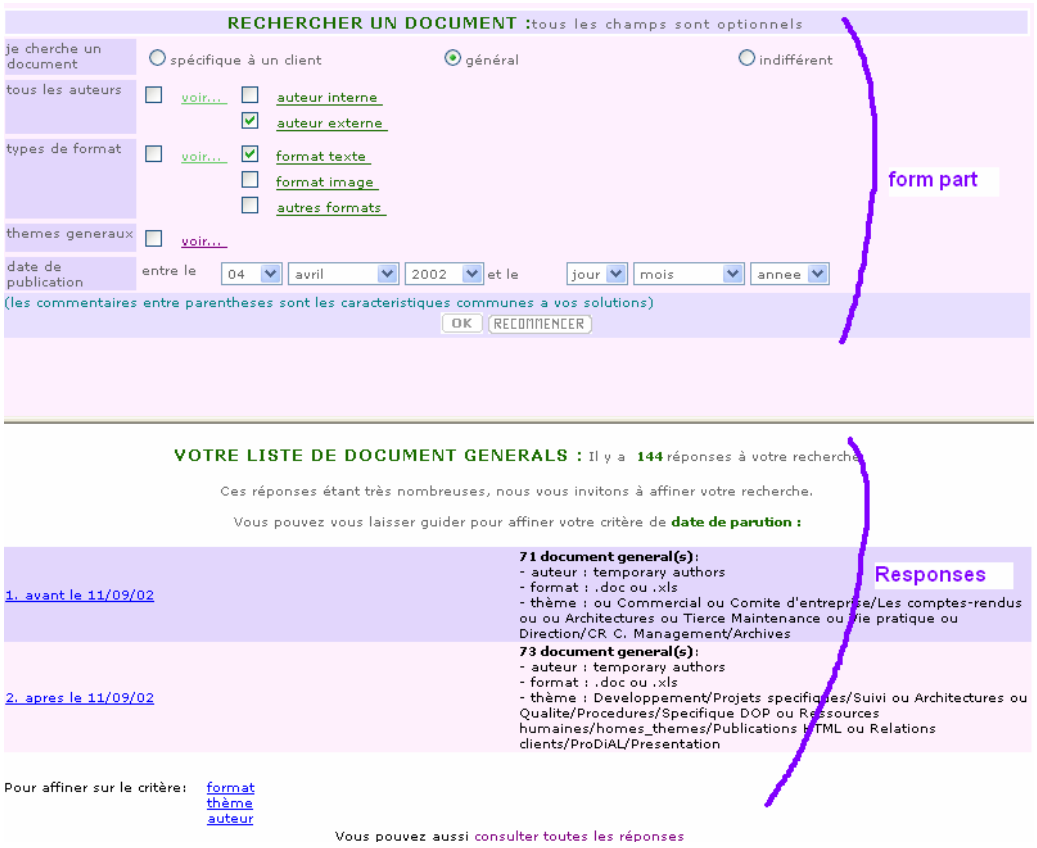
#### 3.1.1 Too many answers

When answers are too many to be decently displayed and easily understood by the user, the Wishbone platform developed in our laboratory presents the solution set grouped in several packages. As example, the picture below shows a typical interface of our application (**Figure 1**).

The request in the case below is the following :

"**General documents, written by external authors, of text format and published after april 4, 2002**".

We distinguish two parts, the form (up) and the responses (below):



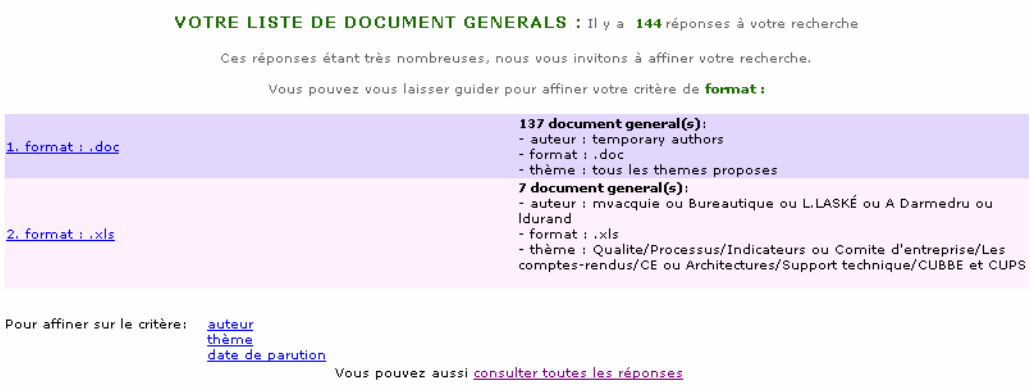

**- Figure 1 -**

The query solution set contains 144 answers. Instead of forcing the user to refine her request in order to reduce the number of solutions, or displaying all solutions, the application returns an image of all answers classified into groups depending a chosen criterion.

The engine grouped them in 2 packages chronologically classified (published before and after September 11, 2002). We notice that each cluster has been enriched with common values for fields like the author, the format and the theme). This is the result of a segmentation performed by the engine upon all solutions (according to a criterion, either chosen by the engine as the most performing one from a clusterisation point of view (default behaviour), or either by the user if she wants another view).

Moreover, in each group, if all documents show one or more common characteristics, (for instance here, all documents published before September the 11$^{th}$ are xls or doc files), these are disclosed (**Figure 1**). This takes part of a enhanced converging process: the user is informed of the most relevant characteristics of each cluster, a sort of "one step in advance", thus allowing her to choose the most appropriate one in the less hazardous way.

If the current filter criterion doesn't satisfy the user, she can switch to another one as she wants (here below the solution set was switched to a *format* point of view) (**Figure 2**):




**- Figure 2 -**

By clearly grouping the answers and attributing each group pertinent characteristics, while allowing the user to see the answers from the desired point of view, we obviously improve the user productivity in accelerating the converging process. Even if use of different views for the same data is a well-known method in databases, allowing access to numerous data using a monodimensional clustering process has never been reported before. This therefore brings a solution to the second problem mentioned in the beginning, that is to say how to manage a large amount of information in the most ergonomic manner.

### 3.1.2. No answer

In case of there is no solution to the request, instead of falling into the usual scenario of forcing the user to re-formulate her request, the engine proposes alternatives by displaying the closest solution spaces corresponding to the initial request. For instance, the request in the following example is "**find documents published by an external author and with a image format**". This present request has an empty solution set, but still the user gets the following response (**Figure 3**):


**- Figure 3 -**

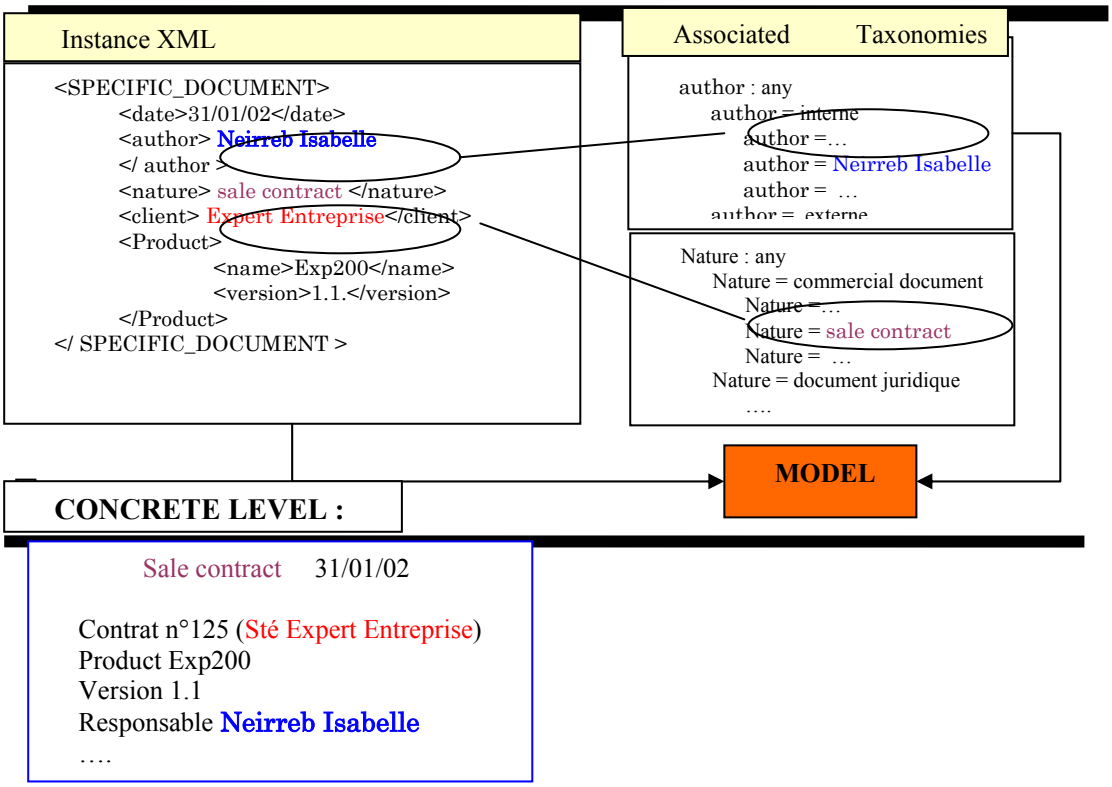
This example shows that the two criteria (*e.g.* **external author** and **image format**) could not be both kept in order to get a response. The engine smoothly relaxed each of them until it gets valid solution spaces [Laburthe, 2002]. The user is then proposed 2 alternatives: the first one retains the external author request and looks in the closest values for the format field (**Word file instead of image**) and therefore can choose the "137 solutions" set. The second proposition kept the format value request (image) but had to extend the author field initial value (initially "external author"). We herein find out that quite a large part of the image documents has its author field value not filled!

### 3.2 A research model

The application relies on a two level description (**Figure 4**).

**The first level** (the abstract one) concerns descriptive objects of documents. This is the level in which the research application acts and is responsible for the user dialog logic. We organize this level in concepts, classes and relations.

**The second level** (the concrete one) deals about the documents themselves, and is taken as reference for all the classes' instances which constitute the whole object database.




**- Figure 4 -**

The **Figure 4** illustrates the organization abstract/concrete for a particular document (here a sale contract). This sale contract document is linked in the abstract level to an XML described instance (label on left part), of which certain elements take their values in an appropriate ontology [Gruver, 1993] according to the object model.

In this example, we have an instance of the SPECIFIC_DOCUMENT class. The SPECIFIC_DOCUMENT class in our model is a subclass of the DOCUMENT class and represents documents containing specific information like the reference to a client (a regular instance of the DOCUMENT does not). Here it refers to a client, "Expert Entreprise", the author is Isabel Neirreb, and it's got a reference to a product (PRODUCT class) which contains itself 2 slots: its name and its version.

At this point, this organization is similar to the one of Topic Maps[3] [XTM Authoring Group, 2001]. In effect, in both cases we have a 2 layers separation, one

[3] The Topic Maps concepts relies on:

being abstract and the other being concrete. The couple ("topic", "association") could be completely identified to our research model, and the "occurrences" to the document catalogues used by the application. Besides, as it is possible to type topics and associations (associations being also topics)[4] within a Topic Maps model, this research model allows naming of the descriptive elements and link them to a type hierarchy.

As similarities are quite many, our model proposes nevertheless a deeper organization in the structure.

First of all, the model marks the objects, giving a quicker access to class instances. For example, if the request points to a commercial document which is considered in our model as a specific document, the application works directly on the instances of the SPECIFIC_DOCUMENT class. Therefore, it allows a straight access to sub-objects like the product. It also interacts with disjoined hierarchies of concepts: the model lets us define class slots pointing to graphs (associated taxonomies in Figure 1) which nodes are specific values of the given slot. In the upper example, we use different hierarchies of defined values (author, type document…). This gives the opportunity for each slot to be referenced in a specific domain of organized values. From a semantic point of view, this lets the application describe each class slot in the most real manner.

At last, in Topic Maps, whereas the links between occurrences require going through the abstract level in order to navigate, our research model allows a direct access within transversal pointers.

### 3.2.2. UML implementation

Even if our model has functionality similarities with Topic Maps, it mainly differs by our model which follows a UML implementation.

In the application we present here, data are encapsulated in a class hierarchy and are manipulated not as typed literals (cf. Topic Maps URI) but as objects. The user can therefore express his request upon more complex components, like for instance find all partners having dealt with a given contract.

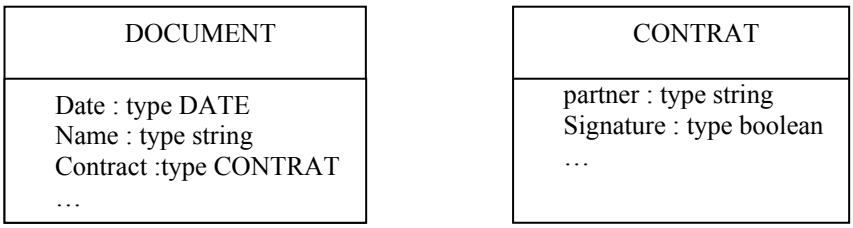
---

- a *graph of topics* (computing representation of a subject) linked by
- *associations* (relation between 2 or more concepts, where each concept is an actor of a predefined *role* defined by the association), each of this topic being instantiated by
- *on or more occurrences* (a URI pointing to a resource (a Web page, a document part, a picture..)).

[4] Marking an association in Topic Maps corresponds to naming fields in our model. Besides, occurrences in Topic Maps could be of different ranges. It could be pictures, numbers, text… The Topic Maps standard handles these distinctions through occurrence role concepts and types (being themselves topics). This is equivalent in our application to the range specification of a slot class .

This has been made possible with a document modelisation containing a DOCUMENT class which contains itself a complex type slot CONTRACT, which contains itself a slot pointing to the name of the partner (**Figure 6**).

**Exemple :**




**- Figure 6 -**

## Multidimensionnal annotation

The object model we refer to is issued from a class hierarchy, each class of it having slots of different types :

Basic: *Boolean*, *string*, *num* and *date*

Extensible: *symbol*, from a set of predefined taxonomies, *object* (in a class hierarchy).

Our goal is to improve the precision and the fastness in the information retrieval by offering the user the opportunity to handle with coherence her own research process depending on different criteria. For instance, the user can choose to retrieve a document from a **publishing date** point of view, and, if she wants it, change filter and go further in the exploration by choosing the **author name** criteria.

## Slots with values in a hierarchy of concepts

A large effort is made today to organize data. It appears that a classification of superior order should allow an easier retrieval of information. For instance, Yahoo[5] performs classification by organizing links in sections. This engine, labelled as directory, shows undoubtedly a semantic dimension since it can classify responses in topics from the most general to the most specific ones. Another example is the search

[5] http://www.yahoo.com/

engine Lycos[6] which accomplishes research upon predefined categories like journeys or computers[7].

Nevertheless, even if these approaches are close to our, they can't guarantee to return the most pertinent pages. Moreover, it is sometimes impossible to express a request giving a satisfying solution. For instance, taking Yahoo in order to find all information about Paris (as keyword input) in an educational context (for this, we click in the education area), we get answers like Paris airport, Paris-Match (a people magazine). The Sorbonne University, which is as matter or fact, an interesting result, only appears among a huge amount of inconsistent responses.

## Semantic annotations

It is of course possible for a single object to be defined by numerous descriptive systems. It can fit in different classifications, a classification being considered in our case as a point of view referencing a domain of concepts. We propose here to define hierarchies of concepts and transform the document data into instances having slots with space values falling into these hierarchies.

In a related work, conceptual clustering approach has been studied [Chu *et al*, 1996], based on frequency and value distributions of data. But if this allows discovering high level concepts of numerical attribute values, it does not apply to semantic ones, and also is performed in a multidimensional approach.

The taxonomies files used by the application are references to hierarchies of symbols (concepts). They are trees in which each node is an identifier linked to the others by a specialisation relation. A taxonomy represents for a given slot the set of all values taken by the database objects for this slot, still with a deeper organization given by the graph. We distinguish several levels of details, which purpose is to allow a better instances organization and also to permit parent concepts inheritance. As matter of fact, performing a query about a construction site in Dunkerque (a French northern city) implies that it could be done just by refining documents issued from a request upon northern construction sites.

This hierarchical structure gives mainly during the request process an additional dimension to the one simply given by atomic values slots (see example §2.3.4).

## Request example

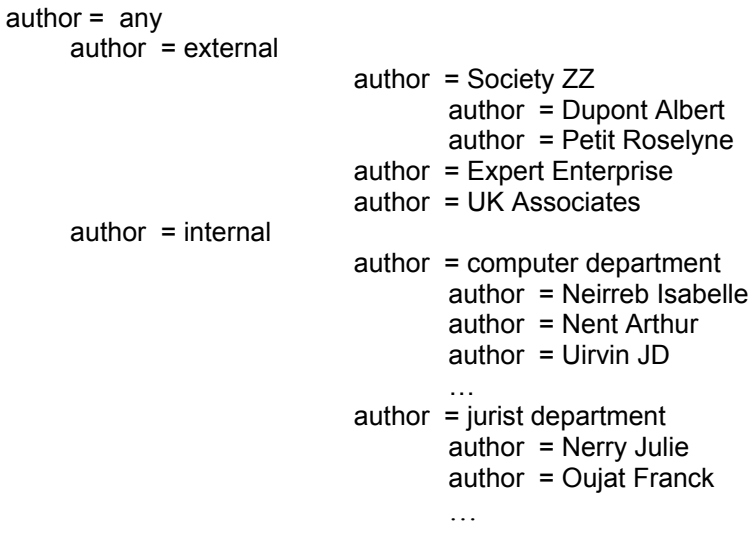
Let's imagine a set of Topic Maps about distinct domains on which we would add a supplementary organization, letting us manipulate the values in a different manner than associations. A concrete request example is the retrieval of recent documents written by external partners of the **Expert Enterprise** society. We present below a

---

[6] http://www.lycos.com/

[7] search engines like « dogpile » (http://www.dogpile.com/) are beginning to appear (http://mc42.free.fr/moteurs.htm) which allow the user to express a query about specific document types (picture, MP3 files..)

part of the reference taxonomy file (*e.g.* the concepts hierarchy) for the slot "author" of the class DOCUMENT (**Figure 7**).




**- Figure 7 -**

All instances of the DOCUMENT class have an author slot with values pointing to this graph of concepts. Values like **Neirreb Isabel**, **Nent Arthur** or **Uirvin JD** could all be described by the concept **computer department**. This given hierarchy lets the user know that these authors are a part of the external authors, also that there are 3 different partners' enterprises (Society ZZ, Expert Enterprise, UK Associates). This concepts organization let us also deduce that the topic **computer department** is closer to one of the three mentioned authors than **Roselyne Petit** for instance. We must assimilate the term "closer to" to "best approximation".

All the solutions held inside the inheritance given by the tree, which are in our example the different departments in which the authors work.

## Criteria combination

The search engine Excite[8] for instance, like other numerous engines, order the solution sets depending on the combination of various criteria like the keywords frequency, links pointing to the pages,…These engines first of all sort the results according to a keywords filter (it is the only given field), the other criteria coming after. In order to get an adequate answer, we get two choices: either we remain patient and take a look of all the answers, or we give more than a single keyword in order the

[8] http://www.excite.com/

precise the context. But this method drawback is that we can face a "no solution situation" [9].

The model gives the possibility to handle rich data, and the opportunity to formulate request with many criteria is therefore accessible.

In response to the first difficulty invoked in the beginning of the article about the benefit of working with well described data, our model allows to handle a rich data organization. It lets the user express her request in a manner close from her wish and therefore get pertinent answers.

### 3.3. Production and consultation of data

From a new document publishing to the criteria required to capture it during the query process, we distinguish several steps:

#### 3.3.1. Assisted publishing of documents (Saisame)

Wishbone has integrated dynamic forms for publishing or querying based on techniques analogous to those proposed by [Hermens et al, 1993].

The user, when publishing a new document, is proposed values for each field gradually while filling the form, that he can validate or not. These values are the result of prediction in extension of the most probable values for each slot. It is not only for facilitating the work but also to prevent from wrong publishing. For instance, if an author is in charge from its beginning of a contract coming made from the UK Associates Enterprise about the product **UKPro**, while filling the field corresponding to the name product, the service may propose the value "**UKPro**" (from statistically results issued from the database). Besides, the database content evolutes in time and this assisted process can be view as a preventive procedure against absurdity [Sullivan, 2001]. For instance, we can mention the site Yahoo which hires at least 200 librarians only to ensure pertinence and quality of the exposed concepts [Russom, 2001].

#### 3.3.2. Capture of request criteria

We describe below the help given to the user while searching for a document.

*Model-assisted values capture*

For fields pointing to a hierarchy of concepts, the application form only proposes values of the corresponding taxonomy. For instance, the taxonomy of authors only

[9] Search engines like Google (http://www.google.fr) improve enhance the pertinence of hits by gathering the 2 approaches (directory + engine). For instance, Google shifts its request to Yahoo if it doesn't find a solution.

concerns the documents authors and the hierarchy of the document types only the field of document nature. For this, we settled a workshop for interface production. This workshop is used for generating periodically a synchronized interface in regard to the model. This allows integrity and also source data conformity of the form fields to be fully preserved. If a new document is published and its author (for instance **Mrs Roselyn Petit**) does not yet appear in the corresponding form field (her first published document then), an automatic refreshing process of the form taxonomies will induce the display of **Roselyn Petit** in the list of selectable authors.

*Light capture*

The form presents an interesting feature in that all fields are optional. In return, the query process seems smoother. The user can fill whatever she wants (from 0 to all fields) while knowing that whatever her request is, she will get a response allowing her to continue in the navigation process without having to go backwards.

*Dynamic capture*

Also, the form can display if wanted additional fields associated to the slots of a sub-object (of the query object). For instance, when searching for commercial documents (which instances are of class SPECIFIC_DOCUMENT), the form displays a part dedicated to the contract (which is a sub-object of the SPECIFIC_DOCUMENT class). This dynamic display renders the interface lighter (not on the screen if not needed), plus, it gives the user an opportunity to understand the base logic more quickly.

### 3.3.3. Integrity rules: model role

The model is very important in the sense that it fills several functions: the first and principal one is that it acts as a validation component for data. For instance, navigation using clusters is quite sensitive to isolated elements. If the database contains a document which was published in year 1299, the application will point out to this document by creating a set of clusters starting from the 13$^{th}$ century. Thus it is important to have a precise model of the admissible values for each slot. We have implemented a mechanism of constraint validation, similar to the XML Schemes [WWW Consortium, 2001] according to the following criteria:

- definition of a values space for each class slot
- implementation a default values
- required slots, types slots
- key-referenced objects (“idref” in XML descriptors like XLINK [W3C, 2001].

The Wishbone engine loads the database and validates each object. The ones which don’t verify constraints defined by the model are ignored and thrown. The use of constraints for allowed data values is a regular data modeling approach allowing complete data cleaning. Therefore, they are inaccessible by the research engine. This is called the **validation data phase**.

All these mechanisms (assistance, validation) have been implemented in order to make easier the maintenance and the publishing of coherent data. It gives a response to the document database management.

### 3.3.4. Scalability and performances

The Wishbone engine has been tested over a 50000 objects database within an average time response of less than a second per request.

Tests have been performed on a Pentium 4 Xeon (2.4 GHz), but charge could be distributed through a client-transparent in-between component enabling load balancing and queuing.

# 3. Conclusionn and future work

Our approach of navigation into content improve the user experience not only from a qualitative point of view (quality tests in work), but also from a quantitative level (good appreciation from the persons in charge of the protocol in study). The generated interaction by the Wishbone application, being mostly inspired from human dialogs, settles an atmosphere of confidence with the user.

"You want all documents treated by UK Associates since 1998? No problem! We've got 7, the most recent ones, concerning the product UK200. Otherwise, we have 10 documents published between 1999 and 2001 and they're all commercial contracts; and we've got 15 documents published before 1999 and all the authors are internal to the enterprise."

The user disposes of a friendly interface (dynamic capture of criteria values, coherence field values, optional fields), which enhances the data accessibility through a high pertinence of the response display (views, relaxation).

This could be done with a light and safe maintenance circuit we call information cycle, with a major attention to apply an organized -intelligent like- frame to the data according to an appropriate business model.

Still, part of our future work will focuse on

- The portability of the engine in regard of internet stored data. This means of course previous meta-data extraction but this complex process is not the purpose of the engine, rather takes place as second phase in the loop. Many related works like annotating data have already focused on that subject [Shet *et al*, 2002][Staab et al, 2001].
- Extension of clustering algorithms on semantic structures in order to improve the pertinence of clustering

**References**

[Le Grand, 2001] Le Grand, B. (2001). *Extraction et visualisation de systèmes complexes sémantiquement structurés*, Thèse de doctorat Université Paris 6.

[Sullivan, 2001] Sullivan, D (2001). *Five Principles of Intelligent Content Management* Intelligent Enterprise Magazine, August 31, 2001 – pp1-5

[Chu *et al*, 1996] Chu, W. W. ;Chiang, K. ; Hsu, C.-H. ; Yau, H. (1996). *An Error-based Conceptual Clustering Method for Providing Approximate Query Answers*, Communications of the ACM, 39,12es, pp 216-230.

[Staab et al, 2001 ] S. Staab, A. Maedche, and S. Handschuh (2001). *An annotation framework for the semantic web*, In Proceedings of the First Workshop on Multimedia Annotation, Tokyo, Japan, January 30-31, 2001,.

[Shet et al, 2002] Shet, A. *Relationships at the Heart of Semantic Web: Modeling, Discovering, Validating and Exploiting Complex Semantic*

*Relationships*, 29th Annual Conference on Current Trends in Theory and Practice of Informatics Nov. 24 -- Nov. 29, 2002

[Russom, 2001] Russom, P. (2002). *Managing Spaghetti Content* Intelligent Enterprise Magazine, May 28, 2002 – pp1-2

[Caseau and Laburthe, 2002] Caseau, Y. and Laburthe, F. Bouygues S.A. (2002) *Method for the data-driven adjustment of queries over a database and system for implementing the method*, European patent request 02290920.4

[XTM Authoring Group, 2001] TopicMaps.Org XTM Authoring Group (3 March 2001), *XTM: XTM Topic Maps (XTM) 1.0 : Topic Maps.Org Specification*.

[Gruber, 1993] Gruber, P.(1993) A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993

[Hermens et al, 1993] Hermens, L. A. and Schlimmer J. C. (1993). *A Machine-Learning Apprentice for the Completion of Repetitive Forms*, Proceedings of the 9th Conference on Artificial Intelligence for Applications ({CAIA}'93)" pp164-170, IEEE Computer Society Press.

[Berrien and Laburthe, 2003] Berrien I. and Laburthe F. (2003). *Meilleures Interfaces entre services et utilisateurs*, JFT 2003 (accepted for publication).

[WWW Consortium, 2001] World Wide Web Consortium (2001), *XML Scheme Definition Language*, W3C Recommendation, 3 May 2001.

[W3C, 2001] W3C (2001). *XML Linking Language (XLink) Version 1.0* W3C Recommendation, 27 June 2001.