HyTropes: a WWW front-end to an object knowledge management system

Jérôme Euzenat

INRIA Rhône-Alpes
655 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France
Jerome.Euzenat@inrialpes.fr

Abstract: HyTropes is a HTTP server allowing the manipulation of a knowledge base written in the Tropes object-based representation language through the world-wide web. It allows the navigation through the knowledge base as well as the invocation of search queries (filters). The display can be customised in order to best suit the needs of the applications. HyTropes will be demonstrated through three prototypic knowledge bases: ColiGene and FirstFly devoted to the genetic regulation of various organisms and STG, a bibliographic knowledge base.

HyTropes is a HTTP server allowing the manipulation of a knowledge base written in the Tropes object-based representation language through the world-wide web. It is briefly presented here through screen shots. Some paragraphs below are extracted from a companion paper (Euzenat, 1996b). The technical description of the system, the Tropes manual together with several on-line demonstrations is available at `http://hytropes.inrialpes.fr`. The description of HyTropes is made through several knowledge bases developed in Tropes:

Real-estate is a sample knowledge base concerning real-estate agency. It illustrates the potentialities of HyTropes;

ColiGene (Perrière and Gautier, 1993) describes the regulation mechanism of gene expression in the *E. coli* bacterial genome. Only the "conceptual schema" of the base is available (no object is visible);

FirstFly (Chemla, 1996) describes the genetic regulation in the early stages of the *D. melanogaster* (fruit fly) development;

STG is an experimental bibliographic knowledge base about granularity in representation of space and time (Euzenat, 1995c).

After providing the motivation behind HyTropes (§1), a guided tour of the system is given by describing Tropes and its browsing through HyTropes (§2), the use of structured queries (§3), a simple customisation by including HTML pages in HyTropes pages (§4) and a more elaborate customisation which enables to enhance HyTropes by adding particular functions and re-designing instance pages (§5).

Motivation

Mixing knowledge bases and hypermedia has been already achieved for long. This idea can be put forth for various reasons:

- The knowledge in knowledge bases cannot be considered in isolation from other knowledge sources available in firms or laboratories (e.g. bibliographic references, full text papers, experimental data and programs). This relationship must be established in one way or another. For anyone who wants to explain, to annotate or at least to document a knowledge base, "hypermedia" are now the main support;
- The World-Wide Web (WWW) enables the publication of a knowledge base. Thus even users with little computer knowledge can access it at low cost: one server can have many

users on inexpensive workstations (Riva and Romani, 1996; Farquhar, Fikes, Pratt, and Rice 1995);
- Another deeper reason is that the possibility to consult a knowledge base like an encyclopaedia is considered a major interest by several researchers. Knowledge-based WWW pages can be read as hypermedia documents and also consulted for problem solving (Gaines, 1990). This idea is related to the knowledge medium concept (Stefik, 1986). It has been further investigated by (Gaines, 1990) and (Rechenmann, 1993).

The success of the WWW is an opportunity to test this latter idea in the large. The WWW is a very good support for the diffusion of knowledge. However we claim that the formal representation of knowledge is a very important issue for the WWW — see also (Weld, 1995; Hüser, Reichenberger, Rostek, and Streitz, 1995). As a matter of fact, the results actually provided by the various WWW worms are so numerous and so often irrelevant that formal organisation of knowledge will soon be unavoidable. Moreover, such an organisation should help the user and the server to share some common ground.

As already noted by various authors — e.g. ISF (Rees, Edwards, Madsen, Beasley, and McClenaghan, 1995), WebMap (Gaines and Shaw, 1995) — an object-based representation system is already a web of related objects. The similarity between WWW pages and objects is quite obvious and the mapping from one to the other is straightforward.

Knowledge bases can be used as WWW servers whose skeleton is the structure of formal knowledge (mainly in the object-based formalism) and whose flesh consists of pieces of texts, images, sounds and videos tied to the objects. Turning a knowledge base system into a WWW server is easily achieved by connecting it to a port and transforming each object reference into an URL. If the knowledge base is already documented by WWW pages, the latter remain linked to or integrated into the pages corresponding to these objects.

The object-based knowledge representation system TROPES is provided with its HTTP server counterpart HYTROPES (Euzenat, 1996). HYTROPES has access to the internet, the internet has access to HYTROPES but access can also be easily restricted to the so-called "intranet". The advantages of such an approach with regard to the previous proprietary hypertext systems are chiefly the availability of the knowledge base content to a wide and untrained audience. However, other advantages are found in the consistency of the base (for instance, there is no dangling link since the skeleton is generated automatically and formal information is maintained sound).

NAVIGATION THROUGH TROPES

A knowledge base is divided into a set of concepts. These concepts constitute a partition of the objects of the base. For instance, they can be `gene`, `protein` and `interaction` in a biological knowledge base and `house` and `room` in a real estate knowledge base.

The concept first determines a set of fields (a.k.a. attributes or slots) which are possible for the objects of the concept. The name space of these slots is the concept (i.e. the same name refer to the same field throughout the concept while it can refer to something else in another concept). The fields are defined through their name, type (i.e. string, integer or another concept like `room`) and constructor (a single value, a set of values or a list of values). A subset of these fields constitutes the key of the concept which is used for identifying the objects. For instance, in a real-estate knowledge base, a `house` can have a `surface` field whose value is an integer and a `rooms` field whose value is a set of `room` objects (instances of the `room` concept).

The concept also determine a set of objects which belong to the concept. Each object corresponds to an individual in the modelled domain. It is described by the values of its fields. Only the key fields are compulsory since they enable the identification of the object. For instance, the `myHome` object is identified by its address, (i.e. `town name`, `street name`, `number`, `floor` and `id` fields whose values are respectively “Montbonnot”, “rue de l’Europe”, 655, 2 and 215).

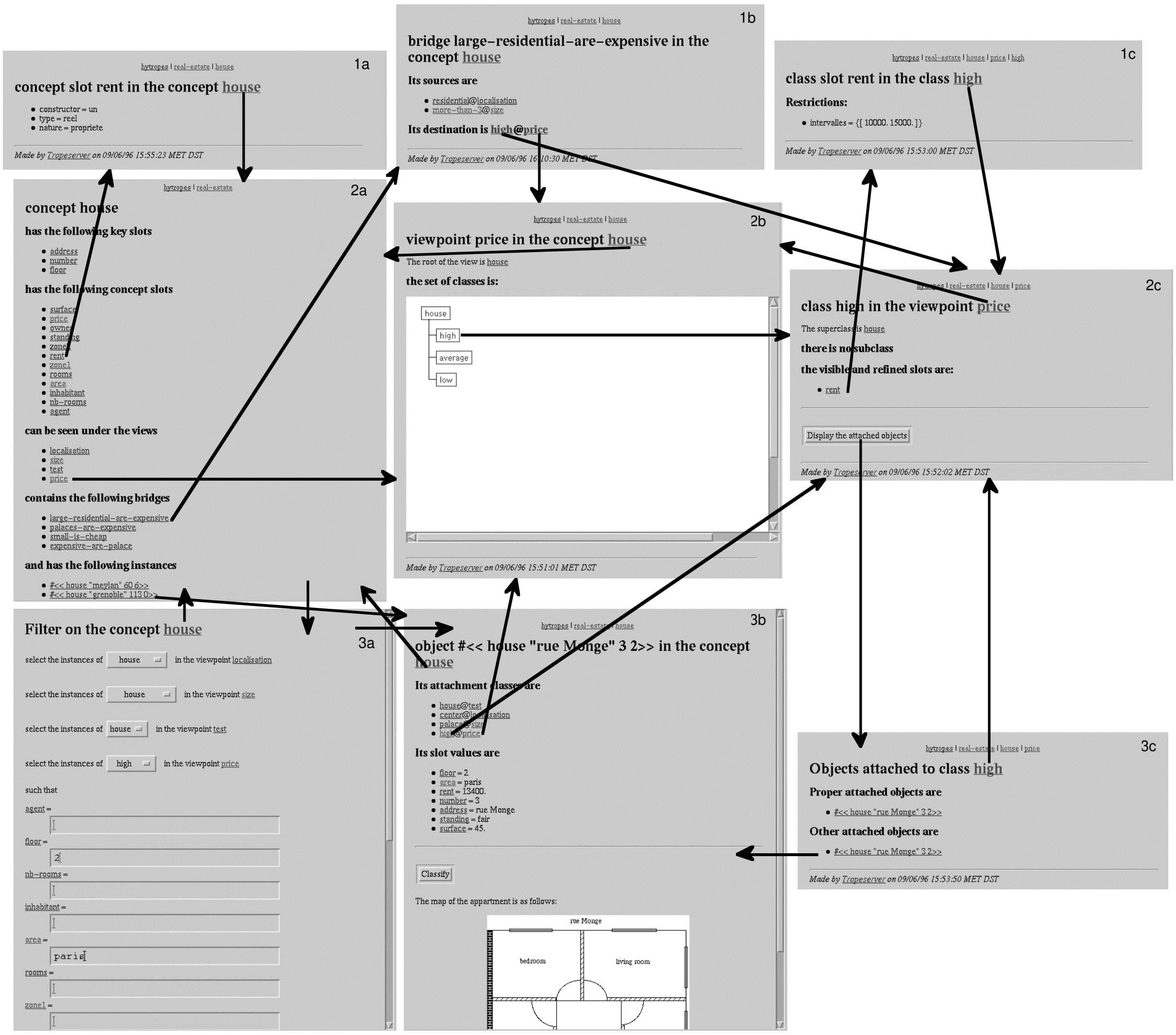

Figure 1. The connection between the pages concerning TROPES entities (Real-estate). They correspond to the following entities [rows are numbered and columns are lettered] (2a) concept, (2b) viewpoint, (2c) class, (1a) concept slot, (1b) bridge, (1c) class slot, (3a) filter, (3b) object and (3c) list of attached objects. The links between pages correspond to direct links between them.

The concept determines a set of viewpoints identified by their names. Each viewpoint is made of a class hierarchy. This is a very special feature of TROPES. Usually object systems allow only one hierarchy in the system. Seeing a set of objects (a concept) from several viewpoints which organise the objects differently eases the understanding of the knowledge base. In the real estate knowledge base, the `houses` can be seen under the `price`, `localisation` and `size` viewpoints which allow to directly access the corresponding classifications. Under the `price` viewpoint, the hierarchy is divided into high, moderate and low prices. A class is characterised by its name, its super-class, its sub-classes, its fields (a.k.a. class slots) and its set of attached objects (the objects which have been classified under that class in that viewpoint). The fields constitute a sub-set of those defined in the concepts and can support additional constraints restricting the range of possible values to a particular set of values, intervals or classes, excluding some values or restricting the cardinal of sets and size of lists. For instance, the value of the `rent` field under the `high` priced class is restricted to be in the [10000. 15000.[interval.

Finally, a concept also defines bridges which can be understood as monadic Horn-clauses. They specify that the objects attached to a set of classes (under different viewpoints) must be attached to another particular class under yet another viewpoint. In the real estate knowledge base, `large` and `residential` house are classified under `high` priced houses.

Each of these entities can be explicitly manipulated by the TROPES user (through an Application Programming Interface). It can also be displayed on a HTTP client through its own HTML page (figure 1 and 2). All the pages are generated on demand (i.e. when a URL comes through HTTP) by the HYTROPES program. They make numerous references to each others like figure 1 demonstrates. They also display various documentation and give access to TROPES functionalities as shown below (figure 2).

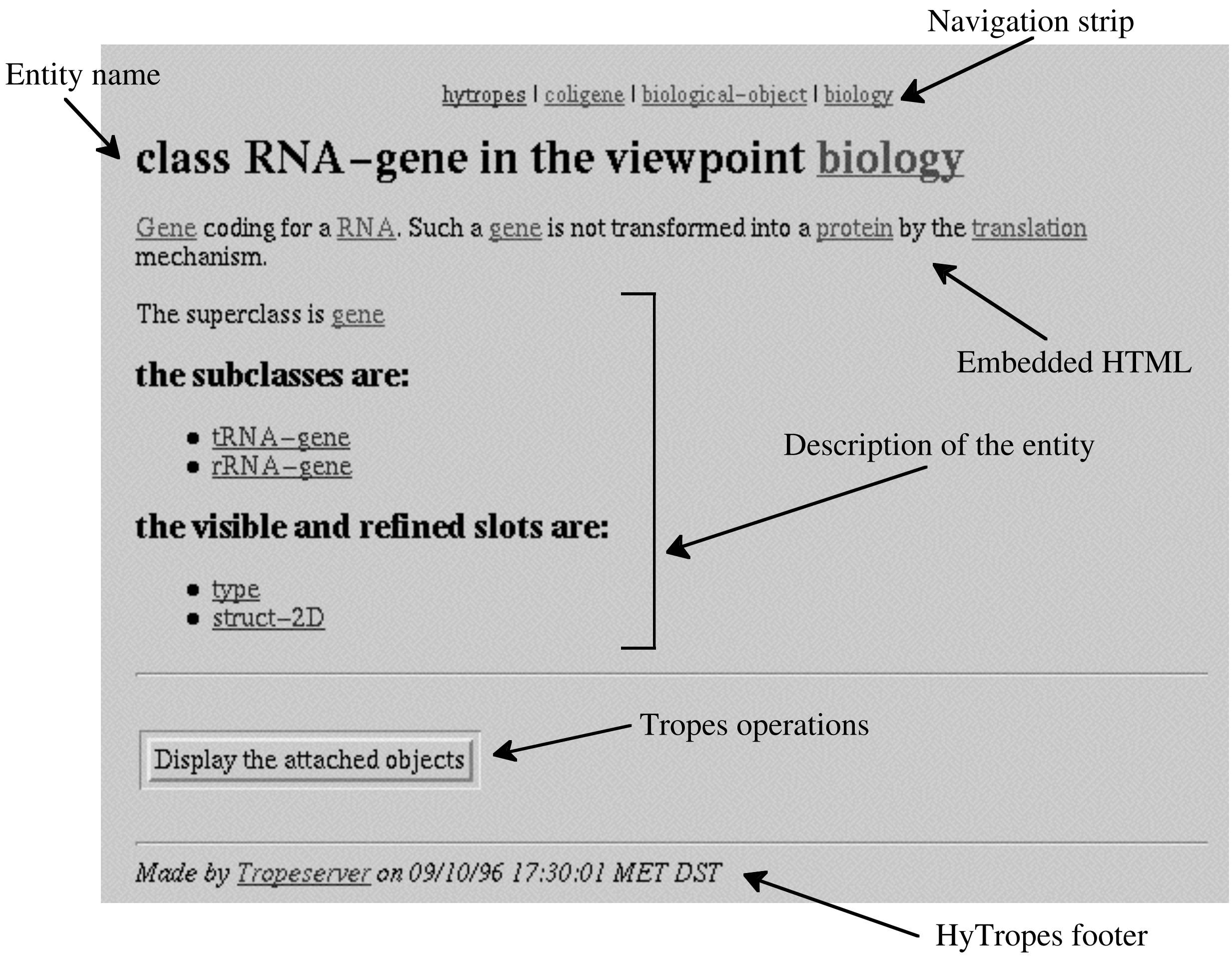

Figure 2. Detail of a TROPES generated page (ColiGene). It shows the general patterns of a HYTROPES page: a strip accessing super-entities, the identification of the current entity, an embedded description of it, its description from the TROPES content which includes links to other entities, a set of buttons triggering TROPES functionalities, another embedded HTML description and the HYTROPES footer.

QUERYING THE KNOWLEDGE BASE

World-wide availability and safety are valuable contributions, however they are quite restricted with regard to the possibilities of knowledge bases. As a matter of fact, from a knowledge base server it is possible to build complex queries grounded on formal knowledge (see figure 3). For instance, a user looking for an apartment in a real estate knowledge base can first select a filter form from the `house` concept; the user can select one of the objects given as answers and have a look at the ground map and at a picture of the house together with the usual precise information. This combines the advantages of a very structured server with the freedom of usual servers. Moreover, the answer will be given through a semantically sound method instead of using a simple full-text search.

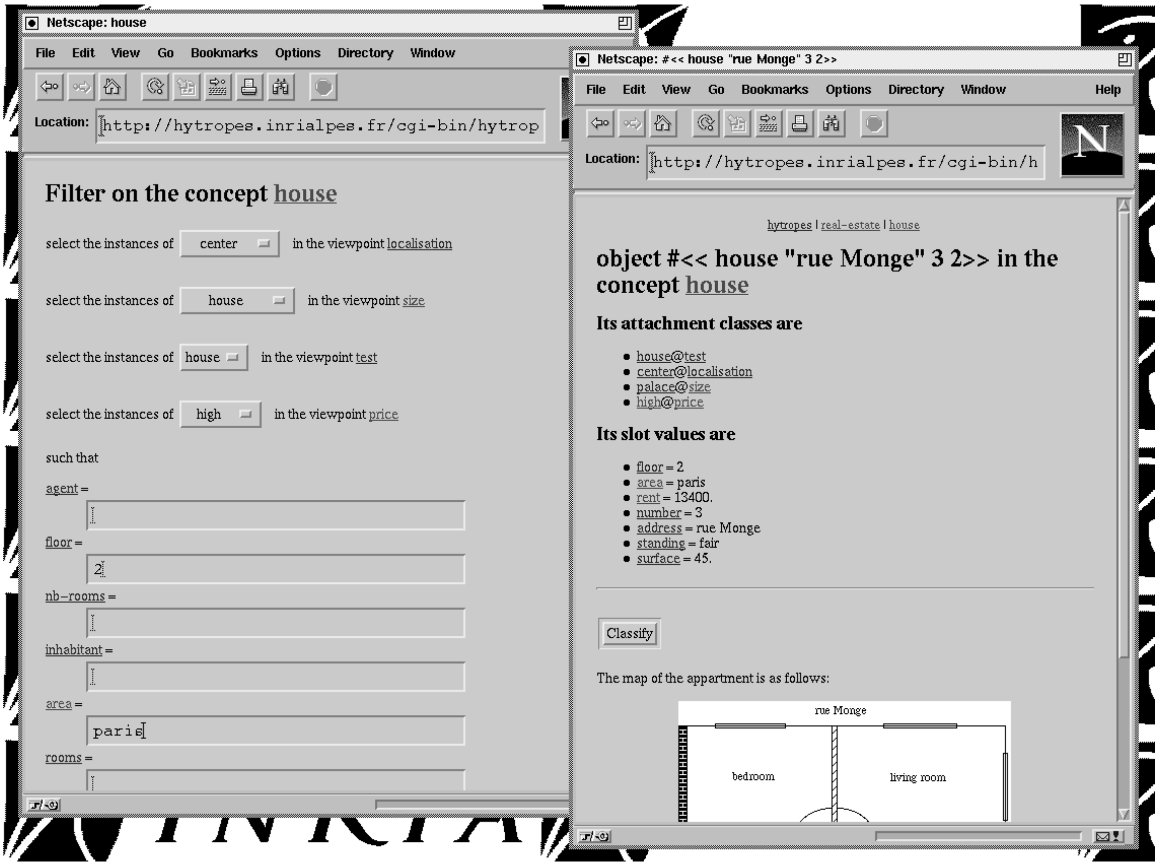

Figure 3. Filter and instance WWW page from a sample real estate knowledge base (Real-estate).

Another well-known aspect of object based systems is their ability to classify objects in various classes according to the values of their fields. TROPES is no exception and allows to process classification algorithms interactively on several viewpoints simultaneously. The result of classification is displayed on a graphic drawing of the hierarchy using colour code to distinguish possible from impossible classes.

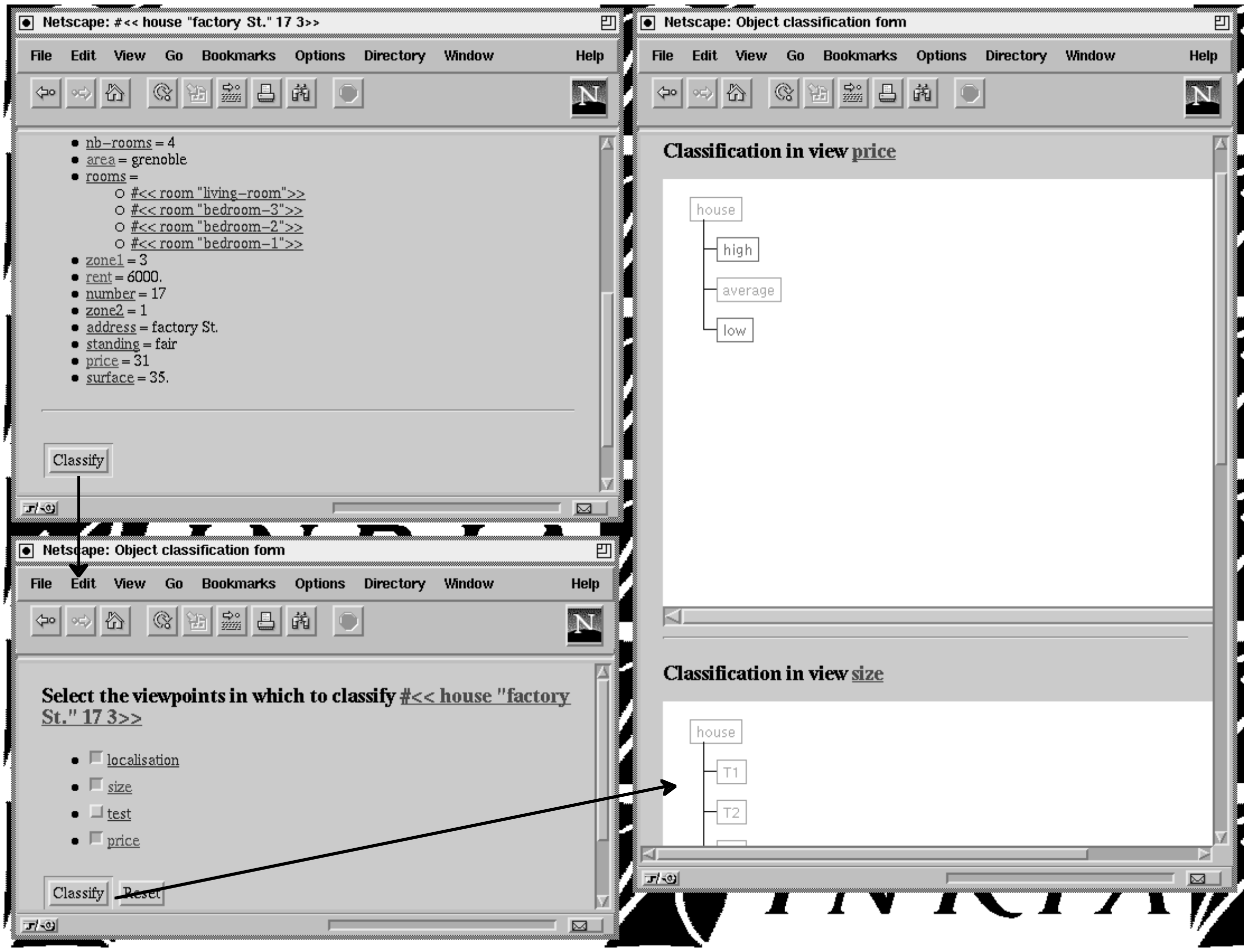

Figure 4. Classification query (Real-estate). The particular house is asked for classification under the viewpoints `localisation`, `size` and `price`. The result is displayed as a sequence of taxonomies whose classes are coloured in red (`high` and `low` under `price`) when the class is impossible and green (`house` and `average` under `price`) when it is possible for the object.

CUSTOMISATION BY ADDING HTML ANNOTATIONS

On each generated page the user can add annotations in the form of a HTML page (which may contain references, images, applets…). These annotations will be included on the fly by HYTROPES at two particular places: at the top of the generated page, just below the title, and at its end, just above the footer (see figure 2). They are usually used for documenting and explaining the various components of the knowledge base. Figure 5 shows the documentation of the ColiGene knowledge base. It is noteworthy that this inclusion can lead the users of HYTROPES to browse to other pages than those generated by HYTROPES, they may never come back to HYTROPES.

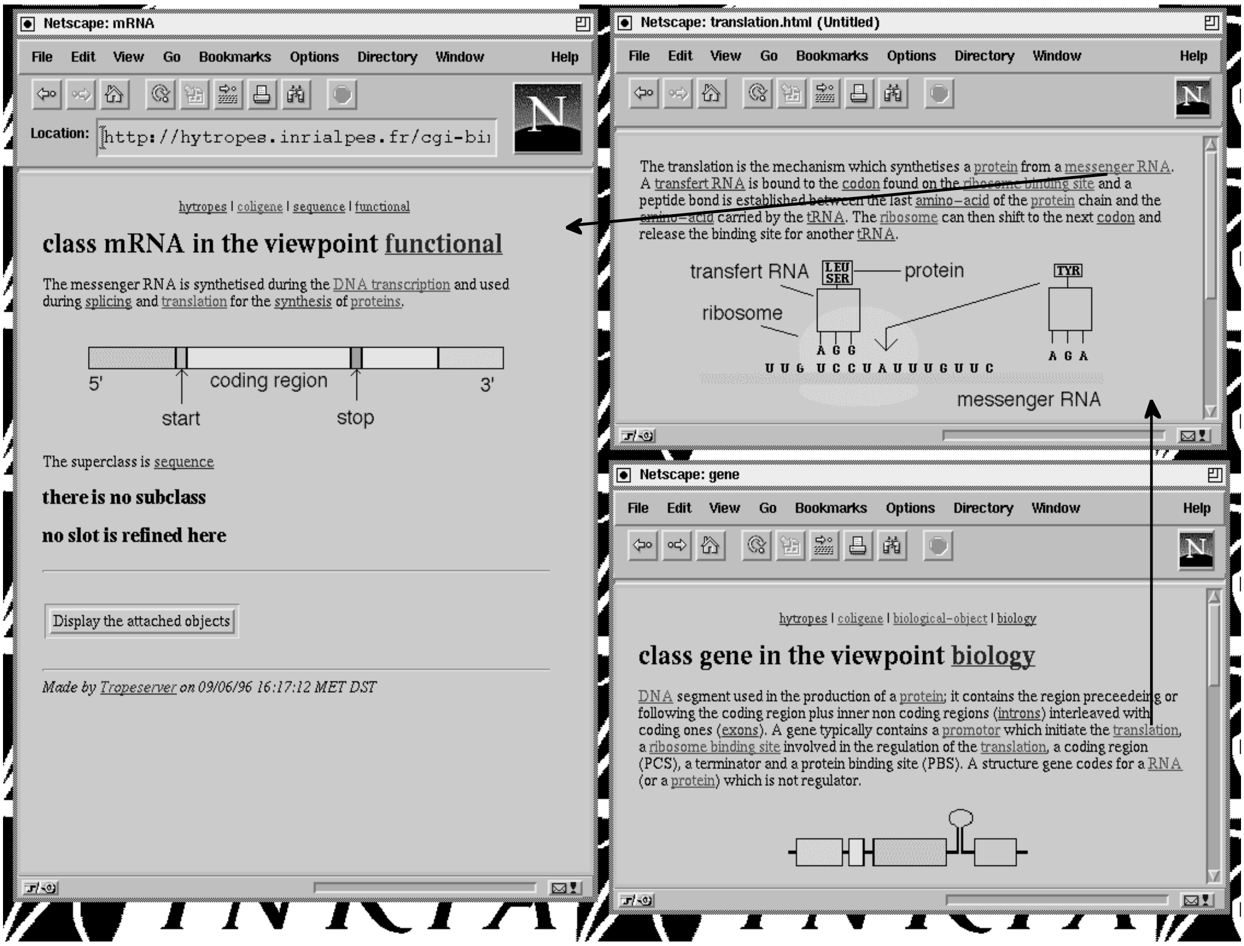

Figure 5. Navigating through hypertext (ColiGene). From the `gene` class illustrated by a text and a picture provide access to another WWW page describing the translation process (not represented by a TROPES entity) which, in turn, access to the `messenger RNA` class page illustrated by a text and another picture.

By simply adding such an inclusion in the footer of the first page of the FirstFly knowledge base (figure 6), the system is able to trigger more complex functionalities that can be added to the genuine HYTROPES through tasks (here they are only LISP functions). The HTML code corresponding to the button on the first panel (in figure 6) is the following:

```
<FORM
  METHOD=GET
  ACTION="http://hytropes.inrialpes.fr/cgi-bin/hytropes/ff/select-path">
    <INPUT TYPE=SUBMIT
      VALUE="Find the paths between two products">
</FORM>
```

It can be interpreted by HYTROPES because HYTROPES allows to define a particular function to be used when it does not recognise the keyword in the URL. This function will here interpret the `select-path` keyword and display the second panel. When the user hit the “Display paths” button, the HTTP client (NetScape here) generates the following URL:

```
http://hytropes.inrialpes.fr/cgi-bin/hytropes/ff/return-path/
  ?begin=%23%3C%3C+protein+%22BICOID%22%3E%3E
  &end=%23%3C%3C+gene+%22hunchback%22+%22FBen0001180%22%3E%3E
  &depth=7
```

which is, in turn, interpreted and parsed by the same function before calling the function searching the paths in the interaction graph.

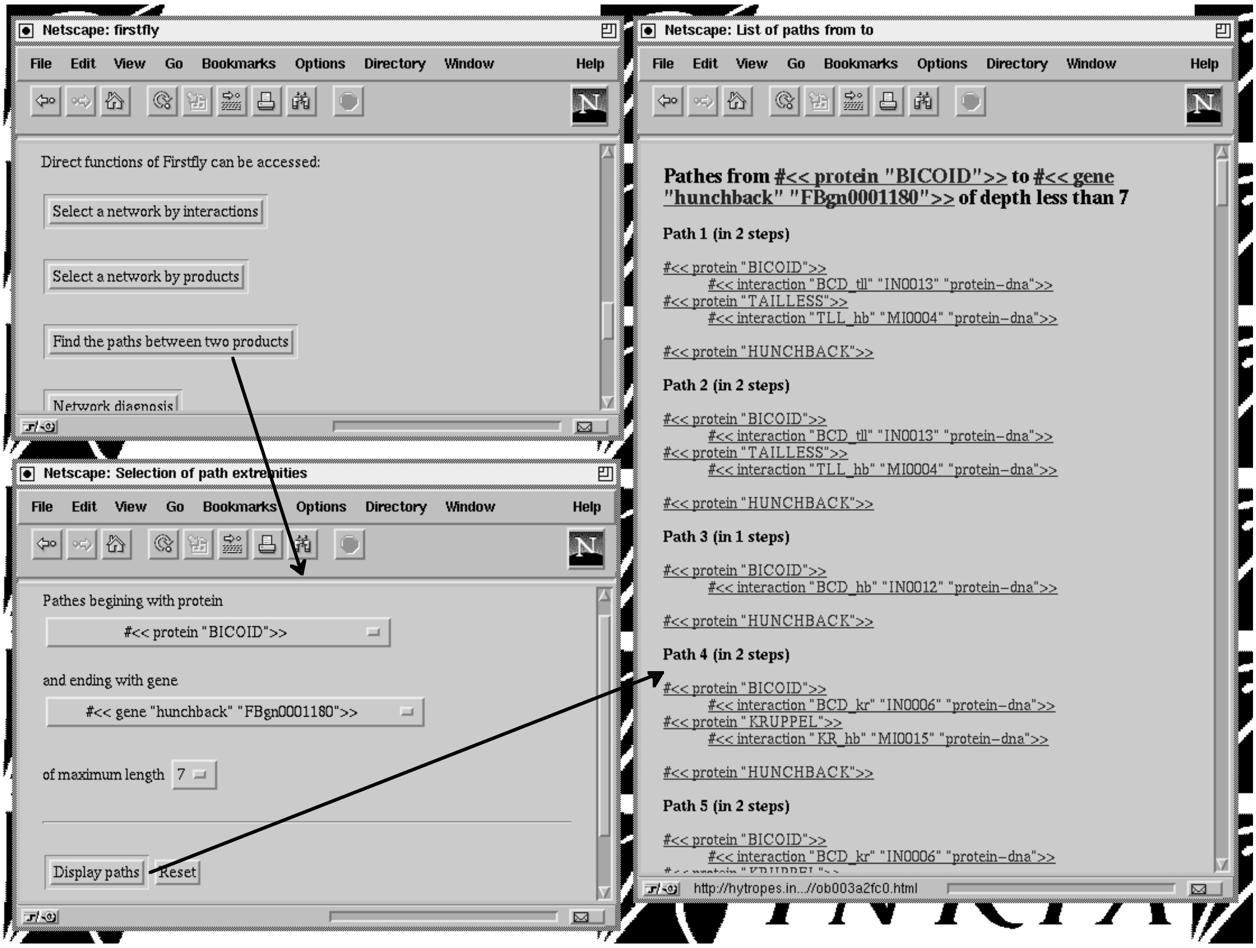

Figure 6. Integration of a task for computing paths in a graph of objects (FirstFly). The button triggers an input panel which invokes a method for computing the paths in an interaction graph.

CUSTOMISATION BY CHANGING THE DISPLAY PATTERN

A more general mode of customising the visual aspect of the knowledge base consists in redefining the function which displays the instances (this is the only one which is customisable). This has the advantage of allowing computations to take place dynamically and thus encompasses the static page inclusion seen above. As a matter of fact, redefining this function (that which does not prevent from re-using the usual shape of the instances) enables the HTML to embed specific calls (i.e. depending on the data displayed).

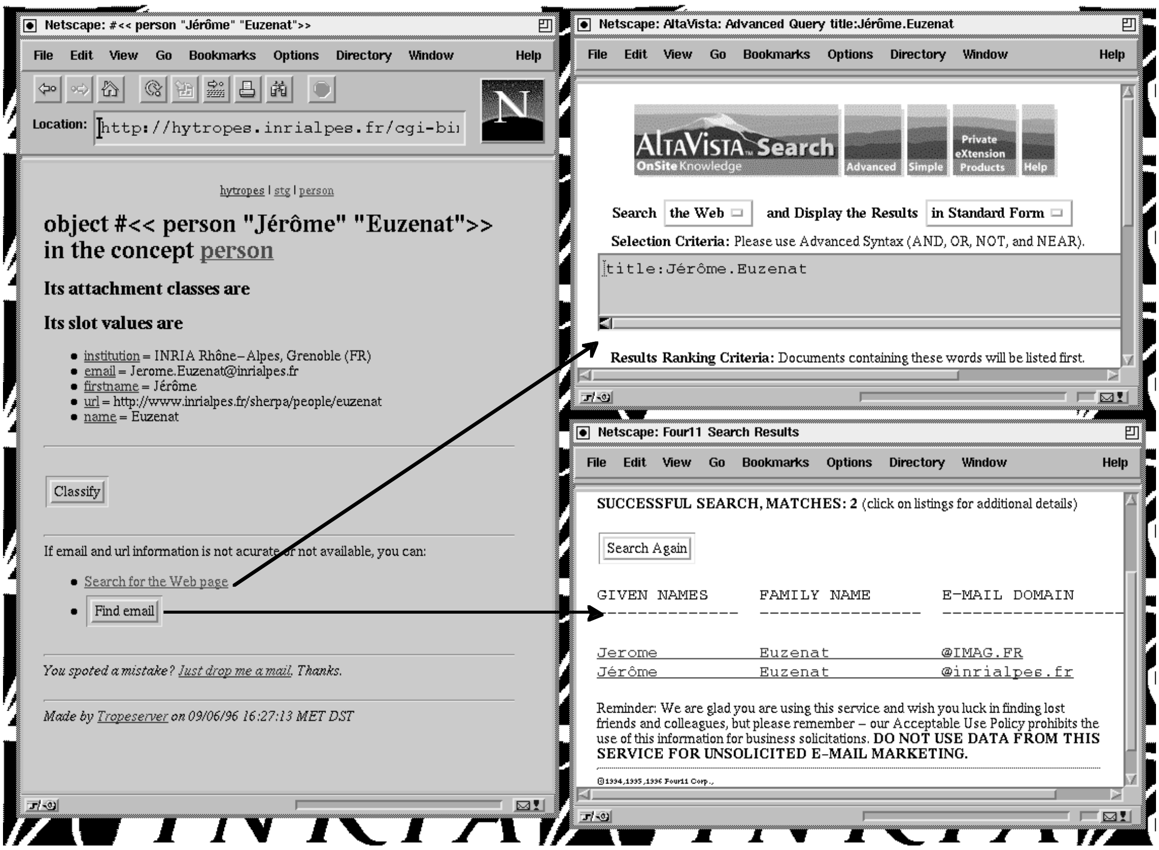

Figure 7. Integration of external CGI script calls (STG). Embedded calls trigger automatically filled queries.

For instance, in the STG base, the display of author objects contains the following HTML code:

```
<UL>
  <LI><A HREF="http://altavista.digital.com/cgi-bin/query
         ?pg=aq&what=web&fmt=.
         &q=title:Jerome.Euzenat">Search for the Web page</A>
  <LI><FORM METHOD=POST
   ACTION="http://www.four11.com/cgi-bin/SledMain?FS,234,2,377FB00,3F70FA11">
         <INPUT TYPE=hidden NAME="FirstName" VALUE="Jerome">
         <INPUT TYPE=hidden NAME="LastName" VALUE="Euzenat">...
         <INPUT TYPE=submit NAME="Search" VALUE="FindMail">
       </FORM>
</UL>
```

which corresponds to the invocation of Altavista or Four11 sites in order to obtain the home page or electronic mail address of the concerned author (figure 7).

In the FirstFly base (figure 8), the network objects are displayed through a Java applet whose invocation is as follows:

```
<APPLET CODE="IntGraphApplet.class"
         CODEBASE="http://hytropes.inrialpes.fr/java/graphdraw/"
         WIDTH=500 HEIGHT=400>
   <PARAM NAME=edges
     VALUE="Kr-Hb,Hb*Kr,tll*Hb,gt-Hb,tll-Kr,Kr-gt,gt-kni,tll-kni,Hb-kni,Hb-
gt,gt-Kr,Kr+kni,kni-Kr,kni*gt">
</APPLET>
```

It is generated on the fly and the client has all the burden of fetching and running the corresponding Java classes. If this operation (displaying graphs or even hierarchies) are repeated several times, the gains in terms of response time become important since the Java classes are cached.

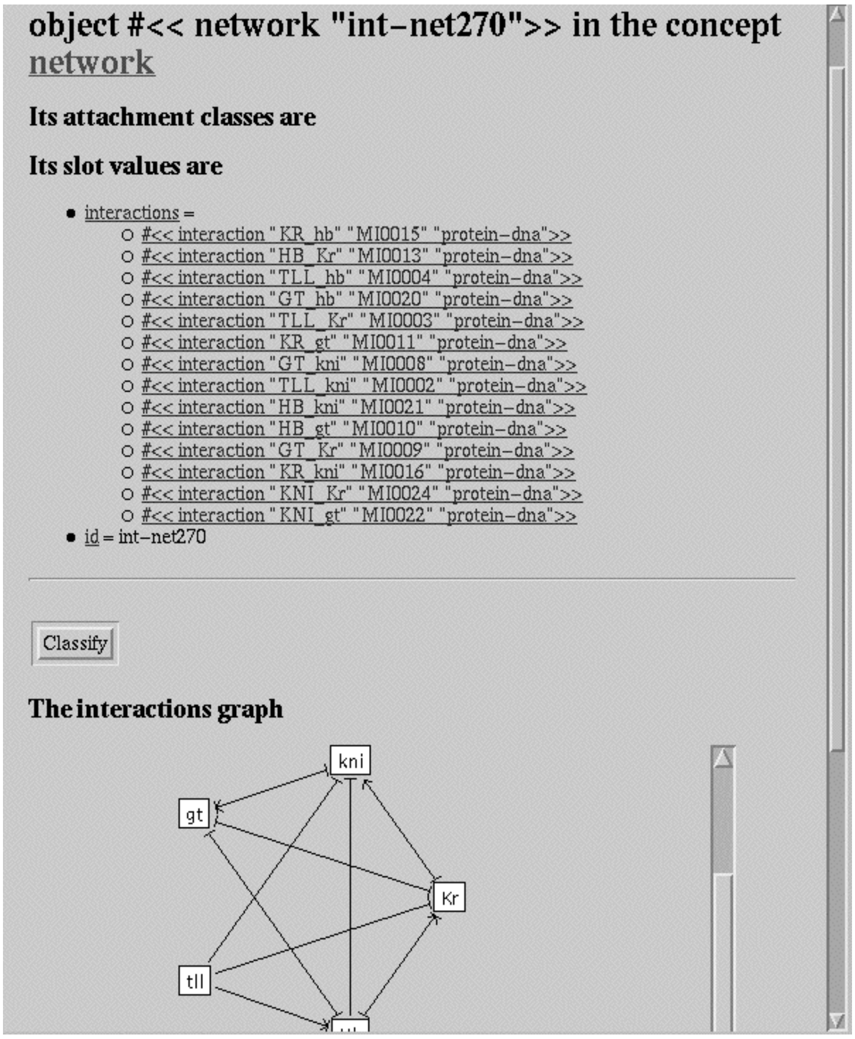

Figure 8. Integration of a Java Applet for displaying gene interaction graphs (FirstFly).

TOWARDS EDITION OF KNOWLEDGE BASES

HYTROPES makes filtering and classification queries available. It also provides access to particular actions implemented as Lisp functions or Java applets. However, it does not allow to directly edit the knowledge base (by creating objects, changing slots values, etc.). Thus, a natural HYTROPES extension is the development of an editing interface through HYTROPES. It is currently under development.

This is an opportunity to rationalise and publish all the URL rules (far simpler than the URL generated by the FORM tag, see above) for editing the knowledge base. This will enable any application disconnected from TROPES to:

- properly query a knowledge base;
- generate WWW pages linked with any knowledge base.

Modifying knowledge bases concurrently raises problems of concurrent access and user support. This is to be dealt with through two particular research issues on which we are working:

- Revision of knowledge bases (Crampé and Euzenat, 1996) which allows to provide possible repairs to detected inconsistencies. This is very important for users who introduce inconsistencies involving pieces of knowledge that they do not understand (because they have been entered by others).
- Cooperative construction of consensual knowledge bases which is detailed in the companion paper (Euzenat, 1996b). It consists in a protocol allowing a group of people to decide if a piece of knowledge must be introduced in the knowledge base or not.

CONCLUSION

HYTROPES allows to browse and consult a knowledge base through the World Wide Web. It is really appreciated by users who are not literate in computer programming and manipulation. The advantages of such a device are:

- A standard browser is sufficient for accessing the knowledge base;
- Updates are fast and transparent to users since they are made on the server side;
- The generated WWW site is consistent since it is generated from the knowledge base;
- The WWW sites are smart since they allow the use of complex queries instead of full-text search;

However, there remain limitations:

- Using HYTROPES requires network connectivity;
- The interaction mode induced by the WWW is very simple and sometime misleading;
- Concurrent modification must be added to and managed by the system.

ACKNOWLEDGEMENT

Christophe Chemla (LGPD, Marseille) for FirstFly, Thomas Koch (GMD, Sankt-Augustin) for the treetool Java applet used for displaying the hierarchies (`http://orgwis.gmd.de/~koch/Java/Applets/TreeTool`), Ulfar Erlingsson and Mukkai Krishnamoorthy (Rensselaer, Troy) for the graphdraw Java applet used in the FirstFly base (`http://www.cs.rpi.edu/projects/pb/graphdraw`) and Sylvain Chicois for the adaptation of this last applet to the needs of FirstFly. The FirstFly knowledge base has been partially financially supported by the ACC-SV13 of the French ministry of education and research (MENESR).

ADDITIONAL REFERENCES

The most part of the references can be found in the companion paper of the present one (Euzenat, 1996b).

Chemla, C. (1996). Une approche bio-informatique des réseaux d'interactions géniques impliqués dans le développement embryonnaire de *Drosophila melanogaster* (in french), DEA de biologie, Université de la méditéranée, Marseille (FR)

Erlingsson, U., Krishnamoorthy, M. (1996). Interactive Graph Drawing on the World Wide Web, Research report 13, Rensselaer polytechnic institute, Troy (NY US) [http://www.cs.rpi.edu/projects/pb/graphdraw/paper.ps.gz]

Euzenat, J. (1995c). An algebraic approach for granularity in qualitative time representation, Proc. 14th IJCAI, Montreal (CA), pp894-900

Euzenat, J. (1996b). Corporate memory through cooperative creation of knowledge bases and hyper-documents, Proc. 10th KAW workshop, Banff (CA)