A Semantic Search Engine for a Multimedia e-Learning Tool

Serge Linckels, Christoph Meinel

Department for Theoretical Computer Science and New Applications,
University of Trier, Germany
{linckels, meinel}@TI.uni-trier.de
http://www.Linckels.lu
http://www.informatik.uni-trier.de/~meinel

1 Introduction

At the beginning of the e-learning vogue in the mid 80s, it was said that computer tools, CD-ROMs or online-courses broadcast *via* a network, would replace teachers. Today, practical, pedagogical and financial matters overwhelmed most of these visions. No teacher, no university professor is or will, at least in a near future, be replaced by some cybernetic being or by a simple on-screen show from a CD or the internet. However, the long research on the field of e-learning brought a lot of improvements in education. Although we cannot imagine school classes without a teacher, there are a lot of new methods and tools that teachers and students can use as a complement. Here are two examples:

- No displacement is necessary and no contact with the teacher is needed. Courses are broadcast live or on demand over the Internet. Thus, students can review a missed lesson or an important topic before a test. Also think about all the great contribution this made to distant learning. Anybody can improve his knowledge by subscribing to an "online school" without having to leave his home.
- The teacher has the possibility to promote autonomous learning by using "smart tools". Students play the role of an explorer who discovers new information. They can even create their own course content by assembling the pieces of information they selected. The multimedia aspect and the attractive interfaces draw the student's attention. In general, motivated students are good students, and good students have better results in school.

However, many e-learning tools and solutions are the results of theoretical and scientific research rather than of practical, concrete and founded needs in education. Consider the following two examples:

- A multimedia encyclopedia is a great tool for teachers to find information for preparing their courses. But students can only use few immediately. The information is often presented in too complicated a language and there is simply too much information on one topic. There are no filter techniques to adapt the content to the skills of the user (for example: less information for a kid, exhaustive information for a teacher). Often, search mechanisms are based on simple keyword searches that are not effective, for example: the user gets a large number of possible results.
- A lot of universities offer courses online, often streamed from a server. We will not discuss any performance or financial constraints; each lesson normally takes over an hour that has to be transmitted. Well, let us suppose that the student has a dozen of such lessons to consider for a test. Even if he is searching for precise information, it is difficult and very time consuming to scan through all the possible streams to find the appropriate part.

In our contribution we tried to start from the needs of teachers and students. Founded on the experiences, critiques and wishes of teachers and professors, we formulated the needs listed in table 1.

Table 1. Enumeration of the needs for an e-learning tool of practical use.

For the student:	• Information appears in an attractive form • Answers are short, precise and easy to understand • Large potential of knowledge • Simple and human interaction • Search of information is easy, but effective
For the teacher:	• Restricted and secure information area • Guaranteed quality of information • Easy to administer and to extend (add/remove information) • Easy to access without a lot of specially required equipment and configuration • Motivating for students

In proceedings of the Association for Educational Communications and Technology / the University of North Texas (AECT/UNT) Summer 04 Conference, Denton, Texas, USA, 17 – 19 June 2004, pages 224 – 231.

2 The multimedia jukebox approach

Our answer to the above listed needs focuses on three key features: the representation of the information in a multimedia form, the splitting of the content into small *clips* and a semantic search mechanism for information retrieval. We firstly published these ideas in [21] and [22].

2.1 The multimedia interface

Today, kids are spoiled with all the wonderful and attractive interfaces of operating systems, applications and games. New software without a graphical user interface in vogue is banned to failure. That's exactly why students prefer websites with colors, images, sound and animations, rather than for example books as learning syllabuses. In fact, isn't it clearer reading something that is illustrated with images, pictures or drawings? Every person is different in his sense of perception. Some understand better if they hear the explanation by the means of verbal communication, some need to write it down, others must see it in the form of a text or a picture and others again have to touch it. A good teaching tool must present the same information in different forms in order to activate as many senses as possible. The psychological foundations were proven by the work of [10] and [11]; *information that is presented at the same time in different forms improves the understanding of the information*.

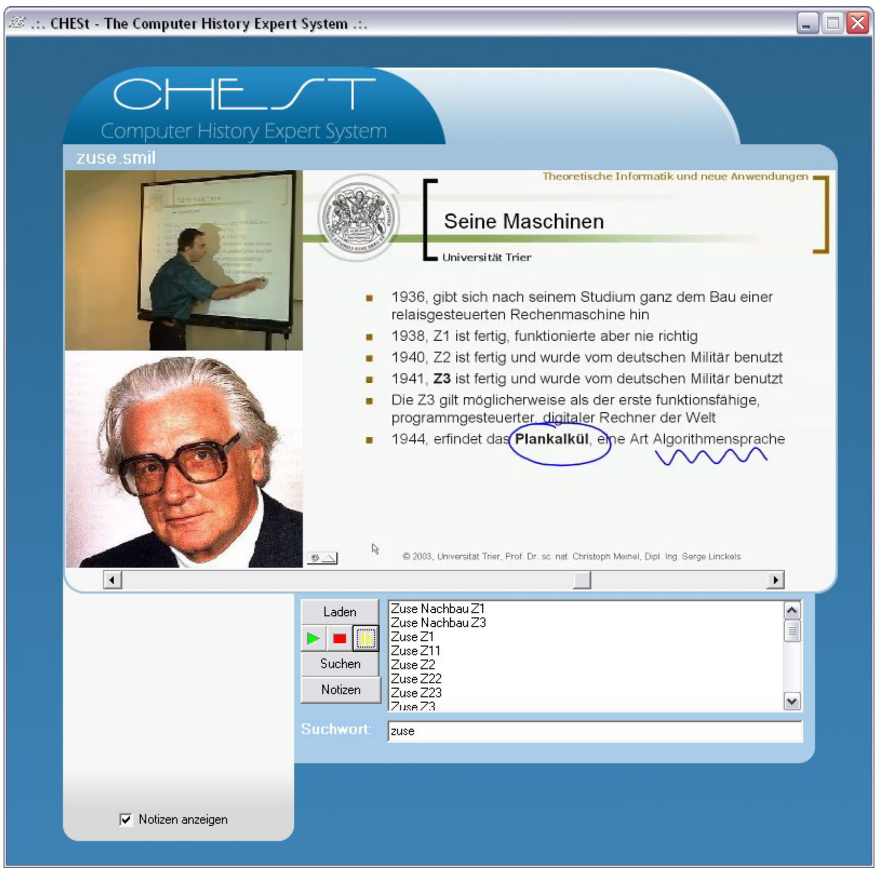

Fig. 1. Screenshot of the prototype *CHESt* with a keyword search on "zuse". The window shows a list of search results in the bottom right-hand corner. Selecting a topic from this list will play the clip, like the one shown in this example, where the teacher uses an interactive board. Added handwritten comments made by the teacher are integrated and applied in real time on the text (top right-hand window).

The interface of our tool is basically organized in three windows (see figure 1). The **first window** (video and audio) shows a teacher explaining something on the whiteboard. This is the student's common view in a classroom and should create a kind of "virtual classroom" atmosphere. Based on practical teaching experience we can confirm that students often take lessons where they use a new computer tool or do research on the web for example, as a kind of game, without relation to the "normal" lessons. The video sequence should keep them concentrated on what they do and draw their attention to what the teacher is explaining.

The **second window** represents the usual blackboard. It is in fact a zoom on the whiteboard that the teacher uses in the video (first window). Although the blackboard is the most used medium in schools, it has a lot of disadvantages, for example:

- It is impossible to represent pictures.
- It is difficult and time-consuming for the teacher to create a complex drawing.
- It is time-consuming for students to reproduce its content in their books.
- The content is not available for later lessons and must be reproduced.

The virtual blackboard in our tool has the following features:

- The teacher can use this area for an on-screen presentation (for example: *PowerPoint)*.
- He can add handwritten information to the *smartboard*, which is reproduced in this window both simultaneously and in exactly the same way.
- He can also display the desktop of his connected laptop, for example in order to explain a certain application, to show a website or to demonstrate the settings of the computer.

The **third window** can be used for any purpose. It can contain links to a photo gallery, hyperlinks to additional information on the web, book references or just a single picture of the subject the teacher is speaking about.

We used *Tele-TASK* [3] [9] [12] [13] to record the lessons in order to create one well-structured multimedia stream. The result is a *RealMedia* file that can be played with any compatible software, for example the free *RealOne Player* [6].

2.2 The clip approach

Essential in our concept is the length of the stored items in the knowledge base; the duration of the video sequences. The younger the user, the shorter the time during which he/she will concentrate on the information displayed on the screen. Furthermore, we mentioned already in the introduction that it is not easy to find the appropriate information inside a large piece of data, for example an online lesson that lasts 90 minutes. Jules Cesar already said: "Divide to reign". Therefore, we divided all our multimedia data into small *clips*. The duration of each clip varies from several seconds to 3 or 4 minutes. Each clip documents one subject or a part of a subject. Together, all the clips of the knowledge base cover one large topic. In our prototype CHESt (*Computer History Expert System*), we focused on one precise topic: computer history. We produced 300 clips about every important event in computer history. CHESt exists as standalone application (we managed to store the whole knowledge base with the application software on one single CD-ROM) and as online application that can be found at [7]. The later uses a streaming server to transmit the clips to the user's browser.

Splitting a large topic like computer history into a lot of small pieces is much easier than we assumed at the beginning. We are now convinced that most courses taught in schools or at universities can be divided into smaller *atomic units* where each covers one precise subject. Teachers of different fields confirmed that this concept is not limited to computer-science and that it could be used in their field too. For instance, in language courses, a teacher could record one clip per grammatical rule. Another concrete test was made in the field of biology where a teacher used our tool to explain the basic function of the heart. Further details would be explained in additional clips.

One more advantage of that clip-approach is the simplicity of administration. If the tool does not cover a certain topic, a new clip can be recorded and added to the knowledge base. The intervention of a computer-science expert is not necessary.

2.3 Finding the right clip

Having a large knowledge base with short multimedia clips is one thing; another thing is to find the right clip. The more clips you have, the better your knowledge base covers a certain topic, but the more difficult it is to find an appropriated clip. A first solution is of course to let the user browse through a table of content where all the clips are listed in categories, for example: *hardware/storing devices* or *people/still living*, etc. and load the chosen clip. This possibility is offered in the standalone version of our tool, not in the online version for the moment. The main disadvantage is that there is no additional information about the content of the clip except for a short designation. Furthermore, this operation is time-consuming and not very effective, because the user has to search and maybe test different clips before he finds the answer. An automated search would be better. At the moment, the prototype CHESt has only a keyword search. If the user enters "arpa", the system will list all clips about the ARPA and the ARPANET. The user then selects a clip from that list to be played. The main disadvantage is that the user must already give a part of the answer. For example, you want to know who invented the first computer. Then you should enter keywords like "Zuse" or "Aiken". You cannot ask: "Who invented the computer?" Another problem is that, depending on the keyword, you will get a long list of possible results. Finally, even if a clip is about a certain topic, it must not necessarily be found from the keyword the user has entered, for example the user enters "disk" but the matching keyword would be "floppy".

The most efficient search mechanism is to allow the user to enter a complete question. The tool should "understand" that question and give a small list of pertinent clips as answer, or better even just one clip. Technical details about a *semantic search engine* are described in section 3. This solution is also pedagogically welcomed because in schools, students are forced to express themselves in complete sentences and not just with keywords. Most important is the fact that the interaction between the student and the tool takes place in a very human and simple way. An imaginable improvement would be a verbal communication where the user could speak his question into a microphone.

2.4 Example of an all day application of CHESt

With the features described bellow, we could imagine that the student who is working with CHESt has his own *virtual teacher*. This teacher's answers are short and presented in an interesting multimedia form. The student can communicate with him in a very simple and human way by typing his question, or in a later improved version by means of verbal communication. As every tool and method, CHESt will not replace every conventional lesson. We see it as a complement useful for certain occasions. It's up to the "real" teacher to decide, for which of his lessons it is appropriated, for example:

- To introduce a new subject by letting the students discover themselves some new information.
- To use CHESt as complement to find illustrations for a certain topic (for examples pictures of old computers or computer pioneers).

The students could work in groups or alone. In fact, they create their own course content: the clips they consult. Depending on the kind of work, they can print a certain scene of a clip, copy snapshots into a text document or simply take notes. The teacher is sure that the information they get is correct and secure. Here a concrete scenario: "*Hi students. Today we are working on computer history. Here is a list of interesting questions. You have 40 minutes to search for information before we discuss your answers together. Of course, use CHESt!*".

- *Who invented the computer? When?*
- *What is the Colossus?*
- *What is a transistor useful for?*
- *Explain the word FTP.*
- *Who sent the first e-Mail?*
- *What was the size of the first hard disk?*
- *Who invented Unix?*
- …

3 Describing the meaning of the clips

In section 2, we described our prototype CHESt from a pedagogical view. The search of a certain clip, not by keywords, but by a freely formulated question is one of the main necessary improvements. Though, before the tool can even try to understand the user's question, it has to "know" what data are stored in the knowledge base. In other words, every clip must be described in a machine-readable form. Therefore, we have to add data to each clip to describe its meaning. That kind of data is called metadata. For this purpose we use the *Resource Description Framework* (RDF), introduced by the W3C in 1998 to build the Semantic Web [16]. In principle, this is done once, at the moment when the clip is added to the knowledge base. However, the computer can assume a part of this task. The different steps are described below.

3.1 The CHESt RDF vocabulary

With our concept to use short clips, we have the great advantage that we can describe the meaning of one clip with few metadata. We divided the CHESt knowledge base logically into two classes: clips that describe inventions (things) and clips that describe inventors (persons). Assertion: an invention was invented by one or more inventors. An invention and an inventor can be a *resource* (in our case: a clip) or a value (just a textual information). Every resource is described with properties. An **inventor** has three properties (*predicates*): his name (`vCard:FN`), the year of his birth (`chest:year_birth`) and the year of his death (`chest:year_death`); if still alive, this property is left blank. As you see, we used the W3C recommendation v*Card* namespace property *full name* (FN) [15]. The class **invention** is divided into a number of subclasses to better organize the different resources (see figure 2). We used the *Dublin Core* (dc) namespace [4] to describe an invention with the following properties (*predicates*): its description (`dc:title`), its date of first appearance (`dc:date`) and its creator (`dc:creator`). The complete CHESt RDF schema can be found at [8]. With these few elements we can semantically describe every clip.

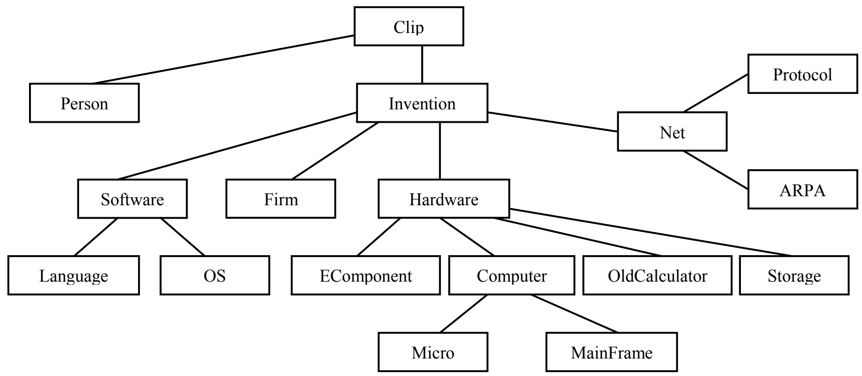

Fig. 2. Class hierarchy of the CHESt RDF classes

3.2 Generating the CHESt dictionary

The next step is to search inside every clip for metadata. For example, the clip, which describes the calculator "ENIAC" should be scanned to find its description, the year where it was first taken into service, and the name(s) of its creator(s). We tried to apply an approved approach in the field of computer linguistics: create a dictionary of synonyms for every CHESt RDF element [5] [14]; in one column one will find the RDF elements and in the other column there is a list of natural language synonyms. For example, if we are scanning for `dc:creator`, we are searching for words like *creator*, *builder*, *constructor*, *inventor*, etc. For our prototype, we decided to consider only the textual data from the PowerPoint presentations and to ignore the teacher's audio information and his handwritten notes for example. With a special tool [17] we are able to convert the PowerPoint documents into pure text files. Then the *stemming process* can begin. All non-words (words that contain digits or special characters) and words with just one letter were eliminated from the generated text files because they have no semantic influence. All words are converted into lowercase and special characters are replaced by a space. Finally a list of 20640 remaining words was created from the whole 300 clips in the knowledge base. All were represented in a tree, where every node represents one letter. The tree is built in less than a second. The words are read vertically from the top (root) down along the branches (see figure 3). This technique also allows to eliminate all double words. Each node contains the number of words that end with that particular letter. There are 4215 remaining unique words with an average length of 8.049 letters per word.

The dictionary of synonyms is built from that tree. The idea is to regroup words with similar spelling and thus with the same meaning (for example: build, built, builds). It is impossible to detect automatically all synonyms, because there are words that have a similar spelling, but not the same meaning (for example: consult, consume). The aim of the stemming process is to limit human intervention by proposing clusters of generated synonyms. Further details of this process are described in [22].

Why didn't we use an existing dictionary of synonyms, for example GermaNet [20]? For two reasons: first, by choosing an existing dictionary, CHESt would immediately be set to a certain language (English, German, French…). Our solution is language independent, because it builds its dictionary from an existing content. Second, even if we still have 4215 unique words to scan for synonyms and RDF elements, it is still much less than a complete dictionary with at least 200 times more words. Note also, that the words listed in our dictionary are words that are used at least one time.

3.3 Generating the RDF description

The final step consists in scanning through the clips (as text files) and searching for synonyms for the RDF elements described in section 3.1. In our case 273 out of the 300 clips were described automatically and without human interaction. In some clips, different concurrent synonyms were found. The most frequent example is the RDF synonym for `dc:date` which represents the date of first public appearance of an invention. For different inventions, there was a date of planning, a date of starting the construction and a date of launch. To solve this ambiguous problem, we programmed our tool so that, in case of concurrence, it chooses the second occurrence and protocols the problem in a log file. The final result is an RDF/XML serialization for each clip (see figure 4). We used Jena [18] to generate the RDF serialization. Jena allows to store de triples in a simple XML-file but it also supports several RDMS (for example MySQL or PostgreSQL).

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:chest="http://www.linckels.lu/chest/elements/1.0/">
  <chest:Person rdf:about="http://sigma957.lte.lu:8080/ramgen
                                     /Archive/Zuse.rm">
     <vCard:FN>Konrad Zuse</vCard:FN>
     <chest:year_birth>1910</chest:year_birth>
     <chest:year_death>1995</chest:year_death>
  </chest:Person>
</rdf:RDF>
```

Fig. 4. Example of a semantic description of a clip using RDF/XML and streaming access to the multimedia files. The clip is about the person "Konrad Zuse".

4 Understanding the user

The number of results (in CHESt a matching result is a clip) will be shorter and more pertinent with a semantic search than with a normal keyword search. Furthermore, the user must not enter a part of the answer in its question, for example: “Who invented the first computer” doesn’t contain the name of the inventor. In fact, the name of the inventor is the information to find.

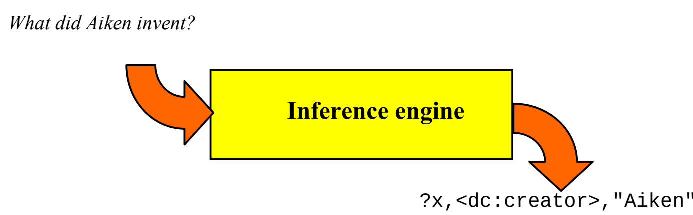

Fig. 5. Principle of the inference engine, that transforms a non-formula question into a well-formulated RDF query.

We now dispose of a well-formulated and semantically described knowledge base in RDF (see section 3). To perform a semantic search, the question entered by the user must be transformed into RDF too, in order to have the same structure for the question and for the database. The backbone of our semantic search is an inference engine which transforms a normal sentence (the user’s question) into a well-formulated RDF query. We used RDQL to access our RDF knowledge base [23]. See [1] [2] [19] for more details about semantic databases. For example: “What did Aiken invent?” should become:

```
select ?x WHERE (?x,<dc:creator>;"Aiken").
```

As described in section 3.2, all the words in the dictionary are basically regrouped in two categories: words that are of any semantic use (which are associated with an RDF element) and words without semantic use (which are not associated with an RDF element). It is clear that this dictionary can only be used in a precise context, which is computer history in our case. The user's question is also put in that same context for parsing, for example if the users asks "Who invented the penicillin?" the tool cannot give an answer because the question is outside the tool's context. Starting with these constraints, the transformation of a common formulated sentence into RDF can be resumed by saying that the system has to replace all semantically important words by the RDF corresponding elements and to throw unimportant words away. Of course, the shorter the questions, the better the results.

Since all RDF elements in the CHESt schema are defined either as {subject, object} or as {predicate} (see section 3.1), there is no doubt about the membership of the recognized RDF elements. Except `chest:Person` and `chest:Invention` (or one of its subclasses), all RDF elements are predicates (see figure 2). As we are dealing with questions, there should always be a missing part, normally the subject or the object. Remember the basic assertion: "An invention was invented by an inventor". Generally, members of the class `chest:Person` are objects, members of the class `chest:Invention` are subjects.

4 Conclusion and Outlook

Our primary aim is to create a tool or even a new method of teaching. The teacher is in the background and the student plays the role of an explorer. Therefore, it motivates the student because (s)he can create his own course content. The information is presented in an interesting multimedia form. The system ‘understands’ the questions of the user and gives efficient answers: there are no long searches for answers, but the requested answers are rendered in a concise form. Of course, a motivated student is a good student and good students normally achieve better results. Thus, this tool is supposed to improve education.

The prototype CHESt covers the field of computer history, but by generalizing the knowledge base, it can be used in nearly every course in any school, college or university. Its advantages are that it promotes independent learning. By adding other clips from other fields (for example: biology, electronics, etc.), CHESt could become more than just an expert system on computer history. Ideas are to use external and existing resources of information, rather than to record new clips for each subject. Another idea is to test how a RDF vocabulary can be associated automatically with an existing dictionary.

The prototype CHESt is tested with a simple keyword search in some selected schools in the summer term of the year 2004. We hope that the collected data will give us a first impression of the reaction of teachers and students with that new tool.

References

1. Agrawal R., Somani A., Xu Y.: Storage and Querying of E-Commerce Data. In proceedings of the 27th VLDB Conferences, Roma, Italy (2001)
2. Broekstra J., Fluit C., van Harmelen F.: On-To-Knowledge. The State of the Art on Representation and Query Languages for Semistructured Data. Administrator Nederland b.v. (2000)
3. Chen T., Ma M., Meinel C., Schillings V.: Tele-TASK, Teleteaching Anywhere Solution Kit. Universität Trier. `http://www.tele-task.de/`
4. Dublin Core Metadata Initiative (DCMI). `http://dublincore.org`
5. Manning Ch., Schütze H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge Londong (2003)
6. Real.com: RealOne Player, `http://www.real.com/` (2004)
7. Linckels S., Meinel Ch: Computer History Expert System (CHESt), `https://sigma957.lte.lu/chest` (2004)
8. Linckels S.: CHESt namespace, `http://www.linckels.lu/chest/elements/1.0/` (2004)
9. Ma M.,Schillings V.,Chen T., Meinel Ch.: T-Cube, A Multimedia Authoring System for eLearning. In proceedings of the AACE E-Learn 2003, Phoenix, USA (2004) 2289-2296
10. Mayer R., Gallini J.: When Is an Illustration Worth Ten Thousand Words? In Journal of Educational Psychology, Vol. 86, No. 3, American Psychological Association Inc. (1994)
11. Mayer R., Sims V.: For Whom Is a Picture Worth a Thousand Words? Extensions of a Dual-Coding Theory of Multimedia Learning. In Journal of Educational Psychology, Vol 86, No. 3. American Psychological Association Inc. (1994)
12. Meinel Ch., Schillings V.,Walser V.: Overcoming Technical Frustrations in Distance Education: tele-TASK. In proceedings of e-Society 2003, Lisboa, Portugal (2003) 34-41
13. Meinel Ch., Schillings V.: tele-TASK - Teleteaching Anywhere Solution Kit. In proceedings of ACM SIGUCCS 2002, Providence, USA (2002) 130-133
14. Carstensen K.-U. et al.: Computerlinguistik und Sprachentechnologie. Spektrum Lehrbuch (2001)
15. World Wide Web Consortium: Representing vCard Objects in RDF/XML, `http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222/` (2004)
16. World Wide Web Consortium: Resource Description Framework (RDF) / W3C Semantic Web, `http://www.w3.org/RDF/`
17. Linckels S.: PowertPoint to Text Converter (PPT2TXT.EXE), `http://www.linckels.lu/logiciels/ppt2txt.zip` (2003)
18. HP Labs Semantic Web Research: Jena, `http://www.hpl.hp.com/semweb/` (2004)
19. Patel K., Gupta G.: Semantic Processing of the Semantic Web. In proceedings of the Second International Semantic Web Conference, Sanibel Island, USA (2003)
20. University of Tuebingen: GermaNet Homepage, `http://www.sfs.nphil.uni-tuebingen.de/lsd/` (2004)
21. Linckels S., Meinel Ch.: An Application of Semantics for an Educational Tool. In proceedings of IADIS International Conference of Applied Computing 2004, Lisbon, Portugal (2004), pages II 234-II 239
22. Linckels S., Meinel Ch.: Automatic Interpretation of Natural Language for a Multimedia e-Learning Tool. In proceedings of International Conference on Web Engineering, Munich, Germany (2004)
23. Miller L., Seaborne A., Reggiori A.: Three Implementations of SquishQL, a Simple RDF Query Language. In Proceedings of International Semantic Web Conference, Sardinia, Italy (2002)