AN APPLICATION OF SEMANTICS FOR AN EDUCATIONAL TOOL

Serge Linckels

University of Trier, Department for Theoretical Computer Science and New Applications
Universitätsring 15, D-54296 Trier, Germany
linckels@TI.uni-trier.de

Christoph Meinel

University of Trier, Department for Theoretical Computer Science and New Applications
Universitätsring 15, D-54296 Trier, Germany
meinel@TI.uni-trier.de

ABSTRACT

In this paper, we present a new educational tool. It can be used in nearly any course. It offers the teacher a new method of teaching, where the student plays the role of an explorer who searches and discovers new information about a certain topic. This information is presented in a multimedia form and taken from a secure and certified knowledge base. Furthermore, we present a project to improve this tool by allowing the student both to search for keywords and to enter a freely formulated question. This paper is about an inference engine that transforms a human sentence into an RDF (*Resource Description Framework*) statement.

KEYWORDS

Semantics, education, multimedia, web, RDF, knowledge base, search engine.

1. INTRODUCTION

Teachers are faced with the task of transferring and explaining knowledge and information to students; to achieve this result, different methods of teaching like frontal teaching, question-and-answer methods, group work and many more are implemented on a regular basis. There is no unique or universal method, but the most appropriate method has to be chosen for a specific topic. For some subjects, the frontal method is the best, for others group work would be a good alternative. The good teacher has to decide which method is best for his topic and lesson. In addition to the method, the teacher also has to choose the media he will use to communicate the information. These are three popular examples:

- The *blackboard* is the most widely used medium. One big disadvantage is that the teacher has to produce everything (text, drawings, graphs, etc.), a job that is very time-consuming. Furthermore, it is not suitable for reproducing images or complex presentations; this proves to be awkward both for the teacher and for the students who have to copy the information from the blackboard into their books.
- *Books* are often perceived by students as boring, especially if there are a lot of pages without illustrations. The bigger the volume, the greater the work to find the desired information and the smaller the student's motivation to use the book.
- The *web* is a motivating playground for students, but covers a lot of pedagogical pitfalls. There are a lot of sites with a content that is not appropriate for education. Furthermore, the quality and correctness of the information can hardly be proven. Finally, the increasing problem of the web is that there is too much information. These facts complicate the search for information. A student has one question in mind and enters some keywords in a search engine, but the result is often frustrating. He expects one precise answer to his question, but the search engine returns dozens or even hundreds of possible web pages.

Both the method and the media that are used have a direct influence on the quality of the education and the motivation of the students. A 'perfect tool' with an appropriate method would help the teacher not only to motivate students, but also to communicate more information of high quality in less time.

2. THE PROTOTYPE CHEST

In this paragraph we describe the steps we made in order to realize an educational tool that is based on the requirements and experiences of teachers that we interviewed . We focused on one precise topic, in this case computer history, to develop a prototype for further research. The prototype of the *Computer History Expert System (CHESt)* can be found at: `http://sigma957.lte.lu/chest`. The knowledge base contains a great number of multimedia clips that can be accessed with specific client software or with a browser via the web.
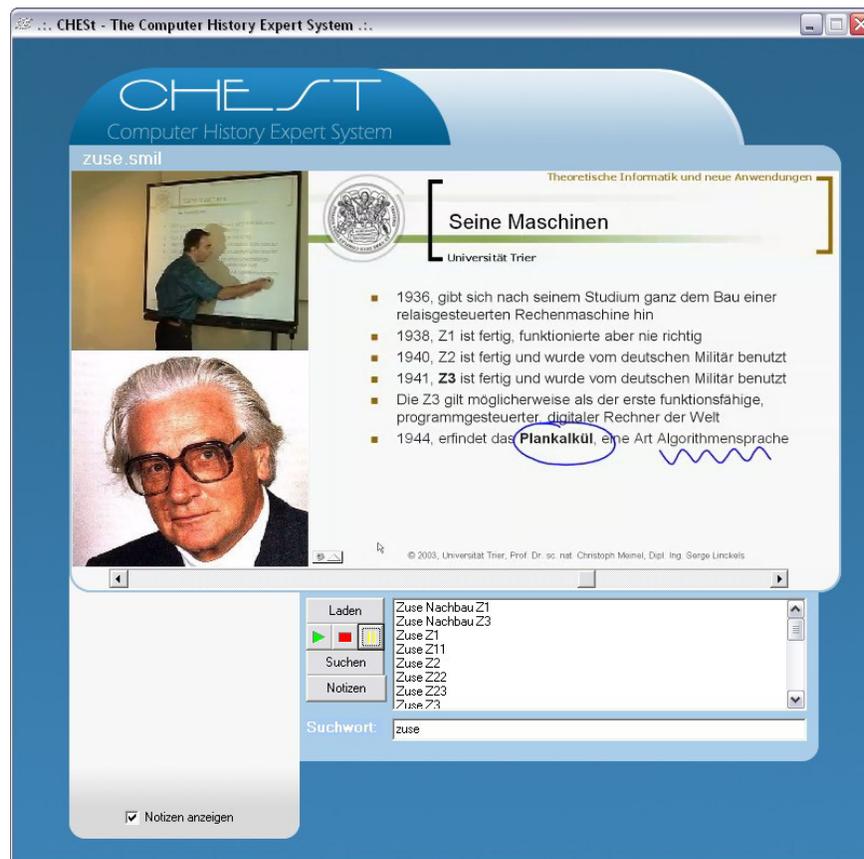

Fig. 1. Screenshot of the prototype CHESt with a keyword search on "zuse". The window shows a list of search results in the bottom right-hand corner. Selecting a topic from this list will play the clip, like the one shown in this example, where the teacher uses an interactive board. Added handwritten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

2.1 The computer history jukebox

One of the most essential requirements for this new tool was a secure and restricted knowledge base with a guarantee for quality content. We were aware that we had to limit our tool to a specific topic in order to cover that particular subject completely with multimedia clips. We decided to focus on the field of computer

history. This is quite an appropriate topic for our research because it is not too complex, of general interest, popular in computer science courses and easy to represent in a multimedia way.

Several universities offer online lectures; they broadcast live on the internet or on demand. This offers a further service to students who can watch/listen to the same lecture several times and look up specific information among the material of the lecture. Unfortunately, a lecture often lasts more than an hour, which makes searching for precise information difficult if the user does not know exactly where the information is located. Furthermore, it often happens that, before a test, a student has particular questions and looks for short and precise answers. Consulting long recorded lectures is often too time-consuming. So our idea was to create a knowledge base with very short clips only. In order to guarantee a large potential of information inside that knowledge base, we had to record a lot of clips to cover the complete field. We produced about 300 clips about every important topic of the history of computer science. The duration of each clip varies from several seconds to 3 or 4 minutes. We documented every important fact from the *Abacus*, the *Pascaline* or *Babbage* to the first digital calculators by *Zuse* and *Aiken*. CHESt also contains information about the beginning of the engineering period of the *big computers* from the 50s to the 70s and the era of the *home computers* and *micro-computers* of the 80s and 90s. Besides the development of computers, CHESt also traces the evolution of *operating systems*, *programming languages* and the *ARPANET/Internet*. Most of the computer pioneers' lives, inventions, contributions and anecdotes are covered by CHESt.

2.2 The encoding of the clips

We used *Tele-TASK* [3] [9] [12] [13] to record the clips in order to create one well-structured multimedia stream. Tele-TASK files can be loaded into any program that is capable of playing a *RealMedia* file. We tried to keep CHESt small. Fortunately, Tele-TASK offers different compression options to achieve the desired size and quality of the clips. We managed to store the whole knowledge base (with the application software) on one single CD-ROM.

2.3   The interaction with CHESt

Tele-TASK is a perfect system for our tool according to the work of [10] and [11]; *information that is presented at the same time in different forms improves the understanding of the information*. CHESt is basically composed of 3 windows, each presenting the information requested in a specific way (figure 1). The first window (video and audio) shows a familiar situation: a teacher explaining something on the whiteboard. We think that maintaining this common teacher-student view is very important for the student, who is used to seeing a person explaining something. The second window shows detailed information such as text, pictures, live handwritten notes from the teacher, etc. This in fact represents the classical blackboard in a classroom with all its advantages (for the student and the teacher) and much more, for example:

- The teacher can use this area for a PowerPoint presentation.
- He can also display the desktop of his connected laptop, to explain a certain application in real time.
- He has direct access to the Internet (without having to configure his laptop to the LAN).
- He can add handwritten information to the *smartboard* that is simultaneously and in exactly the same way reproduced in this window.

Both windows are filled by the Tele-TASK stream (`.rm`). The third window is filled by a hypertext file. It can be used for any purpose, for instance a link to a photo gallery, hyperlinks to resources on the web, book references for further reading or maybe just a single picture of the subject the teacher is speaking about.

Another requirement was that the tool must be simple and ready for use immediately and everywhere. In this way, neither students nor teachers have to run long, tedious and resource-consuming installation and configuration procedures. Therefore, we decided not to use a large database management system. Instead, we opted for a very simple solution: all multimedia clips are stored as simple files. Secondly, CHESt exists in two software solutions: an application (Windows) version and a browser version. The application version was developed in *Borland Delphi*, using the *Real Active-X* plug-in. A platform independent solution in Java is about to be produced. The application can be started on the CD without installation and can also be copied to the hard disk. The web version embeds the free *RealOne Player* into a HTML page using the `<object>` tag. Independently from the software solution, the different multimedia clips can be retrieved as a whole file

or as stream. The later requires that the knowledge base is located on a streaming server. The user has two possibilities to select a clip for playing:

- He can browse through the directory structure and simply select a clip to load. The logic of the file structure helps the user to quickly find an appropriate group of clips.
- The most useful solution is of course a search mechanism. At the moment, our prototype only works with a keyword search engine. The search for information is done by scanning for keywords in the filenames. The whole CD is scanned in about 5 seconds (depending on the speed of the CD-ROM drive).

Another feature of CHESt is that the user can navigate through the clip and add personal notes. These notes are stored locally. The next time that the student watches the same clip, his notes are displayed in the same position. This allows the user to personalize the knowledge base to a limited extent.

3. THE SEMANTIC SEARCH ENGINE

There is often a certain anxiety to address the teacher, especially as far as younger students are concerned. This is partly due to the fact that the teacher plays the role of a respectful person to whom the student has to speak in a polite way. Furthermore, students often express themselves badly, which proves to be unfortunate in this case. Sometimes, the teacher can hardly understand what he or she wants. For the student, it seems that 'there is nobody who understands me'. A computer-based tool that allows the student to enter a question in his own words and not by selecting predefined questions or by answering different questions from the computer, would be precious and helpful for the school community. For example, the user enters the question: 'Who invented the computer?' The system gives the correct answer or a small list of possible answers. By *answer*, we understand a multimedia clip as described in paragraph 2. We describe in this paragraph how we try to contribute to the achievement of such a search mechanism.

3.1 CHESt seen as ontology

A lot of research work has been done in these last years to improve search engines on the web. Most of them start with the idea of the semantic web [7]. One of the biggest problems is that every web site needs to be described with enough metadata, so that search engines and agents can fully understand the meaning of its content. In practice and when applying this idea to a major part of the web, this is not very realistic, at least not at the moment. There is another problem as Tim Berners-Lee mentioned: '[…] *transferring content between web applications is still surprisingly difficult*' [6]. There are still too many different and incompatible formats, protocols and applications, so that the semantic web remains a theoretical draft… and a dream.

We try to start our project from another point of view. With CHESt, we can explore our ideas in the context of a closed and reduced knowledge base. Therefore, we have the possibilities to apply one common structure, add metadata to any item (and not just a part of it) and test different search methods. Maybe in future research, we can generalize a part of our work on the web.

Our basic idea is to use RDF and RDF Schema [15] to directly describe that ontology [5]. First of all, we describe the content of the knowledge base with enough well-structured metadata. Secondly, the freely entered question of the user, which is in a non-formula state, has to be converted into a well-formulated statement that has the same structure as the metadata of the knowledge base. When this is done, both the knowledge base and the user's question have the same structure and can thus be compared. Both steps are described below.

3.2 Describing the knowledge base with RDF

The first step is to add enough metadata to the knowledge base to describe the meaning of every clip. We are working on a mechanism to extract semantics automatically from a given document (for example: PowerPoint-slides that are used by the speaker) and insert them into a database. This tool converts the slides to raw text and submits every sentence to the inference engine (described later in this paper). All sentences are, if possible, automatically transformed into a well-formulated RDF statement (*triple*). In order to be as

precise as possible in the semantic description of the clips, we created our own RDF schema. The class *Clip* is a subclass of the RDF class *Resource*. All other classes are subclasses of the class Clip. The complete CHESt schema is available at [8]. We also re-use the Dublin Core properties `creator` and `title` [4].

3.3 The inference engine

A basic idea of *intelligent* search engines is to filter the results and to extract the unsuitable ones, and then to perform a keyword search on the remaining ones. Unfortunately, keywords do not necessarily lead to all semantically matching results. For example, the user asks the question: 'Who built the very first calculator?'. At first sight, only the word *calculator* could be a good keyword. But in a knowledge base about computer history, there will of course be a lot of content that will match that keyword, even if it has little to do with the inventor of the computer.

Who built the very first calculator?

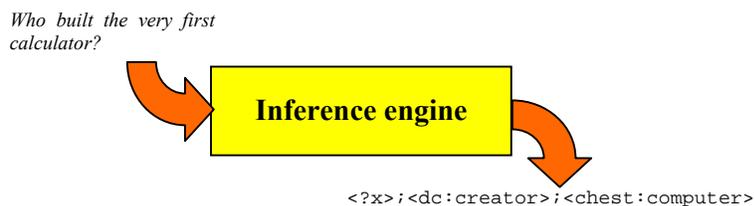**Inference engine**

`<?x>;<dc:creator>;<chest:computer>`

Fig. 2. Principle of the inference engine, that transforms a non-formula question into a well-formulated RDF query.

So, we need to improve filtering techniques in order to keep only words that will help us to create a well-formulated RDF statement (figure 2). As we deal with text – whether the content of the knowledge base or the user's question – we apply linguistic rules in order to filter unimportant words. Unimportant words do not contribute to the meaning of the sentence. Consider the following question: 'Who built the very first calculator?'. Here, examples of such unimportant words are 'the', 'very' and 'first'. In general, we can create a list of such words or phrases that have to be filtered out of the sentence. They are mainly adjectives and adverbs, but also sub-clauses. What remains should be items with a precise grammatical function within the sentence such as *subject*, *verb* and *object*. Thus, we will be very close to an RDF statement.

Next, we have to find the verb and transform it into infinitive. Generally, this can be done whether by applying linguistic rules – depending of course on the language – or in a simpler way, by performing a keyword search on a comprehensive list of verbs. Such lists exist for instance in public dictionaries. The remaining sentence in our example will be: 'Who (to) build calculator?'.

The next step is to transform the sentence into an RDF statement. In other words, what is the *resource*, the *predicate* and the *object*? The different words of the sentence are compared to the RDF schema elements. Unfortunately, we will get to the familiar problem of synonyms. In our example, the word 'build' does not exist as a separate element in our schema. The solution we propose is to find the corresponding synonym for words of the original (non-formula) sentence. The inference engine searches in the list of verbs and resources for the word 'build' (see table 1 and table 2). It finds a match in the list of verbs and concludes:

- 'build' is a verb, thus an RDF predicate and not a RDF resource.
- The RDF element for 'build' is `dc:creator`.

The same is done with every other word. So 'calculator' is found to be the resource with the RDF element `chest:computer`. The word 'who' will remain without any matching element. The inference engine concludes that this is the missing part in the statement. Finally, the query-statement will become: `<?x>;<dc:creator>;<chest:computer>`

This well-formulated RDF query can be launched against the knowledge base, using a query language like RDFQL [14] [2] [1]. The final results will be the clips, covering the biography of the builders of computers . The question word itself can indicate what kind of information the user is looking for.

- 'Who': the user is looking for a person. Thus, the missing part is the object (an RDF resource). The query has the form: `<a>;<b>;<?x>`.
- 'When': the user is looking for a date. In this case, the requested data is stored as metadata (for example: `chest:year_birth`) and is not a resource.

- In any other case, the user is looking for the subject (an RDF resource). The query has the form: `<?x>;<a>;<b>`.

Table 1. Example of RDF predicate synonyms

Verb	RDF predicate
…	…
build	`dc:creator`
construct	`dc:creator`
invent	`dc:creator`
…	…

Table 2. Example of RDF resource synonyms

Object	RDF resource
…	…
calculator	`chest:computer`
computer	`chest:computer`
Linux	`chest:OS`
…	…

4. OUTLOOK

The prototype CHESt is to be tested in some selected schools in Luxembourg in the beginning of the year 2004. The collected data will help to improve the tool, either by adding or changing metadata or by improving search mechanisms. A more general aim is to test and study the development of our inference engine in order to create a general semantic search mechanism, applicable not only for this tool, but also for other purposes, for instance the world wide web.

REFERENCES

1. Agrawal R. et al, 2001, *Storage and Querying of E-Commerce Data*. In proceedings of the 27th VLDB Conferences, Roma, Italy
2. Broekstra J. et al, 2000, *On-To-Knowledge. The State of the Art on Representation and Query Languages for Semistructured Data*. Administrator Nederland b.v.
3. Schillings V. et al, *Tele-TASK, Teleteaching Anywhere Solution Kit*. Universität Trier. `http://www.tele-task.de/`
4. *Dublin Core Metadata Initiative (DCMI)*. `http://dublincore.org`
5. Fensel D. ,2004, *Ontologies, A Silver Bullet for Knowledge Management and Electronic Commerce*. Second Edition. Springer-Verlag, Berlin Heidelberg New York, p. 37-39
6. Hendler J. et al, 2003, *Integrating Applications on the Semantic Web*, In Journal of the Institute of Electrical Engineers of Japan, Vol 122(10), p. 676-680
7. Hjelm J., 2001, *Creating the Semantic Web with RDF*. John Wiley & Sons, Inc. Canda
8. Linckels S., 2003, *CHESt namespace*, `http://www.linckels.lu/chest/elements/1.0/`
9. Ma M. et al, 2004, *T-Cube, A Multimedia Authoring System for eLearning*. In proceedings of the AACE E-Learn 2003, Phoenix, USA, p. 2289-2296
10. Mayer R., Gallini J.,1994, *When Is an Illustration Worth Ten Thousand Words?* In Journal of Educational Psychology, Vol. 86, No. 3, American Psychological Association Inc.
11. Mayer R., Sims V., 1994, *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.
12. Meinel Ch. et al, 2003, *Overcoming Technical Frustrations in Distance Education: Tele-TASK*. In proceedings of e-Society 2003, Lisboa, Portugal, p. 34-41
13. Meinel Ch., Schillings V., 2002, *Tele-TASK - Teleteaching Anywhere Solution Kit*. In proceedings of ACM SIGUCCS 2002, Providence, USA, p. 130-133
14. Patel K., Gupta G., 2003, *Semantic Processing of the Semantic Web*. In proceedings of the Second International Semantic Web Conference, Sanibel Island, USA
15. World Wide Web Consortium, *Resource Description Framework (RDF) / W3C Semantic Web*, `http://www.w3.org/RDF/`