6. Putting things together: the PhiloSurfical tool

6.1 Introduction

In this chapter we describe the rationale and functionalities of PhiloSurfical, the application we created for showcasing our *ontological learning pathways* approach.

The chapter is divided into seven sections; after this introduction, the sections are thus organized: section 2 gives an overview of the purpose of the tool, providing also some background information regarding the specific philosophical text PhiloSurfical has been applied to. Section 3 describes the work we have done in order to build an adequate philosophical knowledge-base. Section 4 describes the system architecture. Section 5 deals with the graphical interface and, in general, the system's user interaction. Section 6 focuses on the discussion of the *learning-pathways* we have created for browsing the knowledge-base. Finally, section 7 summarizes the results obtained.

6.2 System overview

PhiloSurfical (see fig. 6-1) is a pedagogical application, which allows the contextual navigation of a semantically-enhanced version of Wittgenstein's

Tractatus Logico-Philosophicus (Wittgenstein, 1921). The application is available online at the address http://PhiloSurfical.open.ac.uk. It can be expected to run on most web browsers, although it has extensively tested only on Mozilla Firefox (Mozilla, 2008).

By means of PhiloSurfical we can test the functionalities of the ontology presented in chapter 5. In fact, by relying on the multiple levels of abstraction provided by the ontology, the software lets users benefit from multiple perspectives on the text and on related resources.

It is important to mention that, for bootstrapping purposes (as the availability of free and adequate 'philosophical' semantic data on the web is still limited), PhiloSurfical strongly relies on an internal knowledge base created by us; nonetheless, its architecture aims to be open and extensible so to allow future integration and querying of different repositories, using the appropriate web standards (e.g., RDF (W3C, 2004b) or SPARQL (W3C, 2007)).

At the time of writing the prototype is mainly focused on *browsing* functionalities. In future versions we plan to extend it so that users could also *store* their own annotations about the Tractatus and possibly contribute to the creation of a network of philosophical resources centered around the text and its author.

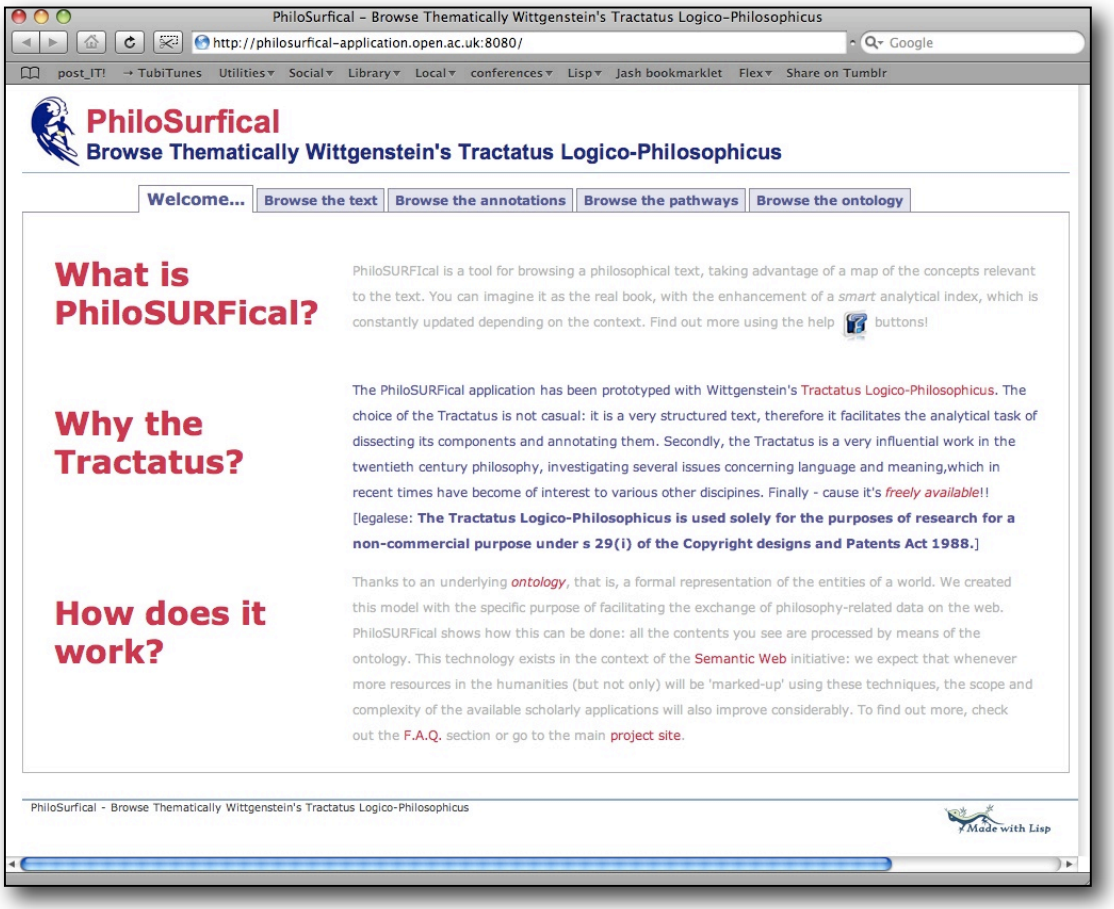


Figure 6-1. Screenshot of the PhiloSurfical application

6.2.1 The choice of the Tractatus

As already mentioned, the tool we have created is based on the Tractatus-Logico-Philosophicus by Ludwig Wittgenstein (Wittgenstein, 1921), which can be found as a freely available text from the project Gutemberg (GutenbergFoundation, 2008).

The Tractatus is widely recognized as a very influential book in the history of contemporary philosophy; mainly, it deals with topics which could be related to the philosophy of language and logic disciplines - Wittgenstein himself presented his work as an attempt to challenge the logical theories of Frege and Russell (Stenius, 1960). It is worth remembering, however, that the

philosophical positions Wittgenstein presents in his work have implications for a much wider spectrum of research fields, e.g., *philosophy of science*, *philosophy of mathematics, ethics, philosophy of religion*. Unsurprisingly, the academic literature on the Tractatus is very vast (Black, 1964) and many are the possible threads of interpretation departing from it.

The book is composed of 526 numbered paragraphs which are structured as *notes about notes* on wide-ranging topics. As Wittgenstein explains in the footnote on the first paragraph,

> The decimal figures as numbers of the separate propositions indicate the logical importance of the propositions, the emphasis laid upon them in my exposition. The propositions n.1, n.2, n.3, etc., are comments on proposition No. n; the propositions n.m1, n.m2, etc., are comments on the proposition No. n.m; and so on. (p. 14)

The book can therefore be read at least in two different ways: *linearly*, one page after the other, or *following the numbering*, that is, reading all the one-digit sections, then the two-digits sections, and so on. Following standard search terminology, we call the first order "depth first" and the other order "breadth-first". The figure below (6-2), taken from one of the first web-editions of the Tractatus (Laventhol, 1996), represents this feature of the book:

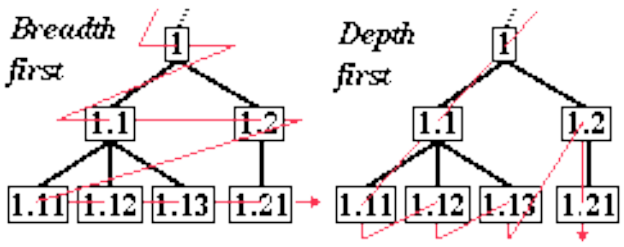


Fig. 6-2 - The two possible ways to read the Tractatus

In general, we have chosen this text for three main reasons.

Firstly, because it is freely available in a digital format. Secondly, because it is much more structured than many other philosophical books: as a consequence, the definition of the basic *unit of meaning* (in our case the *paragraph*) is indeed easier. Thirdly, for the fact that it deals with a variety of themes and research areas, thus guaranteeing the existence of multiple overlapping interpretative contexts on the same 'elementary items' - this feature provided us with an adequate resource for showcasing the advantages of semantic navigation mechanisms.

In conclusion, by using the Tractatus we fundamentally aimed at benefiting from a philosophical text which is *full of ideas, easy to annotate and easy to manipulate*.

It is nonetheless important to stress that we *could* have carried out the same work also with another philosophical text. The main difference, in such a case, would have been the general simplicity of the work. In fact, both the process of breaking it down into meaningful text-chunks and constructing the annotation layer would have taken, in most cases, much more effort.

For example, in an early phase of this research we considered using one of David's Hume's works, the "Enquiry Concerning Human Understanding" [ref], which is also made available online by the project Gutenberg. The main difference with the Tractatus, beyond the obvious differences related to the philosophical subjects discussed, is that the "Enquiry" is not structured at all, apart from few divisions into sections and sub-sections. In particular, even the shortest sub-section available (as defined by the author) usually contains

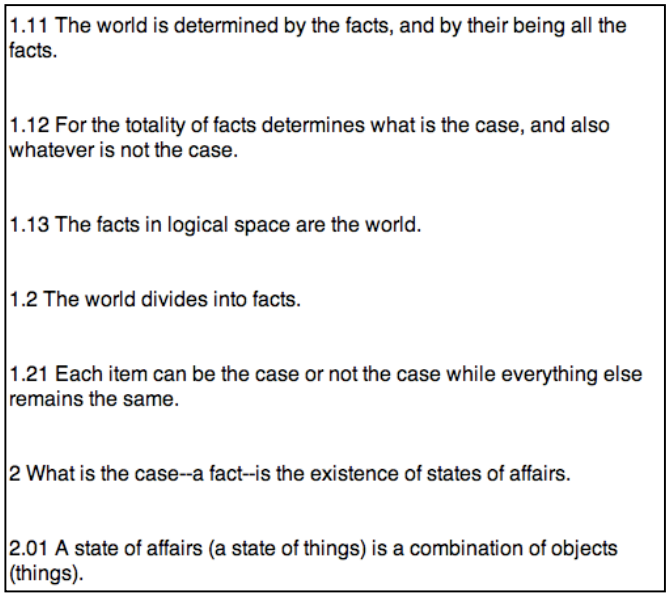
references to a large number of philosophical topics, thus making the construction of precise and interesting navigation mechanisms quite a challenging task. In other words, it would have been necessary to manually break down the various sub-sections into smaller 'units of meaning', in order for them to be annotated using the philosophical concepts in the ontology. We soon realised that the definition of such 'units of meaning' was a delicate task which, although feasible in principle, could not be carried out without the intervention of a domain expert. In fact, this task already implies an outstanding act of 'interpretation' on the source materials. Therefore, we reckoned that this was an unnecessary extra step, considering both the primary aim of our research (i.e., testing the ontological learning pathways) and the fact that other much more structured philosophical texts could have let us go straight to the annotation phase.

6.3 Knowledge base creation

The creation of the Tractatus-related knowledge base can be divided into three phases: 1) the transformation of the text itself into a 'semantic' format, that is, a format compliant with the ontological representations, using classes such as `sentence` and `expression`; 2) the annotation of the text's paragraphs; 3) the enlargement of the knowledge base through the addition of further philosophy-related instances. We will discuss these phases in turns.

6.3.1 Creating a 'semantic' Tractatus

We started by downloading the Gutemberg edition of the Tractatus, which can be found online at http://www.gutenberg.org/etext/5740. We then built a suitable parser to extract the different paragraphs' numbers and text (see fig. 6-3). In various cases, we had to manually refine the resulting text representation because of typos, wrong numeration, or reference to images that were not included in the Gutenberg edition (a summary of such 'mistakes' can be found online at http://PhiloSurfical.open.ac.uk/tractatus/info-gutenberg.html).


1.11 The world is determined by the facts, and by their being all the facts.

1.12 For the totality of facts determines what is the case, and also whatever is not the case.

1.13 The facts in logical space are the world.

1.2 The world divides into facts.

1.21 Each item can be the case or not the case while everything else remains the same.

2 What is the case--a fact--is the existence of states of affairs.

2.01 A state of affairs (a state of things) is a combination of objects (things).

Fig. 6-3 - The Tractatus text as it appears in the Gutenberg electronic edition

For a lengthier discussion about the semantic representation of the Tractatus please refer to section 5.4.2. Here is an example of an instance representing a text's paragraph:

```
(def-instance sentence-7 sentence
 ((part-of-expression Tractatus-pears-english-version)
  (has-form written-english)
  (has-number-reference 7)
 (has-string-content "What we cannot speak about we must pass over in
silence.")))
```

The text of the Gutenberg edition corresponds to the english translation made by David Pears and Brian McGuinness in 1961. According to our ontological categories, this translation has been instantiated as an `expression` of a `work` representing the 'most abstract' propositional content common to every translation of the Tractatus. That is, in OCML:

```
(def-instance Tractatus-pears-english-version expression
 ((was-made-by David-Pears Brian-McGuinness)
  (has-title "Tractatus Logico-Philosophicus")
  (has-date 1961)
  (realizes Tractatus-Logico-Philosophicus-work )
  (has-form written-english)))
```

There are also other translations of the text: for example, another quite important one was made in 1922 by Charles Kay Ogden, with the assistance of Wittgenstein himself.

Since humanities scholars' research is very often based on the interpretation of the differences among the many editions of a text, we decided to construct a semantic representation of the Ogden edition too (a digital version of the Ogden translation can be found at http://www.kfs.org/~jonathan/witt/tlph.html). Finally, we complemented the two main English translations with the original German edition, which is also available online on different sites (the one we used can be found at http://www.tractatus.hochholzer.info/).

Therefore, the instances representing the paragraph mentioned above, in these two other translations, would be the following ones:

```
(def-instance sentence-7-ogden-translation sentence
 ((part-of-expression Tractatus-ogden-english-version)
  (has-form written-english)
  (has-number-reference 7)
   (has-string-content "Whereof one cannot speak, thereof one must be
silent.")))

(def-instance sentence-7-german sentence
 ((part-of-expression Tractatus-original-german-version)
  (has-form written-german)
  (has-number-reference 7)
 (has-string-content "Wovon man nicht sprechen kann, dar&#252;ber muss
man schweigen.")))
```

In section 6.5.2 we will show the details of how the different editions of the text are retrieved and presented to the user.

6.3.2 Annotating the Tractatus' contents

For the annotation phase, we worked in collaboration with a Wittgenstein scholar who had already worked on the Tractatus. Essentially, we went through all of the text's paragraphs with the purpose of extracting the key-concepts they are dealing with. We then drew a map (using an open-source mind-mapping tool, Freemind (Freemind, 2008)) where it is possible to see the association of each concept to the paragraphs where it is mentioned in (see fig. 6-4).

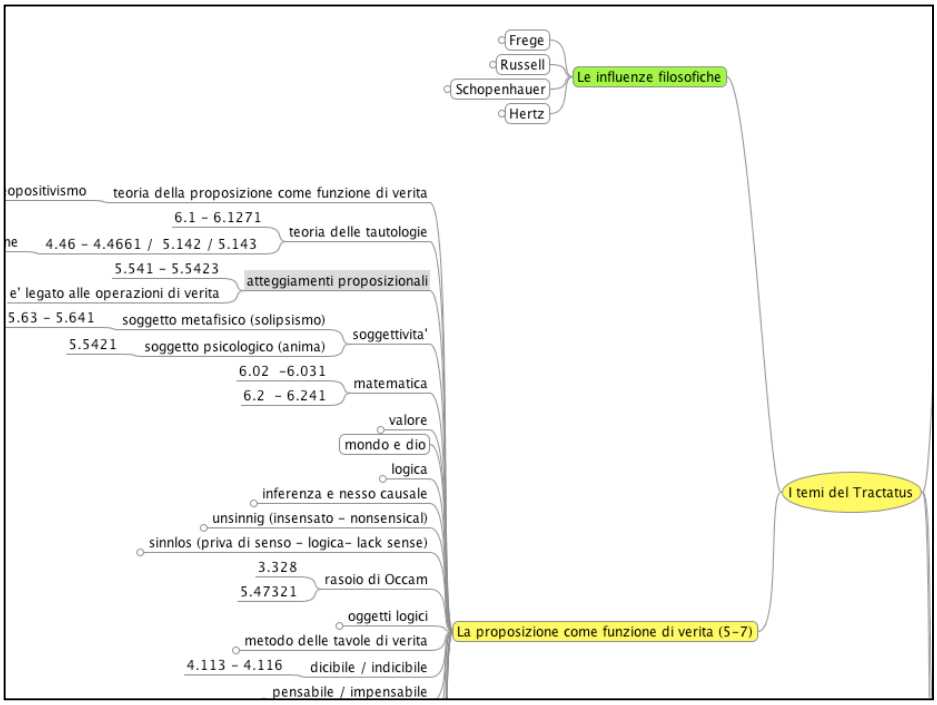


Fig. 6-4 - Extract of the Freemind map with the Tractatus concepts

During this process, our philosophy expert also created some basic *relations* that contextualize the concepts with respect to one another, so to form links among them (inclusion, opposition, similarity...). Moreover, we annotated a number of specific relationships the concepts entertain with other types of philosophical entities (e.g., a theory *belongs to* a *school of thought*, a theory *defines* a *concept*, an author *belongs to a philosophical school*, etc.).

This process generated a first layer of *interpretations* on the Tractatus. As we will see, these interpretations provide the main means used for generating learning narratives (compare section 6.6).

So, in general, this aspect of the annotation has been performed manually. We opted for this solution in order to ensure high-quality results.

An alternative approach would have been to use automatic ontology-population techniques, of the kind used, for example, in (de Boer et al., 2006). However, from an initial analysis of the text we noticed that many of the paragraphs' 'topics' could not be found in the paragraphs as 'words'. For example, paragraph 7 has been annotated with the idea 'mysticism', but such word does not appear in its text. As a consequence, we reckoned that the usage of automatic extraction techniques would have not been straightforward in this case.

For this reason we used information extraction technologies only when populating the ontology with other types of data (see section 6.3.3).

Having created a map of the concepts related to all the paragraphs of the Tractatus, we then started encoding this information in a format compliant with our ontology's representations. In particular, from the semantic point of view, the association of an idea to a paragraph of the Tractatus is achieved thanks to the `interpretation` class (see section 5.3.5.4).

So for example, in order to instantiate the fact that 'paragraph 7' is about the concept of 'mysticism' we followed two steps: 1) creating an instance representing the concept we want to refer to (in this case, ' mysticism' is an instance of `school-of-thought`); 2) creating an instance of `expression-interpretation` where we store the relation between the idea and the paragraph mentioning it (obviously, according to an *interpreter*).

Thus, in OCML we have the following:

```
(def-instance Mysticism-instance-g7685 School-of-thought
 ((has-description "Philosophical doctrine that advocates the reach of
knowledge in non-rational and unconventional ways.")
  (has-common-name  mysticism)))

(def-instance int-7-G20612  expression-interpretation
 ((carried-out-by michele-pasin)
  (interprets sentence-7)
  (has-interpretation   Mysticism-instance-g7685)))
```

Moreover, we also created other `interpretation` instances for stating the relations linking this idea to other ideas. For example:

```
(def-instance school-interpretation-g13547 school-interpreation
  ((has-related-concept  silence-by-wittgenstein  ladder-metaphor-by-
wittgenstein)
  (opposes-view rationalism empiricism)
  (classifies-view Sinoza-mysticism Eckhart-mysticism)))
```

In total, by using this method we created 1591 instances representing Tractatus' sentences, 639 interpretations of such sentences, 434 instances of philosophical ideas related to the text and 290 interpretations of the ideas.

6.3.3 Enlarging the knowledge-base

Since a larger number of instances would have implied a wider range of ways to traverse the semantic space (i.e. of *learning pathways*), we imported in the PhiloSurfical knowledge-base various other philosophy-related structured data. This was done automatically, mostly by 'scraping' the relevant information from websites in the public domain. Afterward, this data was evaluated and sometimes refined manually.

In general, the data imported was transformed into instances of the following classes: `person`, `event`, `interpretation`, `philosophical-idea` and `electronic-publication`. More precisely, this process can be summarized as follows:

1. We imported more than 7000 instances of philosophers from the Philosophy-Tree website (these instances are available in pdf format at https://webspace.utexas.edu/deverj/personal/philtree/philtree.html). Also, since the site is providing information describing the PhD advisor/tutee relationships among the philosophers, we took advantage of this feature by transforming such information into the appropriate ontological representations. Accordingly, we created a series of instances of `learning-at-institution` events (more than 7000 instances), in which we stored the details of the teacher/student relationship. For example we can have:

```
(def-instance  learning-event-G20992 learning-at-institution
 ((degree-of-study phd)
  (has-learner  Stephen-Kleene)
  (has-teacher  Alonzo-Church)
  (has-subject-area  philosophy)))
```

2. From Wordnet (Fellbaum, 1998) and other philosophical websites we imported a number of instances of `school-of-thought` (e.g., “pacifism” or “animism”), `branch-of-philosophy` (e.g., “aesthetics” or “logic”) and `scientific-discipline` (e.g., “biology” or “geometry”).

3. From various Wittgenstein-related websites we imported instances of `expression` and `philosophical-work` relevant to Wittgenstein and the

philosophy of language in general (e.g., Wittgenstein's "Remarks-on-Logical-Form-expression" or Frege's "The-Foundations-of-Arithmetic").

4. From the meta-encyclopedia of philosophy (http://www.ditext.com/encyc/frame.html) we imported information regarding philosophical entries (more than 5000) and urls linking such entries to other web-resources. Since these entries were not structured, we imported them simply as instances of `electronic-publication`. Moreover, we stored the information regarding the *title* of the encyclopedic entry as the (string) value of the *is-about* property of an `interpretation` instance. In other words, an instance of `publication` having title 'Charles Babbage' is modeled as *being-about* the string-content "Charles Babbage". The purpose of this is twofold: on the one hand, this realizes the ideal detachment between an information object and its interpretations; on the other hand, by using a string-matching algorithm, we were able to employ this information during the creation of the learning pathways (cf. section 6.4.3). These instances were formalized in OCML as follows:

```
(def-instance dict-of-phil-of-mind-g3753 electronic-publication
 ((has-physical-medium computer-medium)
  (was-made-by dictionary-of-the-philosophy-of-mind-website)
  (has-uri "http://philosophy.uwaterloo.ca/minddict/babbage.html")
  (is-identified-by "dictionary-of-the-philosophy-of-mind entry about babbage, charles")))
```

```
(def-instance interpretation-g3754 io-interpretation
 ((interprets dict-of-phil-of-mind-g3753)
  (is-about-entity "babbage, charles")
  (carried-out-by michele-pasin)
  (is-identified-by "interpretation of an fragment of
dictionary-of-the-philosophy-of-mind as about babbage, charles
")))
```

In other words, the formal characterization of the meta-encyclopedia of philosophy instances implicitly adopts a *title implies content'* assumption. In other contexts, this assumption could be a matter of debate. However we must remember that in our case we are storing such information as `interpretation` instances, not in 'absolute' terms. In other words, we are formalizing the fact that this is a claim by a specific user (represented by the value of the property *carried-out-by*). People may disagree with this assumption - but this is exactly the situation we intend to support by means of the `interpretation` class.

In conclusion, it is worth underlining that the decision of building the knowledge base was mainly guided by the practical need of testing our ontology-based application. Ideally, the data required for PhiloSurfical to run should be available on the Semantic Web: in such a case the application would have been relying much more on mechanisms such as the querying of distributed resource and metadata providers (e.g., through the SPARQL language (W3C, 2007)), as opposed to using an internal knowledge-base.

Unfortunately, this is not the case yet, especially with philosophical data. However, we must acknowledge the fact that more and more structured data are entering the web every passing day, thanks to projects such as DBpedia (Suchanek et al., 2007) (a structured version of the Wikipedia) or Freebase

(Freebase) (a community built database of the world's information). Therefore, we believe that in the near future it will be possible to rely 'directly' on a number of specialized semantic repositories of *philosophical* resources.

6.4 System architecture

Generally speaking, PhiloSurfical is a lisp web-application running on the Lispworks environment (Lispworks, 2008).

More specifically, we can describe its architecture by highlighting 6 different components:

1. The ontology and the knowledge base: these are entirely managed by OCML (Motta, 1999), a platform for knowledge representation and reasoning, which runs on all common-lisp environments.
2. An ocml-to-rdf/owl translator: this is a lisp library we created for guaranteeing the interoperability between the PhiloSurfical's knowledge base and other Semantic Web standards.
3. A narrative component: this is where the functions formalizing the 'learning pathways' (described in sections 6.6) are stored and called from.
4. The Hunchentoot server (Weitz, 2008): this is an open source http server written entirely in lisp. We use it to manage all the communication between the browser and the application back-end.

5. The PhiloSurfical 'core' application: this is responsible for the creation of the html pages, which are usually constructed by 'filling up' a set of pre-defined templates with the appropriate OCML and narrative functions' results.
6. The *ajax* component: a set of javascript functions responsible for the asynchronous updating (Holdener, 2008) of the user interface.

A graphical representation of the interaction among PhiloSurfical's modules can be seen in figure 6-5. When a user performs an operation on the browser's interface, the ajax component generates a call to the lisp back-end. This http call is handled by Hunchentoot, by means of a 'dispatch table' which associates a set of url-names to specific lisp functions. At this point, the PhiloSurfical core application decides which OCML and/or narrative functions to use in order to retrieve some data from the knowledge base, format them and then pass them back to the browser via Hunchentoot. Finally, the ajax component handles this html/css result by putting it in the right page position.

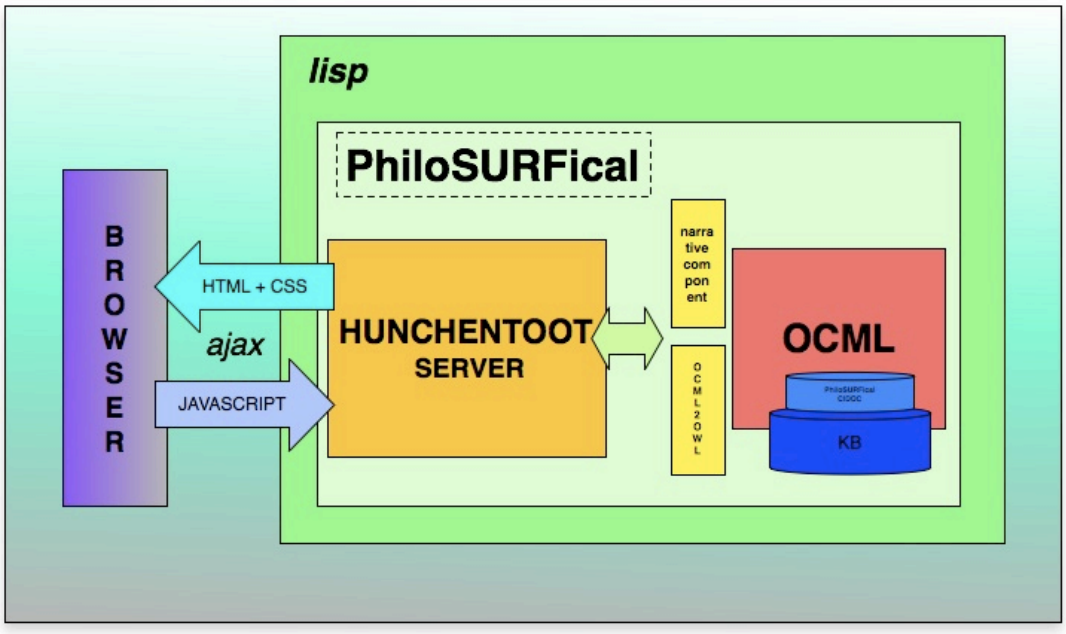


Fig. 6-5 - Technical architecture of PhiloSurfical

6.5 User Interface

For that regards the user interface, we aimed at designing an easy-to-use user interface, which combines a detailed visualisation of the text together with a comprehensive view of the different types of metadata associated to it.

We agree with the authors of (Bach and Manion, 2001) when they point out that:

> The user interface is one of the most important factors in the success or failure of a hypertext, multimediated learning environment. […] Despite these observations, the user interface is one of the least developed aspects of most web-based philosophical works. The rush to create online materials and powerful search engines to navigate through these materials has overshadowed the importance of interface design.

At a first glance, the PhiloSurfical tool can be described as organized into five sections or *tabs* (see fig. 6-1).

We attempted to organize the tabs' sequence according to their increasing difficulty of usage (namely, the first tab requires less 'learning effort' than the second one, the second one less than the third one, etc.). By doing so, we wanted users to have a more 'gradual' encounter with the software. This becomes important especially when considering that not all Wittgenstein's scholars are familiar with web-based educational tools.

Fig. 6-6 - PhiloSurfical tab1 - Welcome page

The five tabs can be briefly described as follows: the *Welcome* tab serves as a 'splash screen' and provides some contextual information and links to relevant resources; the *Browse the text* tab presents three translations of the Tractatus' text in an interface that lets users easily control which translation to use; the *Browse the annotations* tab supports a different type of text navigation by means of a smart-index of the topics associated to the Tractatus' fragments; the

Browse the pathways tab lets users select topics of interest and explore related resources by means of the 'learning pathways' facility; finally, the *Browse the ontology* tab visualizes the tree-hierarchy of the ontological representations PhiloSurfical relies on, allowing users to examine the underlying complexity of the model.

We will now discuss each of the tabs in more details.

6.5.1 Tab 1: *Welcome page*

This section (see fig. 6-6) does not provide any specific functionality for navigating the Tractatus, but it simply explains the purpose of the prototype to the user. It also links to various external learning resources about Semantic Web related topics, which should help users in understanding both the terminology and the functioning of PhiloSurfical.

6.5.2 Tab 2 – *Browse the text*

In this tab users can simply browse the text, which is made available in three versions (the original German edition and the two major English translations).

In order to facilitate this activity, a tree-like outline of the book on the left hand side lets them jump quickly to a specific paragraph. This type of outline is very handy especially in the case of the Tractatus, because of its highly hierarchical structure. As a result, we can quickly have an overview of the text as a whole. As previously discussed (cf. section 6.2.1) the use of other philosophical texts

would have required extra work in order to achieve such a clear-cut hierarchical organization (and possibly, not always with such good results).

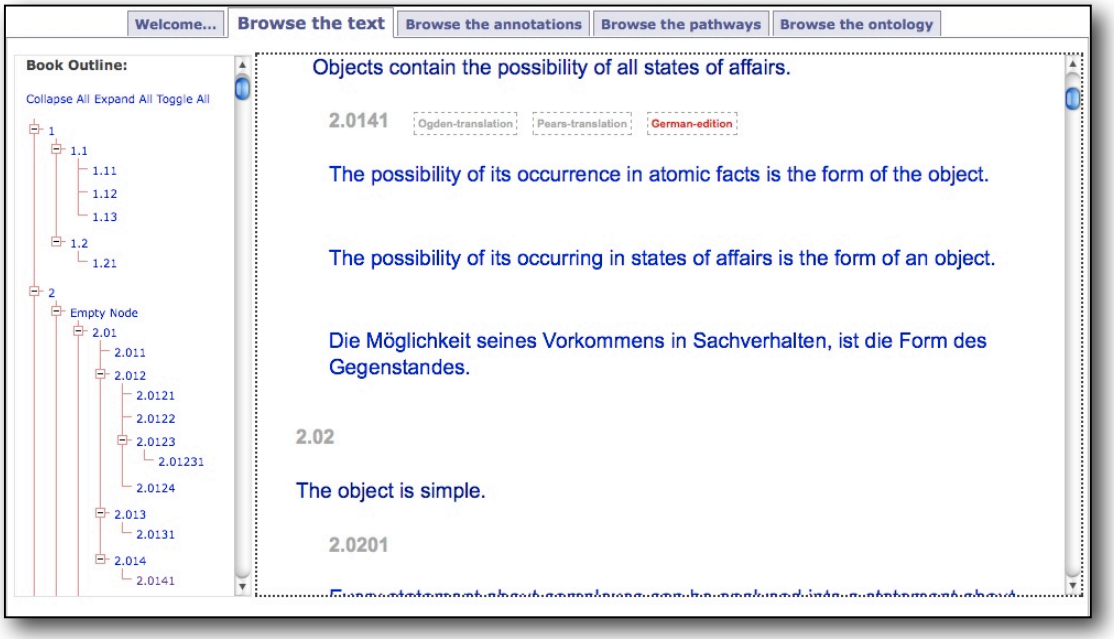


Fig. 6-7 - PhiloSurfical tab2 - Browse the text

An interesting feature in this tab is a javascript mechanism by which it is possible to select what Tractatus' version to visualize: when the mouse hovers one of the paragraphs, this is highlighted and a contextual menu appears above the text. By clicking on one of the available options, it is possible to view more than one translation at the same time (as shown in Figure 6-7, with the proposition 2.0141).

At the ontological level, this is achieved by means of a simple function that retrieves the different `expressions` of a `work`.:

```
(defun sentence-string-content (sentence &optional (expression
'Tractatus-pears-english-version))
 (if (setofall '?x `(sentence ,sentence))
    (let ((content ""))
      (setf content (format nil "~a"
        (first
         (setofall '?x `(and (sentence ,sentence)
                             (part-of-expression ,sentence ,expression)
                              (has-string-content ,sentence ?x))))))
      content)))
```

6.5.3 Tab 3 – *Browse the annotations*

This is where the ontological backbone of PhiloSurfical starts becoming more evident.

At the centre of the screen users can still read the text, but now four panels, two on the right side and two on the left side, provide alternative ways to navigate the Tractatus.

We can imagine this functionality as a ‘smart index’ of the text. That is, an index of the text’s topics (i.e. the ideas being dealt with in the text) that is dynamically updated, depending on what text fragment the user is focusing on.

In order to understand how this works let us briefly go through, once again, the underlying ontological representations.

All the text fragments are represented as instances of `information-object` (specifically, they are instances of the class `sentence`). Each one of them has associated one or more annotations, that is, they have been interpreted by some experts in our team as *being-about* certain topics. At the ontological level, this has been carried out thanks to another class, `expression-interpretation`,

whose instances express the connection between an `information-object` (or part of it) and a `propositional-content` .

As a result, for one text-fragment we can have *multiple* interpretations. In particular, each one of them is represented through a different instance which is totally independent from the other ones and, obviously, from the instance representing the text-fragment itself.

So, for example, paragraph 2.1 can be 'linked' to the philosophical idea of "picture". Similarly, other paragraphs could also be associated to this annotation. Thus whenever we focus on the topic "picture" we should be able to retrieve also the other *relevant* text fragments.

It is important to remember, at this point, that all the Tractatus' interpretations we created are clearly just *our view* of the text's meanings. In other words, they are just some possible interpretations which, although we hoped being significant and thus capable of helping learners in understanding the text, could have been done differently. In other words, the interpretations are used here mainly as a way to showcase the functionalities of the tool (actually, in future releases, we would like to create an interface that supports users in creating *their own* annotations and possibly also sharing them).

Let us now describe these features in more details, by going through the major functionalities of tab3. We can do so by imagining a real-world situation: e.g., a student who wants to know more about the notion of "picture" in Wittgenstein.

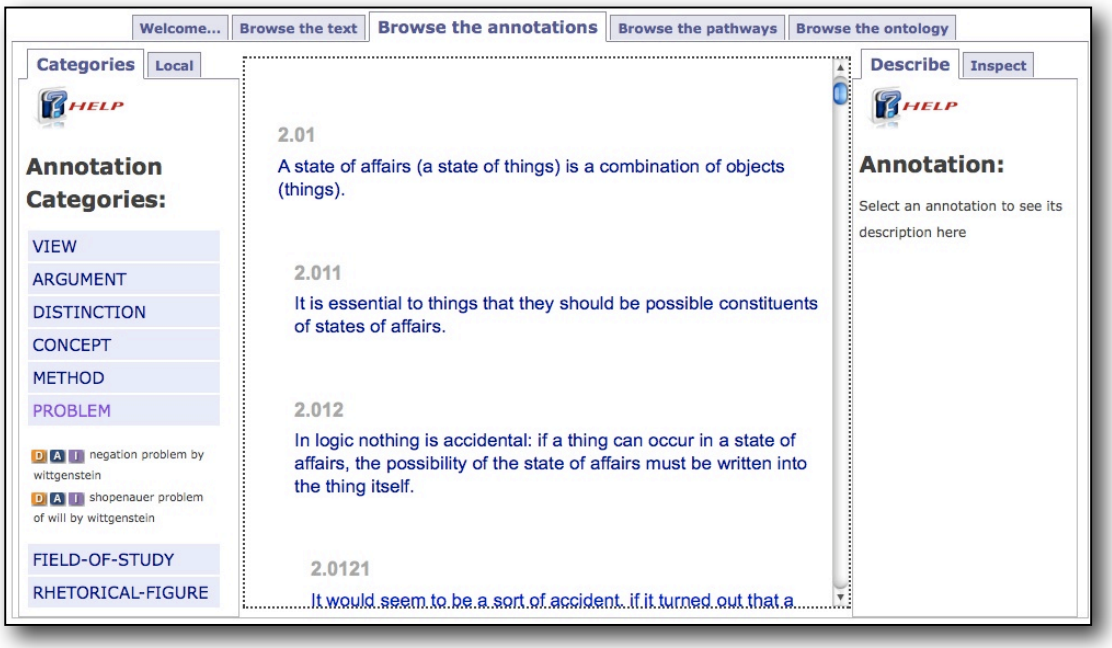


Fig. 6-8. PhiloSurfical tab3: the eight types of philosophical ideas (on the left)

The student could start by looking up “picture” in the **categories panel** (see figure 6-8, on the left). Here we find all the topics associated with the text, organized into the eight idea-categories defined in the ontology (as described in section 5.5). In other words, this panel acts as a *general index* where users can find quickly the contents they are interested in, or simply browse them to see what is available. In particular, the three little icons next to each topic serve to trigger operations on the other panels.

For example, by clicking on the ‘A’ button (= annotations) our student can update the visualization in the central column, so that it will show all the text-fragments interpreted (= annotated) as being about the concept of “picture” (see figure 6-9, central column). This is the first functionality this tab is offering to the user: by selecting a topic, it is possible to see only the sections of the Tractatus that have been associated to it.

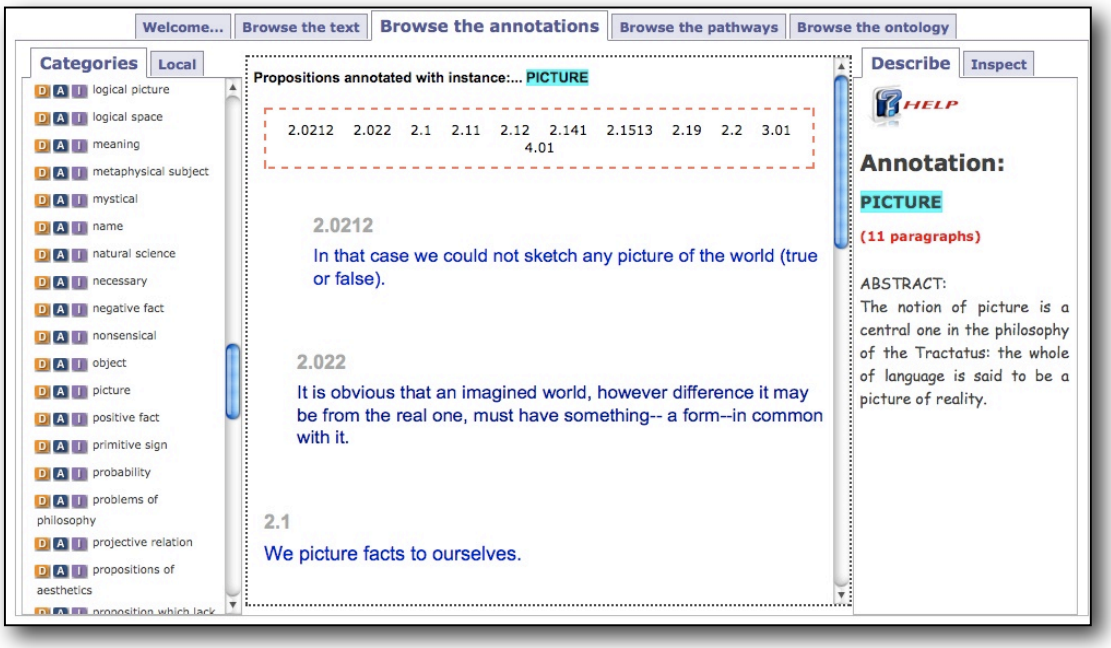


Fig. 6-9. PhiloSurfical tab3 - Browse the annotations ('categories' and 'describe' panels)

Secondly, by using the 'D' button (= describe) the student can read a short description of the topic just selected in the **describe panel** (see fig. 6-9, on the right). This functionality aims at giving learners a first introduction to the selected topic. The description is stored in the knowledge-base thanks to a generic *has-string-description* property which is associated to each instance of `propositional-content`.

Then the student may decide to start investigating the meaning of the concept of "picture" by reading the text itself. In particular, let us imagine that he/she finds particularly interesting paragraph 2.1, so he/she wants to learn more about it.

This can be achieved by clicking on the paragraph itself: in fact, by doing so the **local** annotations **panel** becomes active (see figure 6-10, on the left). Here we can see which are the interpretations associated to a single text-fragment; also,

by clicking on them, we can navigate the text in a non-linear manner. For example, our student might discover that the notion of “picture” appears also in a “picture theory of language”, therefore he/she decides to select this new annotation and explore it further.

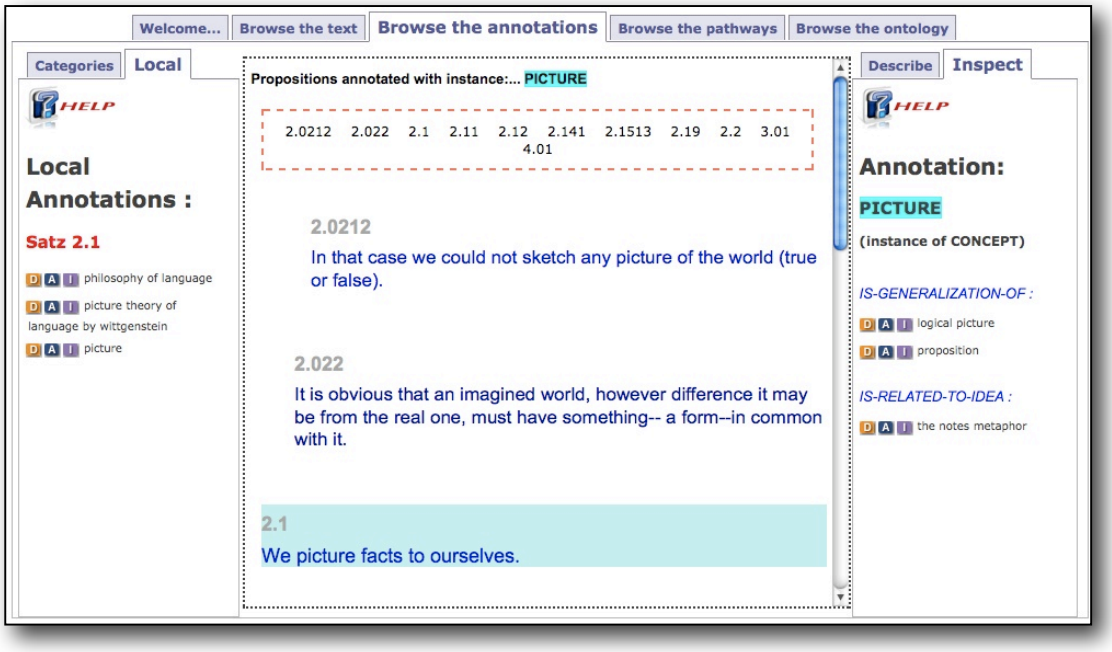


Fig. 6-10. PhiloSurfical tab3 - Browse the annotations (‘local’ and ‘inspect’ panels)

Finally, after having explored a number of ‘lateral’ navigations starting from the “picture” concept, our student might start feeling disoriented. In fact, although he/she has read about various topics which seem to be related to each other, it is not clear yet how they could all fit into a more generic *map* of Wittgenstein ideas.

The ‘**inspect’ panel** addresses this problem (see fig. 6-10, on the right). The panel, which can be activated by clicking on the ‘I’ button, shows a text-based version of a *map* of the various ideas related to “picture” (see figure 6-10, on the right). Also here, the presented ideas are clickable and could be used for starting other navigations.

In general, this last functionality is related to the fact that a certain topic (e.g., in our case, the concept of “picture”) is not just a string or an unstructured tag, but is instead an instance of one of the *types of ideas* we classified in the ontology. In this particular case, it is an instance of the type `concept`.

As a consequence, the instance has been created along with several properties (such as *name*, *description*, etc.) but, most importantly, it can also be *interpreted* in an analogous manner as the text-fragment interpretations we have seen above. Only, this time, this is achieved through a `concept-interpretation` class, which is used for describing a concept-instance using various other properties (as described previously in section 5.3.5.4).

For example, as shown in figure 6-10, the “picture” concept has been interpreted by our expert using the properties *is-generalization-of* and *is-related-to-idea*, whose values are respectively the `concepts` of “logical-picture” and “proposition” for the first property, and the `rhetorical-figure` “the notes metaphor” for the second property.

6.5.4 Tab 4: *Browse the pathways*

As previously mentioned (section 4.3), our approach takes the notion of a ‘digital narrative’ as a “system of specially stored and organized narrative elements which the computer retrieves and assembles according to some expressed form of narration” (Brooks, 1996) and attempts to transpose it within the specific scenario made up of philosophical entities. Accordingly, in the fourth

tab we attempted to create a virtual environment for building user-triggered digital narratives, which we also call ‘learning pathways’.

A ‘pathway’ is essentially a way to retrieve different instances and organize them into a coherent whole; in general, this result can be obtained thanks to the semantic relations formalized in the ontology or (in future releases of PhiloSurfical) by relying on the ‘semantic mappings’ linking the ontology to other services available on the web.

So, for example, we can have a *theoretical* learning pathway (which focuses on the contrasting relations among ideas), a *textual* one (which attempts to retrieve related information objects), a *historical* one (which keeps results in chronological order), a *geographical* one etc. Of course, these queries can also be more specialized: within the theoretical pathway, there can be a *disambiguation* one (which highlights concepts having the same name, but being actually defined by different views), a *contrast* one (which highlights opposing views) etc.

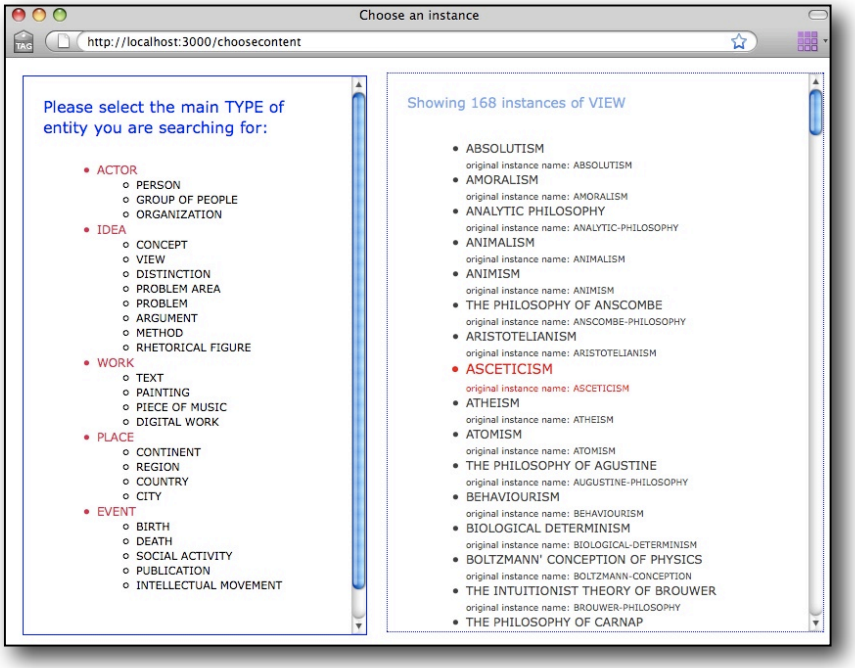


Fig. 6-11 - PhiloSurfical tab4 - Choosing the starting point of a pathway

In more practical terms, learners start by selecting any content of interest (or just use the most recently selected content, which is 'in focus' by default) so to use it as the starting point of a pathway-navigation (see figure 6-11).

Once they have chosen an instance, learners may click on one of the available choices appearing in the 'pathways list' panel (see figure 6-12). Each pathway-type has also a brief description explaining its meaning; such description is shown automatically every time the cursor moves over the items in the pathways list.

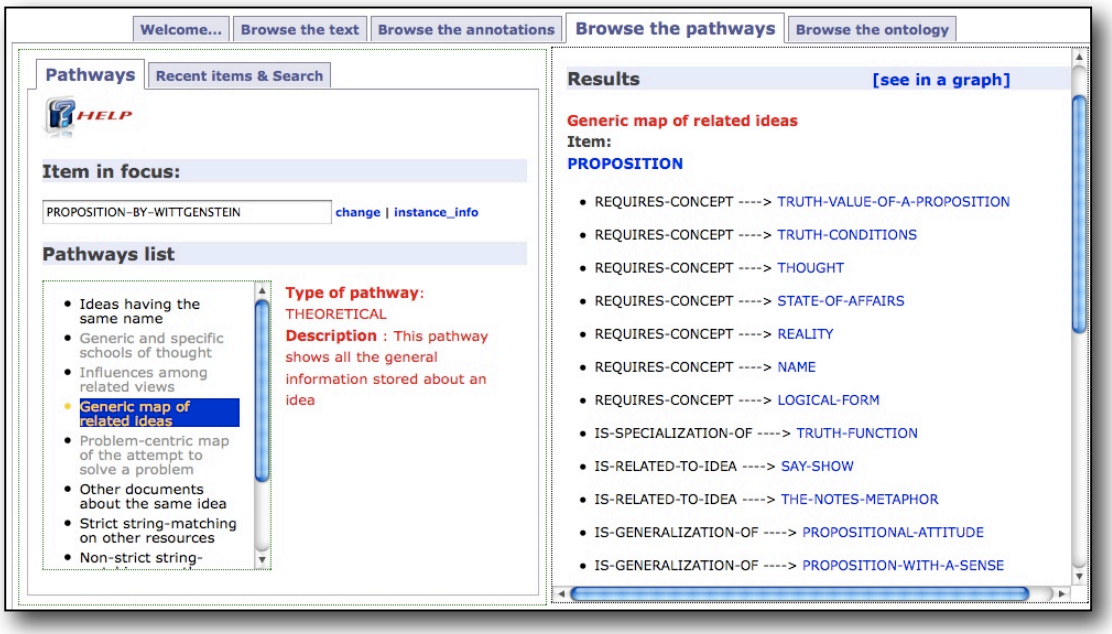


Fig. 6-12 - PhiloSurfical tab4 - Browse the pathways

Once a specific pathways gets triggered, the results are shown on the right panel as a set of interrelated entities. More precisely, results are displayed as a list of triples representing *subject-predicate-object* relationships. By doing so, the significant connections among the pathway's items are highlighted. In addition, results get organized into pathways-specific *clusters* that convey the idea of a *coherent map* to the user.

Needless to say, by clicking on any item in the results' list it is possible to put the item into 'focus', and then start a new semantic navigation - this process can also be described as a *recursive* search process.

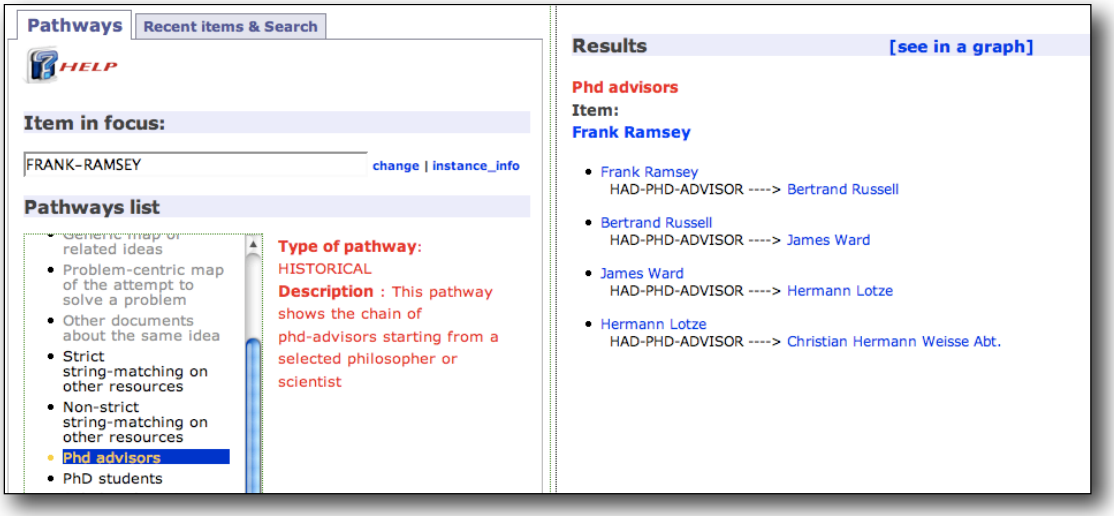


Figure 6-13 . The 'PhD advisors' learning pathway for "Frank Ramsey"

In general, pathway-triggered navigations will usually produce results that go beyond the scope of the Tractatus: for example, starting from the instance of a person named "Frank Ramsey" (as shown in fig. 6-13), by using an *historical* pathway that highlights the chain of PhD advisors, we can find out that he is in the lineage originating from "C.H.Weisse". Instead, if we started from the "problem of the foundations of mathematics" we might find more useful a *problem-centric* pathway, which is highlighting the concurrent views attempting to solve it (cfr. fig. 6-14).

It is also possible to view the pathways results' using a graphical visualization: e.g., in fig. 6-15 we can see the results of a theoretical pathway starting from the idea of "Frege's conception of logic".

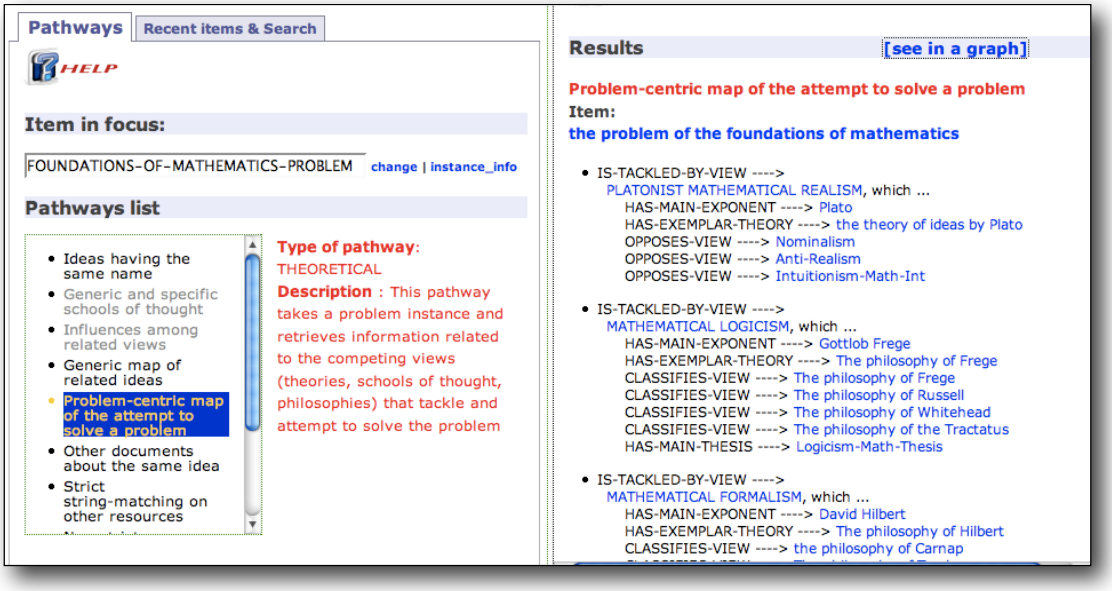


Figure 6-14. Pathway representing the various attempts to solve a problem

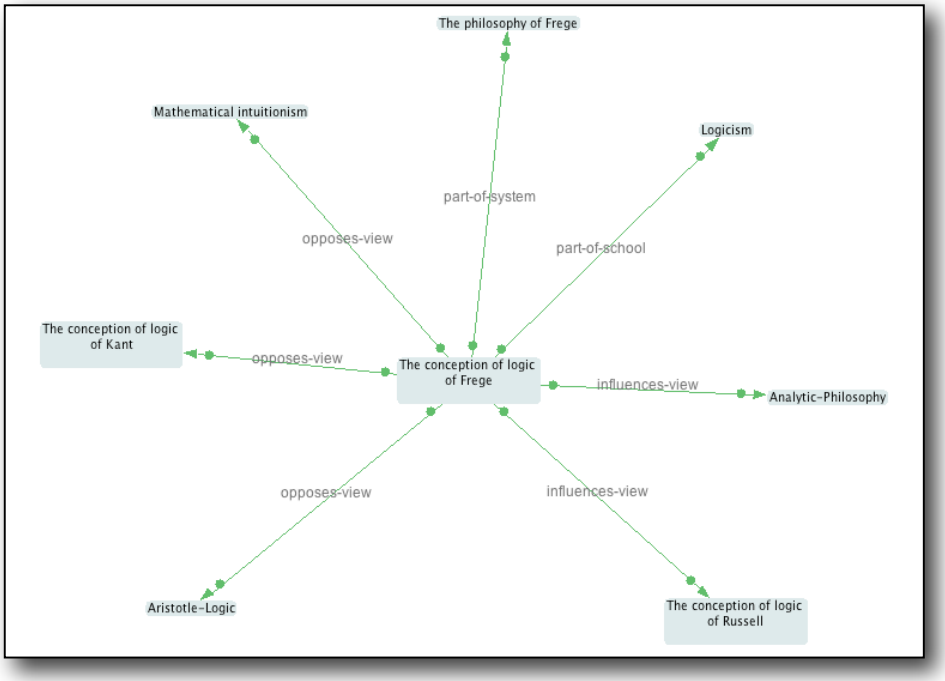


Figure 6-15. Graphical view of a theoretical pathway starting from "Frege"

Since the formalization of the pathways is an important contribution of our research work, we will give an extensive overview of this topic in a separate section, section 6.6.

Finally, we must mention that in tab-4 there is also a 'recent items' panel (see fig. 6-16). This is used mainly to keep track of all the items which have been selected since the start. Also, this panel provides facilities for searching for an item on various philosophy-related search engines.

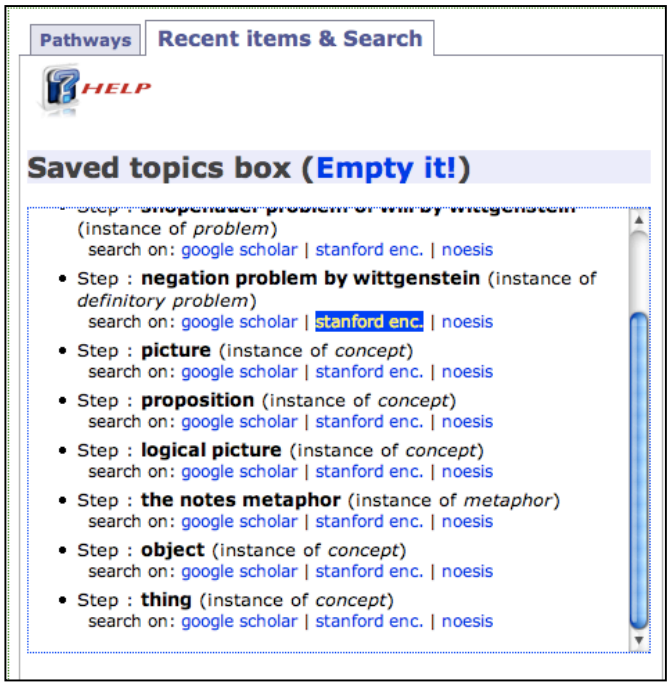


Fig. 6-16 - Recent-items and search panel

6.5.5 Tab 5 – *Browse the ontology*

This section aims at familiarizing PhiloSurfical's users with the underlying technology the application is using. On the left hand side, by means of a tree-like menu it is possible to navigate the hierarchical structure of the ontology used to represent the Tractatus and all the other entities dealt with in PhiloSurfical.

By clicking on the classes' names we can see their description on the right panel: this is composed by information about their position in the ontology (such

as what *super-classes* or *sub-classes* they have), a natural language explanation of the classes' significance and the complete list of the *properties* linking them to the rest of ontology.

For example, in the figure below (fig. 6-17) it is possible to see the description of the `conceptual-object` class.

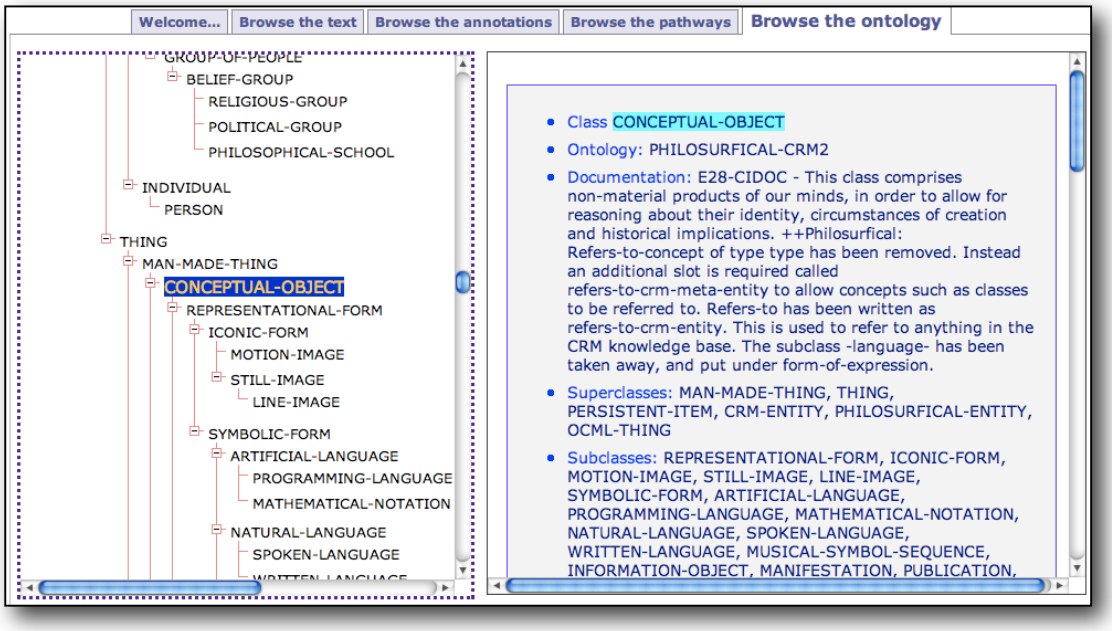


Fig. 6-17 - PhiloSurfical tab5 - Browse the ontology

6.6 Learning pathways for philosophy

In general, the envisaged context of usage which has been guiding the ontology engineering process is the following: the semantic model should support the *reconstruction* of the history of ideas, by relying on structured information about the *practical* domain and the *theoretical* domain of thinkers.

The rationale for the 'learning pathways' is, essentially, the fulfillment of this requirement: providing mechanisms for the dynamic *reconstruction* of a set of

semantically-described resources into a presentation which is both coherent and contextually relevant.

As discussed in the literature review (see section 2.4), there can be various approaches to the problem of navigating semantic spaces. For example, *faceted browsing* (section 2.4.2) is a powerful technique by which users can easily explore an unknown domain. More precisely, in this case an information-space can be browsed 'facet by facet', that is, by relying on a process of 'incremental' and 'minimal' selection (or de-selection) of the various semantic characteristics which have been associated to a knowledge domain.
In a sense, also *faceted browsing* offers navigational 'pathways'. In particular, in this case the pathways are totally *open ended,* since the only structure imposed by the system is the one of the semantic model itself (that is, there is no meta-organization of the navigational functionalities).

We initially thought about replicating a similar browsing functionality also with the PhiloSurfical knowledge base. However, we soon realized that for the pathways to be truly 'learning' ones, they could not be completely open-ended. In fact, if they were so, even if relying on the 'semantic' links among resources (as opposed to the more usual 'syntactic' ones) they would have easily reproduced well-known phenomena such as *information overload* or *inconclusive navigations* (which would be of impediment to a learner).

Thus, in order to tackle this problem we created a series of *semi-structured* ways to query the knowledge base, which reduce the number of results by

constraining them only on certain *types* of semantic relations. In other words, we formalized a number of 'generic query templates' representing the most interesting ways (from a philosopher's point of view) to browse the ontology across one of its dimensions (or more than one simultaneously).

In general, the idea that inspired us here is that the resulting organization should attempt to mimic some of the classic *ways* the discipline employs to *narrate itself.* The semantic pathways thus created are in fact replicating key *structural* aspects of the domain.

As a consequence, we expect learners to be facilitated in apprehending philosophy by increasingly grasping its distinctive 'narrative structures' (e.g., argumentative narratives, historical narratives etc.).

In total, we defined four different generic types of pathways: *theoretical, historical, geographical and textual* ones. The rationale for this choice is that these are the learning pathways that came out of the domain analysis (cf. section 5.2) we performed together with domain experts. Moreover, although it is not possible to claim that this typology is exhaustive, in practice my experience, both as a graduate philosophy student and during the discussions with several domain experts, suggests that essentially these are the four key dimensions for analysis in philosophy. Each one of them highlights different structural characteristics of the philosophical domain. Also, each one of them can support learning along a different dimension. In the following sections we will discuss these topics in more details.

Before that, however, we should spend a few words on the description of the pathways' selection mechanism.

6.6.1 Pathways selection

This mechanism can be divided in two parts: 1) the updating of the user interface whenever a user selects a new item of interest; 2) the construction and presentation of the pathway.

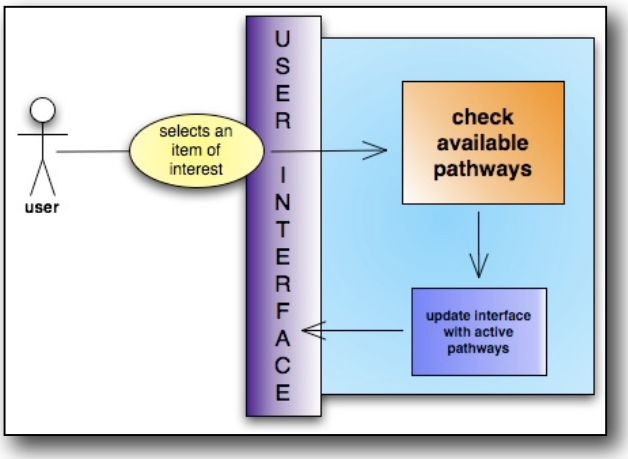


Fig. 6-18 - Schema of how the available pathways get updated

In the first case (see figure 6-18), a user triggers this mechanism every time he/she clicks on an instance, thus putting it 'into focus'. In fact, at this point the system has to update the 'pathway list' (see section 6.5.4): this is done by activating (i.e. transforming into clickable items) only the pathways which can be used with that specific instance. In general, the *check-available-pathways* routine works by checking the ontological type of an instance and comparing it with the pathways' pre-defined input-type.

The second mechanism is instead more complicated (see figure 6-19). In this case a user has clicked on a pathway-name, thus triggering the *get-pathway-results* routine. This is basically a meta-function that retrieves the relevant

pathway-function, feeds it with the selected 'starting' item and then collects the results.

But at this stage the output is simply a 'concise' representation of the final set of triples, i.e. it just contains a set of essential information which then need to be 'exploded' into a more 'explicit' representation. For example, the results of a theoretical pathway involving different concepts related to the same author may omit the repetition of the *is-author-of* property. On the contrary, the data fed into a specific visualization module needs to explicitly mention all the triples.

At the moment, the 'explosion' is done by means of two routines, depending on whether the results are presented in *html* or in the *java*-based graphical applet. In future releases of PhiloSurfical, it is likely that we will add also other types of results' graphical presentations.

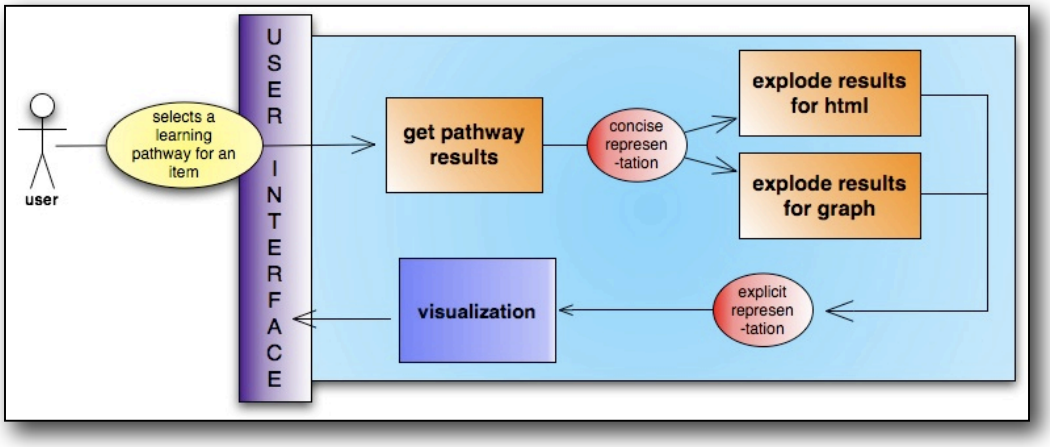


Fig. 6-19 - Schema of how a pathways is constructed and visualized

We will now describe in more details the pathways created, organizing them according to their generic *typology* (i.e. theoretical, textual, historical, geographical).

Let us underline that we are using this 'quadruple' classification mainly for *presentation purposes*. In fact, often the pathways' *types* could appear quite

'artificial', since the four dimensions are constantly intersecting each other (e.g., a historical perspective brings in elements also from the theoretical domain, and so on).

In conclusion, we ask the reader to focus on the features of each single pathway we formalized, and take the general classification only as a *possible* way to group them.

6.6.2 Theoretical pathways

Theoretical pathways are characterized by the fact that they put the accent on the *theoretical* dimension in the history of philosophy. Usually they take as an input an instance of `propositional-idea` or of one of its subclasses.

In general, theoretical pathways start from *philosophical* ideas and operate by recollecting information which has mainly to do with the 'theoretical' dimension of philosophy, i.e. with the relationships among ideas (e.g., philosophical concepts, theories, arguments etc.).

The resulting 'map' is intended to shed new light on the significance of the idea we started with, by putting it into a structured context.

Name (input type)	Description
Ideas having the same name (`propositional-content`)	This pathway populates the graph with ideas having the same name but a different meaning than the selected one. E.G., we can look for authors who talked about the idea of 'fact' (such as Frege, Russell and Wittgenstein), although all of them had different interpretations of it.

Name (input type)	Description
"Generic and specific schools of thought" (`school-of-thought`)	Starting from a school of thought, this pathway retrieves a set of related schools of thought which are all specializations of the same general one. This pathway is related to the formalization presented in section 5.5.5: e.g., by focusing on 'atomism' we would be able to see the related *contextual* versions of it, such as 'logical atomism' , 'metaphysical atomism', 'social atomism', etc.
"Influences among related views" (`view`)	Starting from a view, this pathway is a *recursive* function showing information about others views that support/compete with the first one. E.G., starting from Wittgenstein's theory of language, we could go to the Russel's theory of language (which opposes it), then to Whitehead's theory of logic (which supports Russell's) etc.
"Generic map of related ideas"(`propositional-content`)	This pathway shows all the general information stored about an idea. E.G., from any idea, we would see all its *interpretations*, such as *opposed-to, similar-to, causes-concept,* etc.
"Problem-centric map of the attempts to solve a problem" (`problem`)	This pathway takes a problem instance and retrieves information related to the competing views (theories, schools of thought, philosophies) that tackle and attempt to solve the problem.

Table 6-1. The theoretical pathways

Let us see in more details how these pathways have been formalized. We must remember that a pathway is essentially a query-model to be used on any knowledge-base compliant with our ontology. So, for example, the *'ideas having the same name*' pathway accepts an instance of `concept` as argument and retrieves all instances of `philosophical-idea` whose *has-common-name* slot value is the same as the one of the first `concept` (cf. the 'find-ideas-with-name' routine below). If results are found, each one of them is returned together with the value of its *defined-by-view* property:

```
(defun th-ideas-with-same-names (idea)
"From an idea (concept) instance ----> outputs other ideas having the
same name, with the respective defining views"
 (if (has-granfather? idea 'concept)
    (let ((other-ideas (find-ideas-with-name (common-name idea)))
          (p '()))
      (if other-ideas
          (dolist (eachone other-ideas)
                   (push (list eachone (first (my-slot-values eachone
'defined-by-view))) p)))
      p)))

(defun find-ideas-with-name (name)
"Just returns a LIST of the instances of ideas with a given COMMON-
NAME"
 (let ((ideas (setofall '?x `(and (philosophical-idea ?x) (has-common-
name ?x ,name)))))
   ideas))
```

A more complex example is the *''influences among related views*' pathway. In this case we have a recursive function that checks a number of predefined slots on `view` instances (i.e., *influences-view*, *influenced-by-view*, *opposes-view*, *supports-view*). When the function is started with a `view` instance *v1*, it will retrieve all the other theories or schools of thought which are related to *v1;* the process will be recursively repeated also for each one of them depending on the *maxlevel* setting (below it is set to 2). It is important to remember that in the PhiloSurfical knowledge base all relationships among philosophical ideas are stored as `interpretation` instances (cf. section 5.3.5.4). That is why in the code below we are making us of a specific function, `find-idea-interpretation-content`, in order to retrieve all the interpretations about an idea. This function can accept as an argument also the *author* of the interpretation - thus retrieving, for example, only the interpretations made by a specific user. This further functionality is not currently used in PhiloSurfical, but we expect that in future

versions of the tool it will support its transformation into a *collaborative* learning environment.

```
(defun th-views-influences (view)
“Recursive pathways that retrieves a network of views influencing each
other”
(let ((out '())
      (level 0)
      (maxlevel 2))
  (setf out (th-views-influences-internal view level maxlevel))
  out))


(defun th-views-influences-internal (view level maxlevel)
(let ((out '())
      (ints (find-idea-interpretation-content view)))
  (if ints
      (dolist (int (first ints))
        (let ((rel (first int))
              (vals (second int)))
          (if (or (equal rel 'INFLUENCES-VIEW)
                  (equal rel 'INFLUENCED-BY-VIEW)
                  (equal rel 'OPPOSES-VIEW)
                  (equal rel 'SUPPORTS-VIEW))
              (push (list rel
                          (let ((rec '()))
                            (dolist (val vals)
                              (push (list val
                                          (if (< level maxlevel)
                                                  (th-views-influences-
internal val (+ 1 level) maxlevel)
                                            nil)) rec))
                            rec))

                          out)))))
  out))
```

6.6.3 Textual pathways

Textual pathways are characterized by the fact that they focus on ‘texts’, intended in a broad sense (i.e. cultural objects such as books, audio or video

resources, paintings etc.). That is, from the ontological point of view, instances of `information-object`, `work` or `manifestation` (cf. section 5.4).

In general, by using a textual pathway we expect learners to be able to find other textual resources about a topic, to compare the publication year of related texts, or to follow 'hidden' relationships among philosophers (or scholars in general) by investigating the relations among their publications (see below).

Name (input)	Description
"Other documents about the same idea" (`propositional-content`)	This pathway retrieves other information objects (beyond the Tractatus) which deal with a specific idea, only by using stored semantic relationships. The results are presented with an historical ordering.
"Strict string-matching on other resources" (`PhiloSurfical-entity`)	This pathway combines semantic and syntactic data; it displays other resources where the value of the *is-about* attribute (usually a string) matches *almost exactly* the name of the content searched. We introduced this pathway for querying poorly structured data (such as the ones presented in section 6.3.3). Since we are using a string matching algorithm, results may be noisy.
"Non-strict string-matching on other resources" (`PhiloSurfical-entity`)	This pathway is similar to the one presented above; the main difference is that it searches for resources where the value of the is-about attribute matches *at least 50 per cent* of the name of the content searched. As a consequence, this pathway may produce a *large* number of unwanted results.
"Textual lineage of references among works" (`information-object`)	This *recursive* pathway takes an instance of information object (e.g., a book or a painting) and explores the interpretations about it, looking for patterns expressing relations to other information objects. E.G., from a painting of Matisse, we could go to a related work of Sartre (interpreted as 'inspired' by the painting), then to a work of Kierkegaard which was cited by Sartre's, etc.
"Production of an author" (`person`)	This pathway retrieves information about all the *works* (in a broad sense) produced by a scholar in his/her lifetime. Results are presented with an historical ordering.

Table 6-2. The textual pathways

As an example of textual pathways formalization, we can see below the code for '*other documents about the same idea' .* Essentially, in this case we are looking for all instances of `information-objects` that have been interpreted as being about a specified `propositional-content`:

```
(defun txt-also-has-idea-as-subject (idea)
"Retrieves other information objects (beyond the tractatus) which deal
with a specific idea"
(if (has-granfather? idea 'propositional-content)
    (let ((other-docs (setofall '?x
                    `(and (io-interpretation ?i)
                          (has-interpretation ?i ,idea)
                          (interprets ?i ?x)
                          (not (PART-OF-EXPRESSION ?x Tractatus-pears-
english-version )))))
          (out '()))
      (if other-docs
          (setf out (list idea other-docs)))
      out)))
```

6.6.4 Historical pathways

Historical pathways are characterized by the fact that they underline the importance of the *temporal* dimension in the history of philosophy. From the ontological point of view, these pathways usually rely on the various features of the `event` class and its specializations.

Usually, by using an historical pathway a learner is facilitated in apprehending the course of events leading to a famous fact in philosophy, or in seeing the evolution of an idea in time.

Name (input)	Description
"Chain of Phd advisors" (`person`)	This *recursive* pathway shows the chain of phd-advisors starting from a selected philosopher or scientist. For example, starting from Edmund Husserl we would arrive to Otto Mencke Abt. through eight difference generations.
"Chain of Phd students" (`person`)	This *recursive* pathway shows the chain of phd-students starting from a selected philosopher or scientist. For example, starting from Karl Popper, we would arrive to Alan Musgrave through Jeremy Shearmur.
"Events in a lifetime" (`person`)	This pathway recollects all the information concerning events related to scholar, presenting it in a timeline.
"Causal chain of events "(`event`)	This generic pathway, starting from an event, looks in the knowledge base for events which have been connected to it through the *causally-connected-to* property. For example, from the event 'death of Socrates' we could be taken to the 'accusation of Socrates by Athens'.
"Historical perspective on influences among views" (`philosophical-idea`)	This pathway starts from a philosophical view (e.g., a theory or school of though) and shows the chain of influences (i.e. values of the property *supports-view*) by putting the results within a timeline.

Table 6-3. The historical pathways

For example, with the ocml code below we represented the recursive pathway named '*chain of PhD advisors'.*

```
(defun hs-advisors-tree (person)
"It outputs a list, first the starting person, then the ascending
tree"
(if (has-granfather? person 'person)
    (let ((tree (next-advisor-up person))
          (out nil))
      (if tree
          (setf out (append (list person) tree)))
      out)))
```

```
(defun next-advisor-up (person &optional (list nil))
"Takes the instance-name of a person, and outputs a list of the
advisors tree"
  (if person
      (let ((advisor (first (setofall '?x
                              `(and (person ,person)
                                    (learning-at-institution ?l)
                                    (degree-of-study ?l PHD)
                                    (has-learner ?l ,person)
                                    (has-teacher ?l ?x))))))
        (push advisor list)
        (next-advisor-up advisor list))
   (reverse (rest list))))
```

6.6.5 Geographical pathways

Finally, geographical pathways are characterized by the fact that they are centered around the *spatial* dimension in the history of philosophy.

So, for example, learners may explore how scholars born in the same area (region, state or continent) dealt with a chosen philosophical idea (e.g., a problem or a distinction). Or they could investigate how well-known a text was in a particular geographical area, so to formulate hypotheses on philosophers' influences.

Name (input)	Description
"Scholars born in same area" (`person`)	This pathway retrieves information about the birthplace of a person, and shows other scholars who come from the same area.
"Geographical perspective on an idea" (`philosophical-idea`)	Starting from a philosophical idea, this pathway retrieves information about ideas related to it (i.e. *supporting, contrasting,* or *dependent* on it), organizing the results according to the geographical provenience of the ideas' authors.
"Life traveling of a scholar" (`person`)	This pathway gathers information about all the trips a person has done during his/her life.

Name (input)	Description
"Use of a text in a geographical area" (`information-object` or `manifestation`)	By focusing on a specific text, this pathway attempts to highlight evidences of text's usages within a specific geographical area.
"Scholars and philosophies in a geographical areas " (`geo-political-area`)	Starting form a geographical area, this pathway shows a list of the scholars and/or philosophies which have originated in that area.

Table 6-4. The geographical pathways

For example, the formalization below is the one used for the *'scholars born in the same area'* pathway. Notice that since the PhiloSurfical knowledge base includes only a minimal amount of details about geographical areas (e.g., various countries and some cities, without may relations among them), currently this pathway is just checking whether two persons produce an exact match on the *has-birth-place* slot value. In future versions, this functionality can be easily extended so to include some reasoning on the *topological* relationships of places.

```
(defun geo-persons-birthplace-same (person)
  (if (has-granfather? person 'person)
          (let ((birthplace (first (my-slot-values person 'has-birth-
place)))
             (out nil))
        (if birthplace
               (let ((neighbours (setofall '?x `(and (person ?x) (has-
birth-place birthplace)))))
              (if neighbours
                  (setf out neighbours))))
        out)))
```

6.6.6 Alternative pathways' strategies

In conclusion of this extensive description of the learning pathways available in PhiloSurfical, it is useful to recapitulate what are the main differences between our approach and the other possible ways to generate pathways. We are making reference here to the literature review chapter, in particular sections 2.2.5 and 2.4.1, in which we highlighted the main ideas behind digital narrative systems and, more generally, the key features of pathway-oriented applications.

Essentially, it is fair to say that our pathways are *graphs*; in particular, since they have been accurately formalized using our ontological categories, these graphs can be successfully used on a knowledge base. The result of this application is a subset of the knowledge base in which a number of entities and relations are organized and presented according to a specific coherence principle (e.g., theoretical, geographical, etc.).

There can be at least two other approaches to the creation of pathways, so we will discuss them in turns.

1) First, an approach focusing on more user-centered and interactive mechanisms for the generation of pathways. This is the approach so often found in role-playing computer games (Ryan, 2001), or in educational applications such as the aforementioned Walden Paths (Dave et al., 2003,Shipman III et al., 1998) or Scene Driver (Wolff et al., 2004). In such cases pathways are usually predefined by the programmer or the educator, but users interact with them in a more gradual way, that is,

incrementally, one step at a time. The main advantage of these systems is that they are likely to be more entertaining, for they give users a stronger feeling of being 'immersed' in a different world - a sandbox where each of their choices is immediately evaluated and can lead to different results. This feature makes these systems particularly suitable for realizing a truly constructivist learning environment (as discussed in sections 2.2.1 and following). The main downside of this approach is instead the fact that, since users are relatively free in the choices they make, it is not guaranteed that the navigation or pathways they create are always the ones envisioned by the designer. Nonetheless, we looked with interest at this approach and intend to explore further how effectively it could be translated to a philosophical domain.

2) Second, an approach based on the formalization of pedagogical strategies. This type of pathways are mentioned by Crampes and Ranwez in the context of the 'ontology-driven' approaches to navigation (Crampes and Ranwez, 2000), for they are constructed by relying on some pedagogical knowledge, which is formalized in a separate ontology. For example, this extra ontology can be describing learning *styles* or learning *designs* (Chen and Mizoguchi, 2004) (Mizoguchi and Bourdeau, 2007). In practical terms, according to this approach the mechanisms creating the pathways should keep into account pedagogical factors such as the knowledge level of users, their preceding navigations, their individual learning styles, etc. Thus, when interacting with the system users are presented resources both according to their description in a domain ontology and to the role they can play within a wider pedagogical strategy.

In general, we reckon that this approach too should be further investigated in the context of the philosophical domain. It is worth noting though that the aforementioned lack of agreement concerning what pedagogical approach is best to be used when teaching philosophy (cf. section 2.2.2) may cause some difficulties during the formalization of the relevant pedagogical knowledge.

6.7 Summary

In this chapter we have presented PhiloSurfical, an ontology-based web application which aims at facilitating learners' encounter with the philosophical domain. The application makes use of the ontological representations presented in chapter 5, with the purpose of testing their employability in a real-world scenario.

In particular, the application has been instantiated with a specific philosophical text, Wittgenstein's Tractatus Logico-Philosophicus. Accordingly, in the previous sections we discussed various topics connected to the creation of a domain-specific knowledge base. Also, we have described the technical components composing PhiloSurfical and provided a walk-through of the application interaction design.

Finally, we discussed extensively the main idea underlying PhiloSurfical's approach. That is, the provision of various *navigation mechanisms* by which

users can *explore* the world of philosophy. We explained why we call such mechanisms *learning narratives,* and how they are connected to other theoretical and technical approaches we presented during the literature review (especially, the approaches discussed in section 2.2.5 and 2.4).

The following two chapters (7 and 8) deal with the evaluation of our work. In particular, chapter 8 focuses on an user-evaluation experiment of PhiloSurfical. For this reason, here we will not discuss the possible future research directions regarding our application; instead, such discussion can be found at the end of chapter 8 (section 8.5).