



Exploiting Semantic Web and Knowledge Management Technologies for E-learning

Sylvain Dehors

► To cite this version:

Sylvain Dehors. Exploiting Semantic Web and Knowledge Management Technologies for E-learning. Human-Computer Interaction [cs.HC]. Université Nice Sophia Antipolis, 2007. English. NNT: . tel-00134114v1

HAL Id: tel-00134114

<https://theses.hal.science/tel-00134114v1>

Submitted on 28 Feb 2007 (v1), last revised 28 Feb 2007 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS - UFR Sciences

Ecole Doctorale STIC

T H E S E

pour obtenir le titre de
Docteur en Sciences
de l'UNIVERSITE de Nice-Sophia Antipolis

Discipline : Informatique

présentée et soutenue par

Sylvain DEHORS

Exploiting Semantic Web and Knowledge Management Technologies for E-learning

Thèse dirigée par Rose DIENG-KUNTZ

soutenue le 2 février 2007

Jury : °

M. Cyrille DESMOULINS

Mme Monique GRANDBASTIEN

Mme Danièle HERIN

M. Wolfgang NEJDL

M. Peter SANDER

Examineur

Rapporteur

Rapporteur

Rapporteur

Président

1. INTRODUCTION	7
1.1. <i>Historical background</i>	7
1.1.1. Knowledge supports	7
1.1.2. Learning and teaching	7
1.2. <i>E-learning and Knowledge Management</i>	8
1.3. <i>Plan</i>	10
2. CONTEXT	11
2.1. <i>Research Areas</i>	11
2.1.1. Knowledge management	11
2.1.2. Semantic Web.....	11
2.1.3. E-learning	11
2.1.4. A multi-disciplinary approach.....	12
2.2. <i>Institutional context</i>	13
2.3. <i>Scenarios</i>	13
2.3.1. Categorization of scenarios	14
2.3.2. Academic scenarios	16
2.3.3. Professional Scenarios.....	16
2.3.4. Other examples of e-learning scenarios	17
2.4. <i>Research Questions</i>	19
2.4.1. Formalized knowledge in the learning context	19
2.4.2. Learning with the semantic web.....	19
3. LITERATURE REVIEW	21
3.1. <i>Learning with computers, Computer Assisted Learning</i>	22
3.1.1. Learning theories	22
3.1.2. The Learning Object approach	25
3.1.3. E-learning Standards	28
3.1.4. User Modeling.....	31
3.1.5. Learning Objects and pedagogy	33
3.1.6. E-learning systems	34
3.1.7. Intelligent tutors and adaptive systems	35
3.2. <i>Knowledge Management and E-learning</i>	41
3.2.1. Organizational memory: inspiration and analogies.....	41
3.2.2. Ontologies	43
3.2.3. Annotations	51
3.3. <i>Semantic Web</i>	52
3.3.1. The vision	52
3.3.2. Standards	53
3.3.3. Architectures for the semantic web	56
3.3.4. Tools for the semantic web	56
3.4. <i>Semantic web for e-learning</i>	60
3.4.1. Visions.....	60
3.4.2. Inspiring projects.....	61
3.5. <i>Conclusion</i>	63
3.5.1. Open Vs Closed worlds.....	63
3.5.2. Standards	63
3.5.3. Reusing resources.....	64

4. DESIGN AND OVERALL DESCRIPTION	65
4.1. <i>Analysis framework for the design approach</i>	66
4.2. <i>Scenarios for learning with web resources</i>	67
4.2.1. Motivations for using and creating web resources	67
4.2.2. Reusing digital content.....	68
4.3. <i>Question Based Learning Strategy</i>	69
4.3.1. Addressed problems	69
4.3.2. Strategies based on questions	69
4.3.3. Theories and contributions justifying the strategy	70
4.4. <i>Using semantic representation in QBLS</i>	71
4.4.1. Enhancing the browsing of course content.....	71
4.4.2. Conceptual navigation.....	73
4.4.3. Semantic representation for knowledge sharing	76
4.4.4. Conclusion.....	77
4.5. <i>QBLS Overview</i>	77
4.5.1. A system build around three components	78
4.5.2. System Description.....	78
5. SEMANTIZING COURSE DOCUMENTS	81
5.1. <i>Overview of semantization</i>	82
5.1.1. Existing problems and actual solutions.....	82
5.1.2. Necessary appropriations by the teacher	83
5.1.3. Separation of content and presentation	84
5.1.4. Characteristics of the semantization task	85
5.2. <i>Proposed semantization method</i>	88
5.2.1. Step 1: Content modeling	90
5.2.2. Step 2: Exploiting semiotic markers linking to semantics	94
5.2.3. Step 3: Automatic annotation	96
5.2.4. Step 4: Manual annotation.....	97
5.2.5. Step 5: Production of formalized knowledge	100
5.3. <i>Evaluation</i>	104
5.3.1. Reusing existing material	104
5.3.2. Qualitative view of the produced annotation	105
5.3.3. Quantitative data on two experiments	106
5.3.4. A graph perspective for domain model	108
5.3.5. Difficulties met.....	111
5.4. <i>Impacts</i>	115
5.4.1. Semantization cycle.....	115
5.4.2. Ontology interoperability	116
5.4.3. Teachers' role and learning technologies	120
5.4.4. Redefining conceptual navigation	121
5.5. <i>Conclusions</i>	123
6. EXPLOITING SEMANTIC WEB TECHNOLOGY	125
6.1. <i>QBLS: A Semantic Web infrastructure</i>	126
6.1.1. Architecture	126
6.1.2. Interaction cycle	127
6.2. <i>Semantic Interfaces</i>	128
6.2.1. Course interface.....	128
6.2.2. Annotation editor.....	132

6.3. <i>Semantic querying for learning interaction</i>	134
6.3.1. Semantic querying on domain knowledge	135
6.3.2. Including pedagogical knowledge	137
6.3.3. Retrieving literal information	142
6.3.4. Context awareness	143
6.3.5. Domain/range querying	143
6.4. <i>Semantic query processing</i>	144
6.4.1. Corese specificities	144
6.4.2. Dynamic aspects	145
6.4.3. Rule standardization	145
6.5. <i>Interface generation</i>	145
6.5.1. Integration of XML technologies	145
6.5.2. Dynamic pages	146
6.5.3. Detailed example of interface construction	146
6.5.4. Editor interface	148
6.5.5. Evaluation of the technical solution proposed	149
6.6. <i>Conclusions</i>	149
7. TOWARDS A SEMANTIC ANALYSIS OF LEARNING PATHS.....	151
7.1. <i>Collecting activity traces</i>	152
7.1.1. Knowledge collected on learner activity	152
7.1.2. Representation of learner activity	153
7.2. <i>System adaptation based on activity traces for learners</i>	156
7.2.1. Exploitation of learner models	156
7.2.2. Conceptual back navigation	157
7.2.3. Evaluation of adaptation mechanisms	160
7.3. <i>Teacher's interpretation of activity traces</i>	161
7.3.1. Teacher's interrogations	161
7.3.2. Visualization of the course in the log analyzer	161
7.3.3. Structural views	164
7.3.4. Representation and interpretation of single user navigation	166
7.3.5. Aggregated views for analyzing behaviors in groups	169
7.3.6. Evaluation of the log analyzer tool	172
7.4. <i>More automated analysis</i>	172
7.4.1. Statistical measures	172
7.4.2. Graph patterns	175
7.5. <i>Conclusion</i>	176
7.5.1. Outcomes of our proposed graph navigation model	176
7.5.2. Uses of interpreted paths	177
7.5.3. A broader perspective through the exchange of log data	177
8. EVALUATION: METHOD AND RESULTS	179
8.1. <i>Generic evaluation of a learning system</i>	180
8.1.1. Specificities of evaluation for learning systems	180
8.1.2. Evaluation focus	180
8.2. <i>Description of the experiments</i>	181
8.2.1. QBLS-1	181
8.2.2. QBLS-2	181
8.3. <i>Evaluation methods</i>	183
8.3.1. Visual feedback through direct observation	183

8.3.2. End-user questionnaires	183
8.3.3. Activity reporting	184
8.3.4. Answer sheets.....	185
8.4. <i>Results and observations</i>	186
8.4.1. Evaluation framework	186
8.4.2. Usage	186
8.4.3. Usability	189
8.4.4. Acceptability	191
8.5. <i>Synthesis</i>	192
8.5.1. Pedagogical considerations	192
8.5.2. Course reuse	193
8.5.3. Navigation model	194
8.5.4. Global setting	194
8.6. <i>Evaluation framework and generalization of the results</i>	194
8.6.1. Specific methods for specific conclusions	194
8.6.2. Best practices for evaluating e-learning systems	195
8.6.3. Recommendations for evaluating semantic web systems	196
8.7. <i>Conclusion</i>	197
9. GENERAL METHODOLOGICAL GUIDE	199
9.1. <i>From annotation to log analysis: a complete method</i>	200
9.1.1. Method overview.....	200
9.1.2. Final outcome	203
9.2. <i>Positioning</i>	203
9.2.1. KM methodologies	203
9.2.2. A Semantic Web approach	204
9.2.3. Learning paradigm	204
9.3. <i>The ASPL integration</i>	205
9.3.1. Exploitation of existing documents.....	205
9.3.2. Semantization of contents	205
9.3.3. Exploitation through semantic services.....	206
9.3.4. Analysis	210
9.3.5. Perspectives	210
10. CONCLUSION	211
10.1. <i>Scientific conclusions</i>	211
10.1.1. Semantics for learning.....	211
10.1.2. Semantic web approach.....	212
10.2. <i>Results</i>	212
10.2.1. A methodological guide	213
10.2.2. A showcase implementation.....	213
10.2.3. Burying the Learning Object paradigm?	213
10.3. <i>Perspectives</i>	213
10.3.1. Integration of annotation tools	213
10.3.2. Towards complex adaptive engines on the semantic web.....	213
10.3.3. The future of e-learning.....	214
REFERENCES	215
<i>Publications</i>	215
<i>Web resources</i>	225

<i>Index of Figures</i>	227
<i>Index of Tables</i>	228
ANNEXES	230

1. INTRODUCTION

1.1. *Historical background*

1.1.1. Knowledge supports

Before dealing with serious matters I would like to recall a short history about the human dealings with knowledge, I am afraid to say, in a very western world perspective.

Prior to the invention of writing, humans exchanged knowledge between them through oral and gesture communication. Historians commonly place the first occurrences of writings in the Neolithic Fertile Crescent, 9000 years ago. With the famous Egyptian hieroglyphs, drawn on papyrus paper, knowledge started to be stored on inert, physical and mobile supports. Libraries, like in Alexandria, started to gather significant pieces of writing and to build a memory that would last over the centuries. Unfortunately, history proved this approach not exactly flawless, especially because of the uniqueness of the pieces. Inspired by the tale of the disaster that destroyed this first centralized attempt at building a knowledge base, millennia later in Europe, monks became the guardians of knowledge. They were replicating existing manuscripts and storing them in various locations to ensure their transmission over middle age. As keepers of this knowledge, they were also in charge of teaching future generations reading and writing so that knowledge would not be lost like the meaning of hieroglyphs. However, manual copy was slow and not as accurate as it should.

The fifteen century marks the beginning of a new era with the invention of printing. Through the dissemination of books, knowledge became suddenly accessible to anyone who could read. Schools were developed, and scientific knowledge of all kinds increased dramatically. For centuries the situation remained stable, with printed books as the favored repository for knowledge, like Diderot's famous encyclopedia. Books were the main source of learning support. However, printed materials are not sufficient for an efficient learning and teachers kept their powerful position of knowledge transmitters. Considering the master/student relationship, things remained unchanged from the early days of our Greek inspired civilization.

At the dawn of the third millennium (which means roughly ten years ago), happens an unprecedented and overwhelming revolution with the entrance in the "*digital age*". The possibility to replicate book content in computer memory becomes unlimited. Data in digital form is exchanged throughout the worldwide network with an unprecedented ease. This technological breakthrough, like printing in its time, triggers new practices and ways of dealing with knowledge and learning. Today, we can consider the World Wide Web as the privileged way to access/exchange and propagate knowledge.

1.1.2. Learning and teaching

Understanding human intelligence has been a constant preoccupation of scientists. This subject has been considerably developed, offering better understanding about learning and its relationship with knowledge in general, and lately with knowledge in digital form. Getting to know is part of intelligence, and intelligence cannot be separated from knowledge. By knowledge we designate not only what can be written down in books, but also competencies, attitude, expertise, etc. that are equally important.

The ancient Greeks already analyzed and discussed teaching methods. Many theories have been produced since. The latest ones, based on advances in brain neurosciences, are still at their early stage. What we learned for sure is that there is no definite way to learn. Such a complex process is deeply rooted in both physiological and psychological origins.

For the newborn, the path to knowledge is long and difficult. Acquiring knowledge, or learning, is both natural but demanding. Why cannot we just learn at once and then remember forever, like in so many science fictions movies? Unfortunately, the human brain is not that simple. We shall never learn easily, but our curiosity and imagination still drives us to look for always better, easier, faster ways of learning. The sought for progress may be what defines human nature. In our “modern” world this quest is called research and is conducted by men dressed in white with thick bubbly glasses and microscopes. Well ... mostly.

Another aspect is the sociological facet of learning. The way we learn is both motivated and guided by the society we live in. Plato did not learn from Socrates the same way our teenagers discover geography with Google-earth. Today it is natural to observe learning taking place using the digital artifacts surrounding us. However, as we are only on the doorstep of this digital era many interrogations subsist. I hope the short story presented, will give the reader at least a glimpse of the incredible importance and impact of this subject. Of course we shall discuss this deeper all along the manuscript. For this introduction, we just moor on the iceberg of learning alongside its emerging shore.



Figure 1 – “iceberg”, retrieved from the World Wide Web, 26/07/2006

1.2. E-learning and Knowledge Management

The broad term of “e-learning” designates for us any activity related to learning and apprenticeship through digital media.

At first, the buzz generated by this term has blinded the technological breakthrough behind it: the increase in the power of personal computers. However, it never meant that learning will become effortless, or that it will automatically provide “*just-in-time education integrated with high value chains.*” (Drucker, 00). Certainly, the range of possibilities now offered goes beyond imagination.

There is both a revolution and very little novelty in this term. For the way people learn, it is a true revolution that we shall detail later. Presumably, the shift from a master / student relationship to a computer mediated, and assisted, course introduces a radical turn in learning practices. Still, there is no magic about it. It requires time and effort to learn, playgrounds do not have to worry about frequentation.

When thinking about the impact of society on the way people learn, the current move towards more flexible, adaptable, and mobile persons finds its counter part in a growing need for adult’s education, professional training and even third age university. The term “*life-long learning*” was introduced to describe this evolution. It supposes that we can no longer rely on early education at school. The evolution pace of the techniques, knowledge, etc has increased so much that the validity period of acquired competences is now much shorter than human lifetime (that increases on the other hand). This life-long learning perspective, now commonly accepted, is seen as one of the main arguments in favor of e-learning. The “any-time, any-where” slogan takes here all its meaning, as learning will have to take place, both on the workplace and at home, and all stages of life.

The industry of e-learning is a flourishing one. Answering a real need of companies to update their staff competencies, and considering the difficulties to combine constraints of classical education (face to face with teachers, manual assignments, etc.) with corporate agendas. The past years have seen a whole industry spring from a ground of small distance education businesses. Economical analysts give various figures about the expansion of this activity, difficult to monitor due to its multiple forms but undeniably following a strong growing curve. For example, technology analysts report that the U.S. market for corporate e-learning is expected to reach approximately US\$10.6 billion by 2007, according to research firm IDC.

“The real bottom line is that whether they need to brush up their skills on a product or learn about a new technology, [...] professionals no longer must spend days away from work, sitting in a classroom. Thanks to a host of well-established and new companies, ambitious technology employees today have access to a wide world of information, labs, educators and research material that can help improve their productivity, capabilities and employability -- all without requiring them to leave their desks.”

Excerpt from <http://www.technewsworld.com/story/32325.html>

Philosophers (Villette, 99) see the use of information technologies for teaching and learning as a revolution in teaching practices.

➤ At the academic level, in schools or universities, such modifications affect the teaching activity itself, but also the role of the teacher. Indeed, when the course takes place remotely, most of the time learners are facing screens and not a physical person. The teacher becomes a mediator between the source of knowledge and the learner. He/she is not the owner of knowledge anymore but the “broker” that guides the learner in his new apprenticeship. Some thinkers express worries about this virtualization of teaching that could create a fake relationship towards reality. Still, the focus is now put on learning more than teaching, which is a revolution compared to previous practices, and illustrates one of the deepest impacts of information technologies for learning. By putting aside the teacher, e-learning focuses more on the learner and his/her crucial activity.

➤ At the professional level, such technologies are well spread. Adoption of new tools is faster there than in the academic context. However, they bring another type of modification in the way companies work. The strict assignment of roles and the taylorist approach are outdated. They only subsist because of organization rigidity and slowly leave place for an “open” organization where workers share a common production goal. This implies continuous training, and pedagogy grounded in reality. Investments are necessary. For example, in an industrial domain it might not be suitable to learn using production tools, but simplified mock-ups. However, those intermediate, potentially virtual, artifacts do not exist yet.

The corporate domain uses the term “*learning organization*” along with the idea of “*life-long learning*”. Companies consider strategic to manage knowledge and competencies that now constitute its main asset. In this scope, they may deploy organizational memories that tightly connect to the training efforts. Memories must evolve with the company and such evolution is now so fast that stability might become an issue if the learning pace increases. Candidates are judged on their attitude over diploma and expertise to ensure that they will pace up with the organization.

Such learning organization shows the deep connection existing between learning and knowledge management in the corporate context. The term “knowledge management” is usually identifies practices of sharing, making explicit and preserving knowledge. These goals

have much in common with professional training, aiming at giving employees new practical knowledge for their work.

With the breakthrough of digital technologies, knowledge management (KM) has emerged as a more technical discipline for managing explicit knowledge in an organization. The knowledge manager tool-box contains programs for managing not only content but also all kinds of digital information (e.g. ERP, CRM, etc.). Those tools implement the methods and practices of KM. They may be specific to an organization or defined as standards. Such programs rely on formalisms and specific algorithms to express and manipulate knowledge, and that is what we shall look at, in the context of e-learning.

1.3. Plan

Keeping in mind the above considerations, we shall use the following plan to present our work and contributions.

- First, we precise the scientific context of this work. The key issues are described and we identify major trends and inspirations. We position ourselves in this context giving the novice reader a better understanding of the proposed scientific contributions.
- Then, an overview of the relevant literature is given. The very large extent of the research domain of e-learning coupled with a well established research community in knowledge management gives us a huge number of relevant publications. We selected the presented contributions based on their link with a technological or a conceptual aspect of our own work. We also selected only major works, based on publications in leading international journals or conferences.
- The fourth chapter presents the general approach for designing the e-learning system at the heart of this thesis. We focus on the analysis of the needs expressed by the teachers we met and on the general architecture we adopted. Details of this architecture that lead to scientific contributions are the subject of the three following chapters.
- We explain why an annotation process was necessary for the automatic exploitation of documents, and detail our proposed solution, along with the experimentations we conducted and the proposed general annotation framework.
- Being one of the motivations for this work, semantic web technologies constitute the ground level on which the application is built. We precisely explain the functionalities brought by such technologies that proved effectively useful. This section details their implementation, potential improvements, and what outcomes can be sought in general.
- Browsing annotated courses brings different ways of learning, but also different ways to look at this activity. We propose an analysis model to better understand and interpret users' path on such spaces. This contributes to the overall evaluation, but also sketches out interesting answers and perspectives for learner modeling, tracking, and assessment.
- A separate section presents the evaluation of the complete system. Reusability and generalization of the tool are thoroughly discussed there.
- Finally, we sum up the global picture to propose a detailed generic method for semantizing on-line courses in the context of computer assisted teaching.

We want to stress the fact that, from the beginning, our vision is strongly driven by application and real-world feasibility. Given the state of current research, the gap between research advances and day to day practice is so wide that we felt this was the most needed contribution. This is illustrated by the slogan of the Knowledge Web European NoE supporting this work: "Realizing the semantic web". Aspects like scalability, usability and reusability will be of prime importance. We will also emphasize the generalization aspects of the presented experiments, tools and methods.

2. CONTEXT

2.1. Research Areas

The research work presented in this thesis is rooted in the formerly called “artificial intelligence” domain. Originally, artificial intelligence was concerned with the mechanization of reasoning and machines that could present so-called “intelligent” behaviors. Because its name raised such expectations, this research domain received much criticism. However, several results are now largely recognized as major breakthroughs. Practical tools like classification algorithms, natural language processing, logic reasoning, etc. proved useful in a wide range of applications. It now feeds applied research areas such as Knowledge Management, Semantic Web and E-learning.

2.1.1. Knowledge management

Knowledge management is the collection of processes that govern the creation, dissemination, and utilization of knowledge. This domain currently focuses a lot of attention from the industrial world because of its direct application in the corporate domain. For (Steels, 93) *“The objective of a knowledge management structure is to promote knowledge growth, promote knowledge communication, and in general preserve knowledge within the organisation”*. We shall see that learning is obviously concerned by knowledge communication but that knowledge growth and preservation are also important.

We consider that the representation and formalization of knowledge is amongst the successful contributions of AI, and is illustrated by the development of knowledge management applications in the corporate domain (Dieng-Kuntz, 04).

2.1.2. Semantic Web

In parallel, the development of internet and the World Wide Web has raised many interests to bring artificial intelligence results up to the dimension of the whole web. The idea of “Semantic Web” presents a particular vision of the web where the network does not only targets the transfer of data but of knowledge. Tim Berners Lee express this in a snippet from his presentation (dates back to 1994): *“For example, a document might describe a person. The title document to a house describes a house and also the ownership relation with a person. Adding semantics to the web involves two things: allowing documents which have information in machine-readable forms, and allowing links to be created with relationship values. Only when we have this extra level of semantics will we be able to use computer power to help us exploit the information to a greater extent than our own reading.”* (Berners Lee, 94).

Much publicized through the famous near science-fi article published in *Scientific American* (Berners-Lee et al., 01), the real development of the semantic web starts with the standards proposed by the World Wide Web consortium (W3C, 06). This standardization framework aims at representing and manipulating knowledge on a world-wide basis, a dimension no other formalism had in mind before. It raises issues directly linked to knowledge management, like the integration and interoperability of formalized knowledge and scalability problems due to the unprecedented size and “open” nature of the web.

2.1.3. E-learning

The research domain of “e-learning” is very broad. It is also the object of a thriving industrial activity and e-learning research issues could be described as questions about the adaptation of education practices with today technology.

The range of existing “e-learning” applications does not help us in describing this domain more precisely as it includes:

- Course management systems, a field mostly covered by industrial products
- Intelligent tutors, adaptive hypermedia, a very research oriented domain
- Collaborative tools used for learning, video-conferencing, etc.
- Digital exams, on-line quizz, etc.

Since the beginnings of e-learning, AI techniques have been tried to enhance the learning experience. Several reasons justify it, some are quite complex, others rely on rather practical considerations. First, most researchers also occupy teaching positions in universities. It is quite natural that AI researchers, looking for application domains, start to apply their ideas on the handy population of students attending their courses. Hopefully, the very complex and challenging nature of human learning is also one of the reasons why AI techniques are applied in this context. This constant research field is illustrated by the famous AI-ED (Artificial Intelligence in Education) acronym, used by major conferences and journals in the domain.

2.1.4. A multi-disciplinary approach

This thesis can be placed at the frontier between those three main research domains involving artificial intelligence:

- Knowledge management
- Semantic Web
- E-learning and in particular AI-ED.

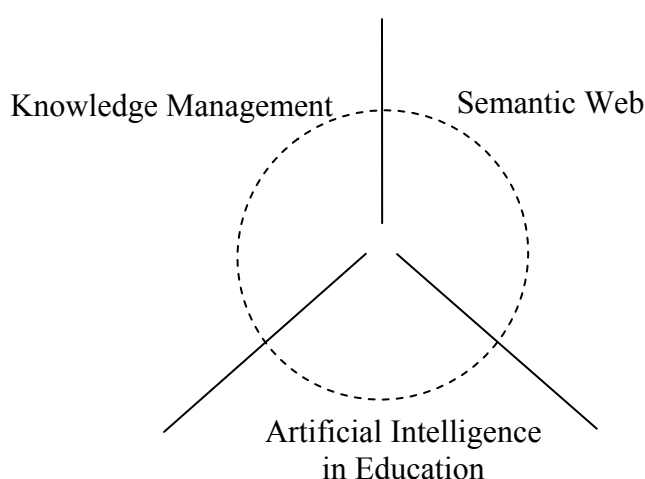


Figure 2 – Research context of the thesis

The connection between those fields is further illustrated by the current move towards web globalization. The “pervasive and ubiquitous” computing, apart from being buzzwords, designate a true evolution of our society now taking place. Most acts of life will soon involve a connected environment. This convergence is illustrated by the current status of the three research domains mentioned above, which have much cross-fertilized in the past years.

We have reached the point, land-marked by this manuscript, where the frontiers are about to fade away. Already, as we shall see later, the number of theoretical proposals mixing semantic web, e-learning, and knowledge management is quite large. For information, and a bit of fun, here is a short list of titles from published articles on the domain:

- “e-learning in the semantic age”
- “e-learning based on the Semantic Web”
- “using semantic web technology for e-learning”
- “towards e-learning via the Semantic Web”

- “the New Challenges for E-learning : The Educational Semantic Web”
- “Education and the Semantic Web”
- “E-Learning Has to be Seen as Part of General Knowledge Management”
- Etc.

The following list of acronyms, the extent of which leaves us in perplexity, roughly covers the range of applications we are looking at. All the different names below refer to “learning with a computer”:

- CAI, Computer Aided Instruction
- CAL, Computer Assisted Learning
- CSCL, Computer Supported Collaborative Learning
- CBT, Computer Based Training
- ICAI, Intelligent Computer Aided Instruction
- WBE, Web Based Education
- WBT, Web based Training

Our contribution takes place within this research context. In the following sections (2.3, 2.4), we shall detail the specific scenarios and research question we are specifically tackling. We already mentioned that the focus will be placed on the exploitation of course documents and that issues about virtual reality, social agents or collaboration for learning are totally out of our scope.

2.2. Institutional context

For twelve years the ACACIA team of INRIA Sophia Antipolis, lead by Rose Dieng-Kuntz, has open the way for a better exploitation of knowledge among organizations. Its main contributions concern the acquisition of knowledge from multiple sources, and the way to store and access it (knowledge-based system, knowledge server, corporate memory and corporate semantic web). In this context the team wanted to settle on a new challenge and broaden its horizon by tackling the difficult problem of managing knowledge not only for the purpose of sharing and conserving it inside an organization but also for transmitting it to humans, in a nutshell for learning.

This work was mainly carried out through three collaborations:

- The Weblearn action gathered French research teams on the subject of semantic web for e-learning. The objective of this group was to produce a state of the art of the actual research situation on semantic web for e-learning and to offer a prospective view on this research field.
- The Knowledge Web European network of excellence (KW, 06) animates the European semantic web research community. Some tasks inside the project specifically target educational applications of semantic web. In particular, a learning resource repository for the domain of semantic web itself has been set up. A demonstration platform for e-learning based on semantic web technologies is also in development.
- Finally, a close collaboration with teachers and researchers from the Mainline team at the near by Polytech’Nice school of engineering has led to several experiments on the use of on-line courses, reported in this work.

The experiments we conducted all concern academic teaching. However, given the generic technology and method applied, we believe that some results at least could be transferable to industrial education.

2.3. Scenarios

To situate the context of our study, it is important to define the various scenarios and situations of e-learning where knowledge management techniques might be useful. This

question of scenario is crucial for learning and choosing the right learning scenario is part of pedagogical expertise. There is no definite answer on what is the good way to learn this or that. The introduction of computers in this process only adds to the problem.

Defining a scenario is quite simple but may take multiple forms. As we shall see, services for an e-learning system may go far beyond the simple delivery of course content and traditional ways of teaching and learning in schools. For example, possibilities offered by the application of knowledge management to e-learning outdate the classical teacher-learner relationship as predicted by philosophers (Villette, 99).

The primary question remains to ensure the effectiveness of the scenario, a problem that cannot be generically answered. Our first step will be to identify the different scenarios and classify them to get a clearer view of the possible orientations in the research context.

2.3.1. Categorization of scenarios

There is a multiplicity of possible scenarios, depending on their outcomes, goals, tools or learning situation. Each scenario calls upon one or several learning theories, a learner model, etc. To try categorizing the different scenarios, we propose the following axes:

➤ Scenarios can be categorized according to the knowledge involved and the goal pursued. For example the following axes are used by (Altenhofen and Schaper, 02) : What, How, Why, Where. This is further refined in the ontology presented on figure 3. The four main axis of this ontology match the following types of knowledge:

- Orientation Knowledge
- Action Knowledge
- Explanation Knowledge
- Reference Knowledge

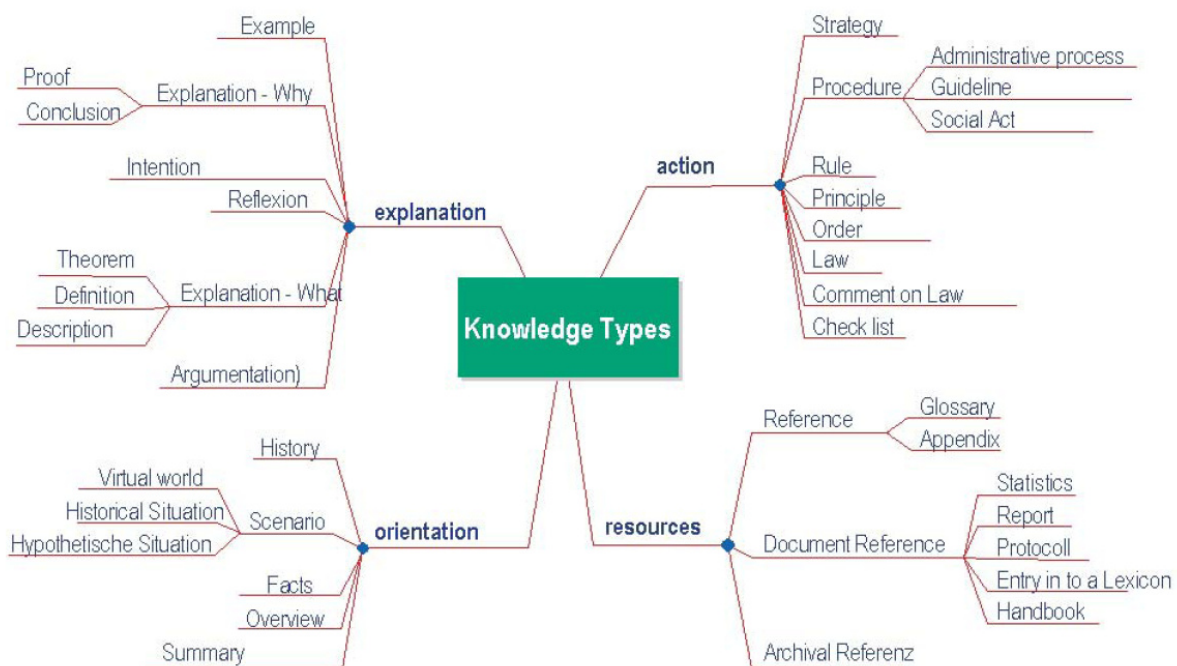


Figure 3 - Ontology of knowledge types, (Altenhofen and Schaper, 02).

➤ The learning activity can be used as a differentiator. For example, 14 activities are listed by (Dalgarno, 98): attending to static information, controlling media, navigating the system, answering questions, attending to question feedback, exploring a world, measuring in a world, manipulating a world, constructing in a world, attending to world changes, articulating, processing data, attending to processed data, formatting output.

- Scenarios may rely on a learning theory, like the Instructional Transaction Theory (Merrill, 99). For a more complete discussion on learning theories, see chapter 3.
- Finally, the origins and goals of the knowledge formalized in the scenario also offers an interesting distinction. For example, scenarios involving the use of formalized knowledge (formal memory) can be viewed through the grid proposed by (Azouaou et al., 03). This grid separates users that create and use annotations on pedagogical documents, either individually or collectively. Table 1 presents a grid adapted from (Azouaou et al., 03). It places authors in column, and users (targets) in line. The following scenarios are thus identified:
 - Learning: scenario of understanding the course content.
 - Feedback: scenario where information on the learning process is used by pedagogical actors (e.g. through adaptation of the difficulty during the course).
 - Evaluation: scenario where formalized knowledge helps the learner or the learning group directly (e.g. by providing hints).
 - Revision: internal scenario for teachers to improve their action.

Table 1 – Usage scenarios for each actor (adapted after (Azouaou et al., 03))

		Individual		Collective	
		Learner	Teacher	Pedagogical team	Class/Group
Individual	Learner	Learning	Feedback	Feedback	Learning
	Teacher	Evaluation	Revision	Revision	Evaluation
Collective	Pedagogical team	Evaluation	Revision	Revision	Evaluation
	Class/group	Learning	Feedback	Feedback	Learning

On table 1, users are separated in two broad categories (teacher/learner). This can be refined into four categories according to (Schneider et al., 03): Teacher, learner, computer and designer. The possible interactions are then multiple. Communication channels shown on this grid define the role of each agent. If channels are mediated by digital tools, roles may be restricted and better defined. Their relations with the different pedagogical artifacts (goals, planning, monitoring, contents and tools) can also describe the participants as shown in table 2.

Table 2 – Role of the participants depending on the pedagogical artifacts, after (Schneider et al., 03)

	Teacher	Learner	Computer	Designer
Goals	help or define	define or refine	Run the “managing tools”	bring ideas and models
Planning	suggest and control	Do and execute		
Monitoring	audits, help on demand	self-observation		Observe
Contents	suggest, produce	use, also produce	store, search, push	provide and develop
Tools	configure, help, suggest	select, train, use	Support	

Another practical differentiator is the clear separation that occurs between academic and professional domains. We adopt this separation to present, in the following two sections, a few examples of scenarios.

2.3.2. Academic scenarios

In the domain of schools in general, pedagogical practices have been in place for a long time. It is estimated that lecturing has been a favored teaching technique since the fifteenth century. Nowadays various pedagogical activities (lectures, assignments, labs, projects, exams, etc.) are proposed. With (Michau and Ploix, 03) we can identify the domains where computers play the biggest role and propose the following scenarios as a selection of examples of “e-learning”.

- The most common scenario is the distance-learning one, as applied in “Open Universities”. Students scattered around the globe can only communicate and access course documents through their machine. The characteristics of the course might not be very different from classical distance education using postal service and paper supports if the computer is just used as a convenient communication channel. In this scenario, the advantages brought by computers are undeniable (speed, richness of the content, etc.).
- A second scenario is the support to project-based learning. This scenario puts forward the freedom offered to students and the possibility to manipulate industrial tools. In this scope, working with computers, turns the student away from the teacher’s influence. Doing this, he/she gains autonomy which is the essence of project-based learning. Learning applications might integrate services that “hide” teacher’s monitoring and guiding to keep things balanced.
- The same idea is present with scenarios targeting collaborative activity. Digital media are easier to share, and collaborative activities can engage students in communication and community building around common learning objectives. Computers may support this exchange and the circulation of knowledge, and thus help learning.
- Finally, scenarios inspired by traditional teaching methods (e.g. lectures, labs) constitute the outer fringe of what is currently accepted in practice. Advantages of digital media in that kind of settings are still to be clearly demonstrated. This is the range of scenarios we are particularly interested in. For example in the following, we investigate the scenario of replacing a classical face-to-face lecture by an on-line access to course contents.

2.3.3. Professional Scenarios

The professional environment presents some specificity compared to the academic world. In professional scenarios of e-learning, the existing practices play an important role. In this domain, the term “training” is used instead of “teaching”. Besides, most companies already possess that training experience. On the theoretical level, active or “hands-on” approaches (often related to the constructivist proposal) seems to reduce the gap between schools and companies. Still it differs a lot from academic practice, and with e-learning the risk would be to consider that a scenario at school would work identically in a corporate environment. In the professional context too, the introduction of computers triggers an evolution of existing practices.

This evolution targets two main goals:

- Quality and efficiency of the acquired knowledge,
- Cost of the training.

Issues of cost, reactivity and evaluation must be considered differently in the corporate context. The notion of return on investments of a training for example is quite specific.

Professional needs vary a lot (ex: mechanics training (Desmoulins and GrandBastien, 02), commercials/engineers (KW, 06)). The specificity of the professional environment lies in the few training modes compatible with professional activity. We present below a brief overview of three classical training modes in the corporate context, and list the pros and cons of each mode:

- Face-to-face training/seminars
 - Short learning time
 - Focused on precise knowledge
 - Strong pressure on the learners.
- On-line tutorial / self-organized training
 - Broader contents, less targeted material, larger domains covered.
 - Difficult evaluation
 - Need for self-motivation, autonomy.
- Long trainings
 - Academic environment
 - Fewer differences with school teaching, different goals and materials
 - Wider audience, heterogeneity of previous knowledge.

In the context of the Knowledge Web European network of excellence (KW, 06) the aim is to teach semantic web technologies to potentially interested industrial partners. The different learning scenarios are identified through the learning needs they answer, and according to the following:

- Several actors in a professional context can be distinguished: developers, architects, project officers, managers, experts.
- Various services are envisioned: technical courses, generic introductions, private consulting.

The role of the research community is envisioned under the form of an expert consultation, training or organization of events. The final goal is the adoption of technologies and methods by the industrial world.

2.3.4. Other examples of e-learning scenarios

The following scenarios are precise examples of e-learning applications and contribute to illustrate the context of this work.

2.3.4.1. Course centered scenario

The scenarios introduced by the European project SeLeNe (Rigaux and Spyrtos, 03) deal with the possible uses of a learning system for course consultation. It offers flexibility, distributed access and reuses existing material. Usage scenarios are separated in three categories, depending on the attitude of the user (whether learner or teacher): active, hurried or cooperative.

- The active scenario relies on the hypothesis that the user accesses the system to build his/her “mental model” of the domain. He/she tries to learn on a domain, often pursuing the goal of solving a given problem. The teacher acts as a guide by assembling courses and proposing relevant paths through the pedagogical material.

- The “hurried” user illustrates the case where learners need quick and precise information, to make up for a small precise lack of knowledge, immediately required in the user’s task.

- The last scenario considers teachers as acting in a community. They publish different versions of the course by correcting each other’s proposals for navigating the course.

This illustrates quite simply the possibilities for course management scenarios. For (Allert, 04) this last collaborative scenario is progressively introduced in most applications.

2.3.4.2. Open universities vision

Relying on open universities’ experience (Salmon, 01) describes four scenarios of e-learning usage, with colorful names.

- *Contenteous* is based on content access through technology. The pedagogical model relies on transmission of knowledge from the expert to the learner. Typical activities include lectures and individual reading of course material.
- *Instantia* depicts the "just for me, just in time, just for now and just enough" scenario. Here the individual will use technology (typically a computer and the web) to fill in a punctual and urgent need of training, usually encountered by a subject during his/her professional activity (but private usages can be thought of as well). It presents a classical vision of actual e-learning, provided by on-line tutorials. This is also a first step towards the 'life-long learning' attitude.
- With *Nomadic*, the third scenario concentrates on technological and mobile aspects involved in a learning activity. It envisions that students do not attend classes but take part in a set of activities performed using mobile devices (PDA, cell phones, etc.)
- The last scenario, *Cafelattia*, emphasizes the role of community and the importance of interactions between learners. Learning is favored by dialogue and exchanges inside the "virtual communities" that emerge from communications over the network. The teacher, acting as a group animator, plays successively the role of expert, motivator, or moderator in the group.

2.3.4.3. Adaptive and intelligent systems on the web

From a more technical point of view, (Brusilovsky, 03) lists the different services offered by the introduction of digital technologies in the learning activity. The focus is put on the realization of "Adaptive and Intelligent Web Educational Systems". Five categories are proposed to classify the different technical scenarios:

- *Personalized navigation presentation*: in this scenario the student takes an on-line course where the content is adapted depending on the user. Typically, this relies on a student profile, used to adapt content according to the objectives, the prerequisites, etc.
- *Personalized information retrieval*: in the case of a less constrained learning scenario, searching for information constitutes the main activity. Search criteria are based on content analysis and/or on collaborative information (rating systems for example).
- *Collaborative intelligent learning*: at first, the collaborative aspect may consist of favoring the formation of groups and identifying the different actors and their competencies. Specific "patterns of collaboration" might be revealed. "Virtual students" assigned to specific tasks may collaborate with real learners during their learning process to increase the collaboration aspects.
- *Intelligent monitoring*: a learning system may only be used to observe and monitor students to provide the teacher with the necessary feedback for him/her to adapt his course. The system can work on a voluntary basis (learners provide this information themselves) or through automated procedures, by recording activity traces for example.
- *Intelligent tutors*: assistance is provided to the student along a problem-solving task assigned to him/her.

Existing examples show that a single application may support several of those scenarios (Brusilovsky and Vassileva, 03).

2.3.4.4. Web portals

A last type of scenario concerns the use of web "portals" that offer teaching material. Such scenarios deal with much more independent types of learners, who know what they are looking for in term of learning objectives. Several research applications rely on this scenario where the user connects to a central server and search for learning material through a dedicated service.

Portals might target teachers as well as learners. A portal like UPB/Educanext (Quemada and Simon, 03) for example, proposes pedagogical material as well as activities (video-conferencing, on-line tutorials etc.) stored in a repository. This is not so much a learning scenario than an information retrieval one. Learning will occur when using the resource and not when finding it. On the web, portals exist for many different types of content. The resource they give access to, and the way resources are described and searched can be used to differentiate them.

2.4. Research Questions

The above scenarios illustrate the wide range of situations covered by e-learning and. They also show that many directions have already been well investigated. Our study cannot obviously cover all the possible scenarios, but we shall address several issues, common to many scenarios. To demonstrate the coherence of knowledge management techniques, semantic web and educational considerations, we examine two practical problems containing both human and technological aspects:

2.4.1. Formalized knowledge in the learning context

All the scenarios presented above involve the use of knowledge directly by a computer program. The expressed knowledge may concern the user, the course contents, the pedagogical approach, etc. Then, problems like *knowledge representation, capitalization and extraction* are crucial regarding the realization of the above scenarios.

Knowledge management addresses these issues (Dieng-Kuntz et al., 05) and we need to ask the question of the application of existing methods to the e-learning context. In this scope, we can consider e-learning as a new scenario for knowledge management and ask the question of its realization in practice.

In particular, many scenarios mention the use of semantic representations using tools such as ontologies. We want to evaluate and give practical answers on the proposal of using such semantics in a learning context.

The growing interest in both academic and industrial domains for the presented scenarios raises questions about the implementation, the generalization or the reutilization of tools and pedagogical material. They imply the use of dedicated *technologies, methodologies and tools*. Then we will also investigate the generic issue of reusability in this context.

2.4.2. Learning with the semantic web

Switching perspective, we observe that the web is now an indistinct part of computer systems. For example some scenarios mostly rely on free search for learning material on the web. In this scope the previous issues have to be considered in a “web” environment”. This leads us to consider the semantic web proposal in an e-learning context.

One of the semantic web goals is indeed to facilitate activities like searching and exchange of knowledge. Thus, issues about information retrieval (IR) and formalized knowledge interoperability are part of the global context of this study. The “*webization*” of learning as we like to call it makes this issue interesting for both the e-learning and the semantic web domain.

As a start to take advantage of semantic web techniques for e-learning, we consider that models, tools and methodologies relying on artificial intelligence techniques for e-learning

can be generalized using the semantic web as a generic framework. We will try to demonstrate this integration and to point out its pitfalls.

Finally we think that specific answers must explain how knowledge will be **effectively used**. By clearly favoring a pragmatic approach, we take advantage of past field experience in e-learning and of more experimental work of “intelligent” educational research, to unite them under the technological paradigm of the semantic web.

3. LITERATURE REVIEW	21
3.1. <i>Learning with computers, Computer Assisted Learning</i>	22
3.1.1. Learning theories	22
3.1.2. The Learning Object approach	25
3.1.3. E-learning Standards	28
3.1.4. User Modeling	31
3.1.5. Learning Objects and pedagogy	33
3.1.6. E-learning systems	34
3.1.7. Intelligent tutors and adaptive systems	35
3.2. <i>Knowledge Management and E-learning</i>	41
3.2.1. Organizational memory: inspiration and analogies	41
3.2.2. Ontologies	43
3.2.3. Annotations	51
3.3. <i>Semantic Web</i>	52
3.3.1. The vision	52
3.3.2. Standards	53
3.3.3. Architectures for the semantic web	56
3.3.4. Tools for the semantic web	56
3.4. <i>Semantic web for e-learning</i>	60
3.4.1. Visions	60
3.4.2. Inspiring projects	61
3.5. <i>Conclusion</i>	63
3.5.1. Open Vs Closed worlds	63
3.5.2. Standards	63
3.5.3. Reusing resources	64

3. LITERATURE REVIEW

Given the growing importance of e-learning and the cross domain approach we adopt, the number of research contributions that inspired us or to which our contribution can be compared is quite large. The number of teams spread all around the world, and mentioned in the following, illustrates the extent of this research area. Presenting a complete literature review is not realistic in this situation. We apologize for those, whose important contribution is not cited here, but time and space constraints have already been stretched for this part

The difficult selection of the referenced work is based either on the popularity (determined by the frequency of citations in the overall selected literature), or on the proximity of the subject with our own contribution presented in the following chapters. As the technologies and usage of the web evolve very quickly, we also tried to select the most recent works at the time of writing.

Some sections address rather technical matters. However, for its largest part, this chapter only requires a small knowledge of e-learning situations. Otherwise, the reader is advised to take a step back and reflect on his/her own experience as a learner.

The following presentation is divided in four main areas. First, the three inspiring established domains: E-learning, Knowledge Management connected with AI-ED and Semantic Web, are presented separately. The rigorous reader will point out immediately the many connections between those sections, and we shall explain some them on the way. The last section presents projects or contributions that explicitly build on the crossover between domains.

3.1. Learning with computers, Computer Assisted Learning

We examine in this section the current research in the uttermost active field of human learning whenever computers are involved.

3.1.1. Learning theories

3.1.1.1. Behaviorism/cognitivism/constructivism

A short wander in psychology is necessary to understand the roots and fundamental inspiration of theories of learning that we will mention afterwards. Even if over simplified and often used in unsuitable contexts, the ideas of behaviorism, cognitivism and constructivism are often presented as a reference to identify learning strategies (Dalgarno, 96):

- The **behaviorism** is based on the premise that the working of the brain is out of reach and we must focus on the external response to stimuli. Typically, strategies based on repetition, or ‘parrot’ behavior, are described as behaviorists.
- The **cognitive** vision rather surmises that mental models and cognitive states decide upon a person’s response to stimuli. To simplify, the whole problem of learning is to get the right model into learners’ head. We shall discuss further the concept of “right” model when we get to the philosophical foundations of Ontology (see 3.2.2).
- Finally, in the **constructivist** vision each person builds his own mental model from his/her personal experience. This is often related to the work of Piaget, but it must be noticed that Piaget worked with very young children, not exactly what e-learning often targets. In this scope, the socio-constructivist theory, normally attributed to Vigotsky describes the role played by interactions with other people.

Many specific pedagogical theories, linked to constructivism, have been proposed and are placed on the triangle below (see figure 4) (Dalgarno, 96). The endogenous, exogenous and dialectic axes express the part of individual knowledge construction, formal instruction and collaboration in the learning process.

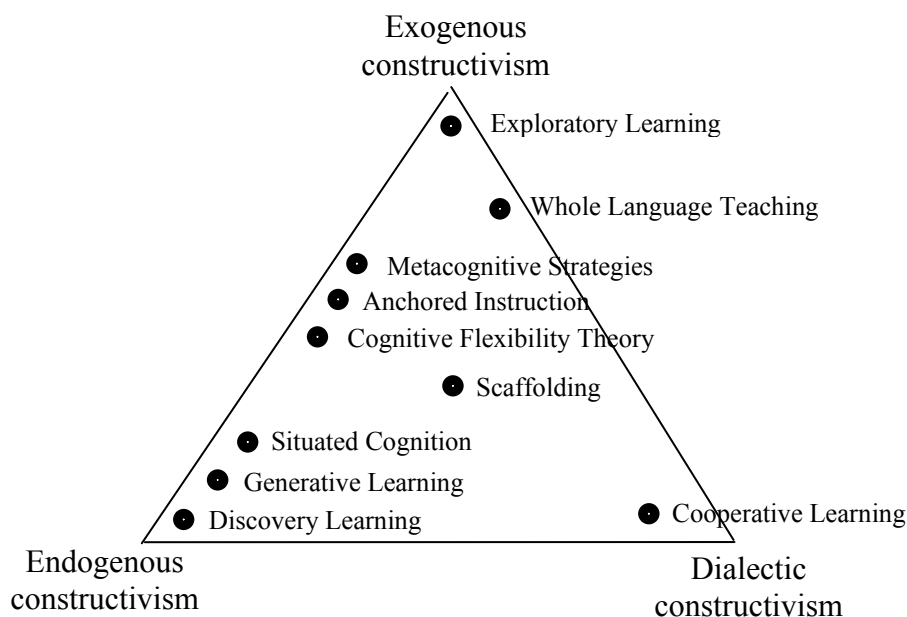


Figure 4 - Constructivist pedagogical theories (Dalgarno, 96)

A different perspective is proposed by (Allert, 04), where the approaches of learning and instruction are presented on a continuum of contextualization. Learning is defined using metaphors: the acquisition metaphor and the knowledge-creation metaphor.

- The acquisition metaphor is close to behaviorist models, as also pointed out by (Schneider et al., 03) where the context should not play an important role.
- Situated approaches of the knowledge creation metaphor are closer to socio-constructivism, and situated cognition. Learning results from knowledge construction from experiences.
- Humanistic approaches place learning at the centre of contextualization.

This categorization highlights the role of peers and collaboration. Such vision is well adapted for its generalization to other domains than academic teaching. For example, it is relevant for knowledge management and the capitalization of variously contextualized knowledge.

Figure 5 illustrates this perspective.

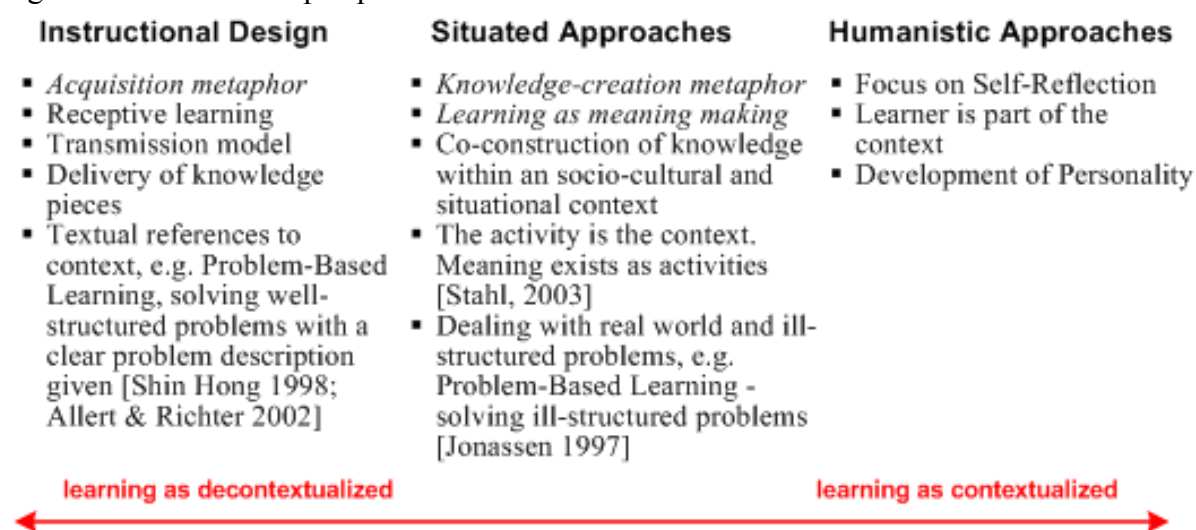


Figure 5 - Approaches of learning and instruction on a continuum of contextualization (Allert, 04)

3.1.1.2. Instructional Design

The term “instructional design” mentioned on figure 5, identifies the effort made by teachers to plan and organize learning. Its place in the “contextualized” part can be discussed. For (Reigeluth, 99) instructional design is not only concerned with selecting the knowledge to be learned but also with the way it should be acquired.

Presenting learners with the exact content they need is crucial and remains a major research issue for computer science. However, the problem of how instruction is designed is equally important. Especially with the freedom brought by the web, the whole range of the contextualization continuum presented above can be addressed.

To define learning designs or strategies, so called “instructional designers” like to rely on theoretical models. The Bloom’s taxonomy, presented table 3, was originally defined as a mean of expressing qualitatively different kinds of thinking (Bloom, 56). It serves as reference model to identify learning objectives (Dalgarno, 98). Recently, it has been adapted to more actual teaching practices with (Anderson, 01).

Table 3 - Bloom's taxonomy of learning objectives

Competence	Skills Demonstrated
Knowledge	<ul style="list-style-type: none"> * observation and recall of information * knowledge of dates, events, places * knowledge of major ideas * mastery of subject matter * Question Cues: list, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
Comprehension	<ul style="list-style-type: none"> * understanding information * grasp meaning * translate knowledge into new context * interpret facts, compare, contrast * order, group, infer causes * predict consequences * Question Cues: summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
Application	<ul style="list-style-type: none"> * use information * use methods, concepts, theories in new situations * solve problems using required skills or knowledge * Questions Cues: apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover
Analysis	<ul style="list-style-type: none"> * seeing patterns * organization of parts * recognition of hidden meanings * identification of components * Question Cues: analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer
Synthesis	<ul style="list-style-type: none"> * use old ideas to create new ones * generalize from given facts * relate knowledge from several areas * predict, draw conclusions * Question Cues: combine, integrate, modify, rearrange, substitute, plan, create, design, invent, what if?, compose, formulate, prepare, generalize, rewrite
Evaluation	<ul style="list-style-type: none"> * compare and discriminate between ideas * assess value of theories, presentations * make choices based on reasoned argument * verify value of evidence * recognize subjectivity * Question Cues: assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize

3.1.2. The Learning Object approach

When working with a computer, learners will manipulate digital artifacts to perform the learning activity they have been assigned to. Such activity can be as simple as reading course on web pages, or very complex like using simulation programs in group interactions. The artifacts involved in the learning process are of prime importance.

For teachers, or instructional designers, it may reveal difficult and time consuming to create this material themselves. They may lack the competencies to master the complex editing tools. Efficiently designing for a computer screen interface also needs training and the specificities of the computer interaction imposes to be more careful regarding the coherence of the course. Studies show that it takes more time to author digital courses for the web than write paper handouts (Christiansen and Anderson, 04). Just displaying static material on screens brings no benefits and even makes it harder to read. Thus, the use of digital displays imposes to construct richer documents like hypertexts. Such documents possess a richer structure than linear books, and their content is suited for computer interaction. Inevitably, the associated creation cost increases.

However, digital content offers a scale economy as its replication is unlimited and does not generate any overhead. With the pervasive spread of the web, the idea has emerged to capitalize this work by exchanging/trading pedagogical material. The “learning object” era had begun (Wiley, 00).

Advantages of this approach are first appealing: exchanging courses over the network should bring not only huge cost savings, but also allows an agent to combine multiple objects and compose personal lessons for an individual learner. The first image used was the Lego brick, illustrating the possible “construction” of the courses like the assembly of a house using the famous plastic brick game.

The LTSC (Learning Technology Standards Committee) from IEEE proposes the following definition of Learning Object:

"Learning Objects are defined here as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning. Examples of technology-supported learning include computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. Examples of Learning Objects include multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning".

(LOM, 06)

This definition is very broad and encompasses any kind of object, virtual or not, digital or not. (Wiley, 00) brings some interesting restrictions, rejecting any non-digital object. He also imposes that the object has some use in the education process. Any object cannot be qualified of “learning” regardless of its role. Most of the other definitions of LO refer to Wiley. Concerning the term “Learning Object” itself, it is the subject of much debate and the following list of rough synonyms exist in the literature:

- "knowledge object", "Components of instruction" (Merrill, 99)
- "pedagogical documents" (Duval et al., 01)
- "educational software components" (Escot, 06)
- "online learning materials" (Merlot, 06)
- "teaching material", "learning items" (Brusilovsky, 03)
- "instructional primitives" (Brusilovsky and Vassileva, 03)
- "resources"

All those terms designate digital artifacts used in learning systems with a similar philosophy as for learning objects. If the original idea appears simple and seducing, major difficulties lie beneath the surface. This vision implies the following three characteristics:

- (1) The LO paradigm is based on the **exchange of information**. Resources must be sufficiently described by metadata to be efficiently searched and exchanged. Text

based search, applied in today search engines, fails in this case because information, like targeted audience for example, cannot be determined by looking in the content.

- (2) Resources must be cut down into right size components. If too large, they may cover too many topics that do not suit the “re-user” need. If too small, they may not contain enough information. The **granularity** is a crucial aspect that must be defined and adjusted to achieve the vision.
- (3) Finally there is no guarantee that content will be **interoperable**. But whether on the technical level or on the content level, interoperating heterogeneous material is required.

We discuss the above requirements in the next three sections

3.1.2.1. Exchanging Learning Objects

The first interest of learning objects is the possibility to exchange them. Large projects usually propose this by setting up exchange platforms called Learning Object Repositories (LORs). Major examples are the ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe) (Ariadne, 06) (Duval et al., 01) and Merlot (Merlot, 06) in the United States. A great number of other repositories are available and may be specialized by domain, by targeted audience, etc... Here is a list of other repositories: Canada School Net, CAREO, EdNa, ETB, GEM, Learning Matrix, SMETE and UBP.

Among those the Universal Brokerage Platform (Simon and Quemada, 02) is a portal that focused on providing a detailed life cycle of a LO, especially regarding the economic modalities of the resource diffusion. In many other portals, this aspect is occulted while it may reveal important to develop a real economy of LOs. Otherwise, all the reviewed portals share the same set of basic functionalities (search, access, etc.).

Efficiently searching LO in large bases supposes the use of “metadata” (literally “data about data”) and exchange between platforms is linked to the use of a common representation for such data. Most of the portals mentioned above rely on a similar metadata scheme, based on LOM or its subsets (see 3.1.3.1). However, describing the objects through metadata is a difficult problem:

- On the one hand, specifying the title of a document for example is simple, but the value of this information in term of automatic processing of the document and inclusion in a course is quite low.
- On the other hand, pedagogically related information is very much context dependant and impossible to express at a generic level.

Manually producing interesting semantic metadata, which are coherent throughout a portal and sufficiently informative for every possible usage, is a noble but unrealistic goal. Experiments on manual filling of metadata clearly show this difficulty (Friesen, 04). Automatic procedures exist but they can only fill in “simple” and “low added value” fields (e.g. date, author, title, etc.).

3.1.2.2. Modularization and granularity

From one application to another, the size of the course smallest entity can vary a lot. Classical aggregations correspond to chapters or modules, down to paragraphs and images for fine grain Learning Objects. (Wiley, 00) proposes an «atomic» vision to illustrate the problem of granularity and contextualization. The “right” granularity for an object is when it is big enough to determine its context of use, and small enough for this context to encompass several use cases. A learning object is also defined as a “minimal unit of pedagogically

reasonable learning content" (Schmidt and Winterhalter, 04). The imprecision of such definitions shows the importance of context in determining granularity.

LOs are supposed to be small parts of courses that may be assembled together. In reality, a complete course is "sliced" to create several LOs that can be composed together later on. The "independence" of each object is relative, and the granularity depends on the context of reuse of the whole course.

3.1.2.3. Combination and interoperability

Learning Objects are supposed to be reusable in different contexts. This idea of combining various bits of material to combine a new course is emblematic of the LO approach. However if the contents of the objects are not coherent with each other, the objects cannot be combined. In the TRIAL SOLUTION project for example (Buffa et al., 05), entire books of mathematic courses are sliced down, but their recombination is limited to the resources coming from a single book to ensure their coherence.

In the original proposal, LOs must be designed to be as generic as possible. However, there is a paradox because the economical argument in favor of LO supposes the reuse of existing objects authored in a specific context. The more the LO is decontextualized, the higher the cost of creating an object is, knowing that objects cannot be completely decontextualized.

3.1.2.4. Conclusion

To sum up the Learning Object approach is quite unrealistic in its original vision. Problems with description, modularization and heterogeneity have been largely underestimated. It is not just a problem of how learning is perceived, as proposed in (Wiley, 03) but a real impracticality. The study proposed in (Christiansen and Anderson, 04) shows the difficulties suffered by the LO approach (lack of context, heterogeneity, difficulty to search, etc.).

The Learning Object approach cannot be considered as a revolution in the practices, but more as a general philosophy of managing learning material. In most cases learning material is already not monolithic, and reorganization inside closed coherent sets of course contents has been experienced (Buffa et al., 05)(Farell et al., 04)(Rigaux and Spyrtatos, 03). The real impact of this approach depends on its practical modalities. Real world applications apply the LO paradigm but they narrow the original hypothesis: by reducing the set of potential objects, by sticking to specific formats or by fixing the scope of use of the potential LOs.

In sections 3.1.6 and 3.1.7, we present examples of learning systems integrating by the Learning Object paradigm in their philosophy. This includes largely used Learning Management Systems (LMS) and Learning Content Management Systems (LCMS), as well as more research-oriented applications with examples of Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia (AH).

3.1.3. E-learning Standards

Despite the impracticalities of the LO approach, still not clearly agreed upon, the industrial community pushed standardization efforts in that direction. The standardization process is long and presents many non-technical aspects, including political or influential ones that are not interesting here. History and origins of the various, often competing, standardization bodies will be let aside to focus on the content of most prominent standards, directly of interest to our subject.

3.1.3.1. Metadata: LOM

Metadata instances for learning objects must describe relevant characteristics of the learning object to which it applies. An international standard LOM (Learning Object Metadata) has been developed in this scope. Its purpose is to facilitate search, evaluation, acquisition and use of LOs. Learners, instructors or software should use this description indifferently. It also targets the sharing and exchange of LO through catalogs and inventories.

The LOM groups the characteristics of Learning Objects into nine categories: general, life-cycle, meta-metadata, technical, educational, rights, relations, annotation, classification. More than 80 metadata fields, placed under those categories, compose the standard. The categories life-cycle, meta-metadata, technical and rights do not target learning specifically. The other categories present some information specific for learning. Value spaces for each metadata field are specified: either free text (strings) or specific values from a vocabulary. The use of the vocabularies entry is not compulsory but of course recommended to ensure interoperability. For (Duval et al., 01) controlled vocabularies improve the precision of descriptors and used terms should be imported from existing domain specific thesauri, like Dewey Decimal classification for example.

The standard opens many interesting opportunities for semantic description of learning objects. However, the specified vocabularies are very limited (few entries for each field):

- Some of them are too abstract like General.aggregationLevel that takes integer values between 1 and 4 to represent the granularity of the object. As we have pointed above the granularity is a crucial problem in the Learning Object paradigm, and such value spaces are not likely to solve it.
- Other vocabularies are also not consistent like Educational.LearningResourceType, which mixes entries like “exercise”, an abstract concept, with “slide” designing a physical object. Usage experiments (Friesen, 04) showed the difficulty faced by practitioners to fill in this metadata using the provided vocabularies.

Only pure objective attributes (title, author) are easily filled in. Technical ones are not meaningful for pedagogical concerns. Pedagogical ones, representing the real benefit for practitioners, are not used and judged too difficult to fill in. As the standard builds on Dublin Core (DC, 06), most of the effectively used attributes are in fact contained in this smaller generic standard.

The poor usability of the standard however does not suppress all its interest. As we shall see in the next sections, most of the identified metadata fields are useful and incorporated in advanced work on learning systems, but with richer and more specific vocabularies. There is no formalism directly associated with the standard, but it proposes an XML binding. An RDF version has also been submitted (Nilsson et al., 03).

3.1.3.2. Packaging : SCORM

The largest effort of standardization supporting the LO approach is the SCORM standard from ADL (SCORM, 06). SCORM stands for Sharable Content Object Reference Model and has reached version 2004. This standard is three fold:

- The CAM (Content Aggregation Model) defines a set of information used to describe LO and their aggregations (SCO or Shareable Course Objects). For instance, it includes the LOM standard. It also specifies how the objects must be stored physically in files and how the descriptions of the LO must be organized and expressed in XML. The specified “format” intends to be generic and the aggregated course can be used in any compliant LMS (Learning Management System).

- The RTE (Run Time Environment): describes how the LMS should interpret the metadata contained in the SCO. An API is proposed to “run” SCOs
- -SN (Sequencing and Navigation): specifies how to organize navigation among the various objects.

SCORM is based on the collaboration of major standardization bodies in the field of E-learning: ARIADNE, IMS, IEEE, AICC as well as industrial partners: IBM, Microsoft and American military. It is an effective industrial standard for LMS applications and most commercial platform accept content in SCORM format. From a research point of view, this standard has little impact. It mainly provides technical means of interoperability but semantic aspects are limited to the LOM and a rather classic content model. Sequencing and “intelligent” functionalities are poorly supported by hand-written scripts.

3.1.3.3. *Scenarization: LD*

A standard still subject of much debate is the IMS Learning Design. The original aim of this standard was to provide an EML (Educational Modeling Language) (Koper, 01) to describe learning strategies and scenarios. Instead of describing a succession of resources involved in the learning task and guiding the learner along that path, it relies on the idea that it might be more efficient to express directly the pedagogical design in a formalized way. Multiple outcomes are expected:

- Learner always performs activities. They are necessary and a content oriented approach does not always provide direct ways to enforce them, LD would.
- It gives practitioners a single language to talk about learning scenarios.
- Machines could automate the execution of these scenarios.

Following the LO approach, learning designs are seen as independent parts of a complete course activity. Thus, the course chunks or ‘units of study’ could also be exchanged and reused like LOs, with potentially the same problems.

Three models form the LD specification:

- A conceptual model defines the different concepts and their relations.
- An information model describes the attributes each instance of concept can have.
- A behavioral model indicates how programs should interpret learning designs and “run” them.

In parallel, the standard distinguishes in three levels of complexity:

- Level A only contains the basic concepts to describe learning designs. An activity is modeled by the following main abstracts concepts: roles, activities themselves, methods, etc.
- Level B adds “properties” that can be interpreted as rules associated with a set of information about the user (user model), the course, the environment, etc. For example, the element “when-property-value-is-set” allows a program to sequence the reading of resources depending on dynamical data. This information is obtained through rules like “on-completion” and “change-property-value” or “condition”. The standard also specifies the boolean expression language for the rules.
- The C level models dynamical reactions of the system (not based on static rules). The goal is to generate active reactions of the system depending on triggering events. The specification only describes mail sending, but many others reactions are possible.

LD is said to be pedagogically neutral. It should support behaviorist, cognitivist and constructivist learning (Tattersall and Koper, 03). The apparent power and appeal of the language must be balanced by the lack of existing use cases. If seducing, the LD standard is

mostly empty and does not model deep mechanisms that would enable real machine processing. To our knowledge, there is no convincing example of learning systems taking advantage of the LD specification as intended. For example the LAMS framework only allows linear sequencing of activities (Dalziel, 03), which presents little interest. More advanced systems are currently developed but without convincing results up to now.

Yet despite this poor impact on practice, from a conceptual point of view, and as for LOM, the concepts identified are crucial and their definition is useful to start modeling and representing learning designs. Other EML, like PALO (Rodriguez and Verdejo, 04), have been proposed but they were not picked up to become the standard. By choosing a specific pedagogy (ex: CSCL, Problem based learning) they loose the generality of LD. However, by defining more practical and grounded aspects, they can lead to effective implementations. This has not been demonstrated yet for LD.

3.1.3.4. Other standards: personalization, test, competencies etc.

Other relevant standards mainly deal with user representation or learner modeling. No unique standard for e-learning as emerged yet, so we quickly indicate the following ones:

- First, vCard (vcard, 01) is one of the main standards for user identification. It contains basic identification data (name, address, etc.).
- Foaf (Foaf, 06) is an initiative related to the semantic web (see 3.3) and mainly describes relationships between persons. It belongs to the “micro-formats” standards, which are simple and small descriptions of major relations and concepts for a specific purpose (rights, persons, etc.).
- The IMS consortium has also issued a series of standards for users (LIP), questionnaires (QTI), competencies (RDCEO). For a complete list of the standards proposed, see (IMS, 06). The information they express remains at a low level and they lack common value spaces and vocabularies to envision interoperability.

Modeling the user, his/her characteristics, profile and knowledge is a very complex task. Advanced research is currently conducted in the cognitive psychology domain. Practical applications of those to e-learning are still a long term objective.

The most prominent contribution come from the domain of user modeling and we describe classical models for user modeling below (see 3.1.4).

Globally the standardization process in the e-learning field has not been as far as expected at first. It took directions that were more economically motivated than thoroughly demonstrated by experiments. If actual “commercials” systems rely on those standards, the benefits are still to be demonstrated.

3.1.4. User Modeling

Adaptation to the learner is often presented as the main motivation for using e-learning systems. It is also described as one of its best advantage over classical learning settings. A personalized approach takes place in a one-to-one relation between student and tutor. Unless there is one teacher per student, this can only be achieved using automatic, electronic tools. Computers offer a great deal of possibilities to record user actions and progress. Such data can be collected in an intrusive manner, through questionnaires or dialog boxes, in which case the feedback information is directly given by the subject, or in an un-intrusive way with loggers, eye trackers, etc. in which case an interpretation phase is necessary to determine the relevant emerging characteristics.

Several broad categories of models exist:

- Some models aim at representing the state of knowledge of the learner.
- Others try to classify learners into predefined categories.
- Finally, some try to represent and explain the thinking processes.

3.1.4.1. Overlay model

The overlay model is a classic representation of user knowledge. It is based on the idea that the knowledge to be acquired is represented in a separate model (graph, ontology, etc.). A user's knowledge can then be reported according to this model by assigning weights to its different elements. This "layer" of weights explains the name "overlay". Weights can be of various nature: percentage (de Bra et al., 03), probability (Henze and Nejd, 01) or atomic values (Bouzeghoub et al., 05). Figure 6 presents an abstract example of overlay model.

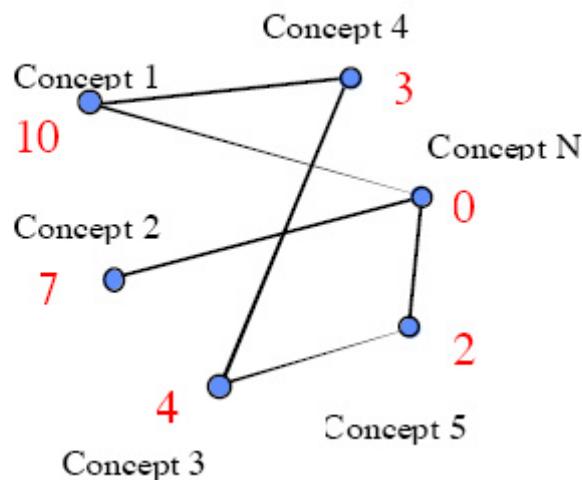


Figure 6 – Example of an overlay model (Brusilovsky, 03)

Knowledge expressed in domain models can be procedural knowledge, in which case it is easier to determine the knowledge level for each learner through simple exercises and check the result of the exercises automatically. In the Aleks system (Falmagne et al., 00), available as a commercial product, domain knowledge in mathematics is represented by the problems that can be asked to students. For example, a set of algebra problems describes the domain of algebra. The user knowledge of the domain is measured relatively to the number of problems he/she can solve. Precedence relations authored manually, structure the domain. Using them, a program can build a personalized path among the instances of the problems classes. Figure 7 shows a visualization of a learner's knowledge state reporting his/her progress on the resolution of exercises.

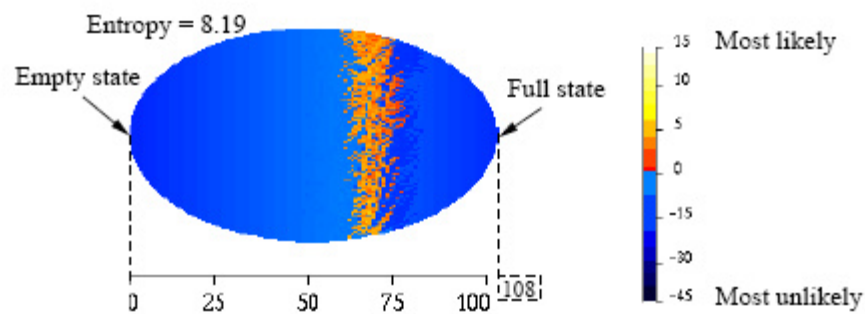


Figure 7 - Representation of a knowledge state in the Aleks system (Falmagne et al., 00)

A problem with overlay model is that they consider learning as a linear process. They do not easily model intermediate states of knowledge, like intermediate conceptualizations.

3.1.4.2. Stereotypes

Another type of user model is stereotypes. Users can be classified into categories depending on their knowledge, ability, cognitive profile, etc. Such models offer rich information for efficiently tailoring material. It is easier for teachers to specify system behaviors with respect to those stereotypes. However, the classification of the users may be somewhat arbitrary. For example, (Brown et al., 05) give the following typical values for stereotypes: holist/analytic, verbalizer/imager, sensing/intuitive. Another example is the framework of (Kolb, 76) that distinguishes four learning styles: Converger, Diverger, Assimilator, and Accomodator. Such stereotypes try to capture the “cognitive preferences” (Dagger, 02).

Another direction for stereotypes models is the “persona” approach where learners play the role of a defined character, close to him/her but with a defined learning style he/she has chosen (Suzuki et al., 98).

3.1.4.3. Predictive Models

The predictive models do not represent the knowledge of the users, but the correct and incorrect procedures he/she is using. The classical example of such model is the “buggy” model for learning subtraction (Burton, 82). Such models are very powerful in understanding user mechanisms of reasoning. However, procedures must be described at the right granularity to match user steps as precisely as possible. The major drawback of this kind of model is the huge work necessary to define the different possible “misconceptions”.

This type of model mostly applies to the acquisition of procedural knowledge. We do not know any example of conceptual knowledge represented by “incorrect” intermediate models, even if in psychology such representations exist.

3.1.5. Learning Objects and pedagogy

Systems supported by standards like LOM and SCORM are said (Allert, 04) to be based on the behaviorist approach and Gagné, and not on the idea of “active” courses despite the fact that new technologies now provide the tools for this type of interaction (e.g. feedback, collaboration , etc.). This is largely questionable as the standards do not specify how to use the objects. The socio-constructivist approach for example presented in (Schneider et al., 03) acknowledges the role of Learning Objects.

The number of theoretical and practical theories for learning is huge. We choose to present three of them that address different paradigms of instruction and acknowledge a LO approach.

- First, we present a socio-constructivist inspired use of computers for learning.
- Then, we adopt a more “applicative” point of view, with the instructional transaction theory.
- Finally, we discuss the problem based learning strategy (PBL), often called upon to enrich e-learning by motivating scenarios of knowledge construction.

3.1.5.1. Socio-constructivist hypothesis

In (Schneider et al., 03) we find an argumentation in favor of Learning Object based systems, and about their correct use in a socio constructivist approach. The introduction of computers in the pedagogical activity is said to emphasize the socio-constructivist model. In this approach, the teacher acts as a guide more than a recipient of knowledge. The instructor/teacher defines a pedagogical workflow.

The learning scenario states the goals, and pedagogically organizes the activity. It precisely defines the role of the instructor/teacher. He/she is a manager, a facilitator and an orchestrator. Hence he/she does not « force » knowledge transfer on students.

The learning Object paradigm suits this learning context, as long as objects are not at the centre of the activity. The question of granularity is met at the scenario level. For each level of detail in the objects, the teacher describes and sequences activities to guide the learner. In this approach, the computer is used for help more than delivery of course material. The construction of knowledge remains personal. It is not “coded” inside the system. Thus, most learning objects problems like coherence, granularity etc do not really matter as LO are not presented in a coherent set.

3.1.5.2. Instructional Transaction Theory

(Merrill, 99) defines the Instructional Transaction Theory (ITT). This theory deeply connects to computer assisted learning. It is based on instructional theory from Gagné and states that learning outcomes can be obtained if the learner performs the necessary steps. It draws a parallel with computing and assumes that those steps and conditions can be expressed just like computer programs and data. Thirteen “instructional transactions” are defined (among them: identification, execution, interpretation) they are described like algorithms of manipulation of “knowledge objects” (artifacts used for learning). Such objects can be understood as learning object possessing a rich description. This vision proposes a very systematic view of learning, certainly well adapted for learning industrial procedures, but it obviously cannot cover every learning domain.

3.1.5.3. Problem Based learning

In a different perspective, and among the various instructional “designs” also called “strategies”, the Problem Based Learning, or PBL (Savin-Baden, 00), is worth a look at in our context. One of the major problems with computer based instruction is to keep the user’s attention because it is more difficult to focus on a computer screen than on a living teacher. Users must be active, and one of the classic ways to keep someone active is to enroll him/her into solving problems, or asking him/her questions. In this scope (Barrows, 86) proposes a taxonomy of problem based learning methods. Using PBL forces students to concentrate and use the information they have read or heard. Applying that information then contributes to transform it into procedural knowledge. This is another constructivist vision that could involve LOs (He, 02).

3.1.6. E-learning systems

After presenting the main standards and some approaches based on the Learning Object approach, we quickly present the main categories and examples of systems where this paradigm may be implemented.

The generic name of Learning Management System describes applications that allow the learner to read and interact with a course. The following more precise denominations exist, each one highlights a specific facet of the system:

- LCMS (Learning Content Management Systems), content oriented.
- ALE (Adaptive Learning Environment), adaptation of the content is provided
- ILE (Intelligent Learning Environment), the systems “intelligently reacts” to user actions
- IES (Intelligent Educational Systems), same
- ILS (Integrated Learning Systems), the system is “pervasive”, it is part of the global learning framework.

- RBLE(Resource Based Learning Environment), rely on content
- WBES (Web Based Educational Systems), offers a web interface.

On these platforms, courses are often composed by aggregation of smaller resources. The major interest is that learner use of the material can be personalized, tracked, etc.

With the integration of the various mechanisms of e-learning (adaptation, tracking, etc) and the convergence towards full web interfaces, the distinction between the above denominations has blurred and the different names are more historically related than linked to the functionalities.

We cite the four following systems as example of standard, largely used LMS. This is not exhaustive but gives an overview of existing emblematic systems. All these systems accept standard SCORM packages as Learning Objects.

- BlackBoard : A commercial platform offering several services that can be combined to build complete learning applications (LMS, CMS, Community support, etc.).
- MOODLE : An Open Source software, offering « course management” for resources. It also integrates communication tools, supports timed quizzes, manage assignment submission, etc.
- WEBCT : now the property of blackboard company, offers a course management system.
- Xtensis : a system that exploits a bit further the LO paradigm, by deploying reasoning mechanism over the pre-requisite links.

In more experimental types of LMS, like ITS (Intelligent Tutoring Systems) or AH (Adaptive Hypermedia) adaptive functionalities are further developed. We present them in the next section

3.1.7. Intelligent tutors and adaptive systems

The introduction of AI technologies in digital learning systems has given birth to the idea of “intelligent” programs that could help learners. Since the beginning of computer assisted learning many of those small highly technical systems have been proposed.

3.1.7.1. Definition

Intelligent tutoring systems

The broad definition of intelligent tutors relies on the basic metaphor of a computer system acting like a human tutor. It brings help and added information to understand a given problem or perform specific exercises. The “intelligent” aspect comes from the highly contextualized feed back given by the system. Double quotes are required here as this term is borrowed from “artificial intelligence” and it could be misleading for none specialists users. This field is inspired from previous research on expert systems for the methods and techniques of managing knowledge.

- “Tutor” means that some kind of interaction must take place. For example, a static on line course cannot be considered as a tutoring system. Interactions must also target learning to justify this name.
- “Intelligent” implies that the reactions of the system are based on external knowledge and a reactive component using this knowledge.

Broadly defined, an intelligent tutoring system is an educational computer program partly relying on an artificial intelligence component. Typically, the program tracks students' work and tailors feedback and hints following his/her actions. By collecting information on a

particular student's performance, the software can make inferences about strengths and weaknesses, and can suggest additional work, readings, or exercises.

According to (Brusilovsky, 99), ITS may offer two broad functionalities:

- Resource selection, or curriculum sequencing, working with the available resources.
- Contextual help, or intelligent analysis of student behaviors to provide interactive problem solving support. Working on a communication level with the learner.

Tutoring systems can interact with the learner through question/answer mechanisms. This is the easiest way to get learner's feedback and to start a dialog with the tutoring program.

Adaptive Hypermedia

When the presence of a tutor is not really emphasized, we rather talk of adaptive systems. An adaptive system changes its behavior depending on the user's interactions with it. Adaptation is expressed through customization of the interface, ranging from coloring links to more complex dynamic activity planning. Adaptive systems are a type of intelligent tutors, but they usually rely on generic adaptive paradigms, sometimes disconnected with learning. For example (Jacquiot et al., 06) relies on situation calculus to determine recommended navigation path in a closed set of documents. Most systems now provide a web based interface using browsers (Mitrovic, 03) and hypertext navigation, which lead to the adoption of the term "Adaptive Hypermedia" (AH).

The mechanisms of adaptive hypermedia involve three types of knowledge (Brusilovsky, 99): knowledge about the domain to learn, about the student and about teaching strategies. These knowledge types will be further discussed in the next section on knowledge management for e-learning.

For adaptive functionalities, adaptive hypertext navigation may rely on the following techniques (Dolog et al., 04):

- *Link annotation*, additional information is given on the suitability of the link in the given context, by using color for example in a traffic light metaphor. Link can even be removed from the document.
- *Link generation* or creation of links using external knowledge and annotations.
- *Resource sorting*, if a link leads to several resources (depending on the interface) they are ordered in an adapted way.

3.1.7.2. Application examples

Intelligent tutor applications and adaptive hypermedia may take various forms. They are best described through illustrative examples. The two systems we present below are quite emblematic of the domain and freely available which motivated our choice. However, they do not present all the possible mechanisms, and we do not intend to be exhaustive in this presentation either. For a more complete list of examples, please refer to (Brusilovsky, 99).

SQLTutor

The SQLTutor (Mitrovic, 03) is a knowledge-based teaching system, which supports students learning SQL. The goal of the system is to adapt to the needs and learning abilities of individual students writing SQL queries. The tailoring of instruction is done in two ways: by adapting the complexity level of problems and by generating informative feedback messages.

The screenshot of SQLTutor, on figure 8, illustrates the various tutoring features:

- (1) The questions is selected by the system according to the user's performance
- (2) Hints are given to the student according to its answers

(3) Students can ask directly for feedback, after selecting the right “feedback level” in the dropdown box.

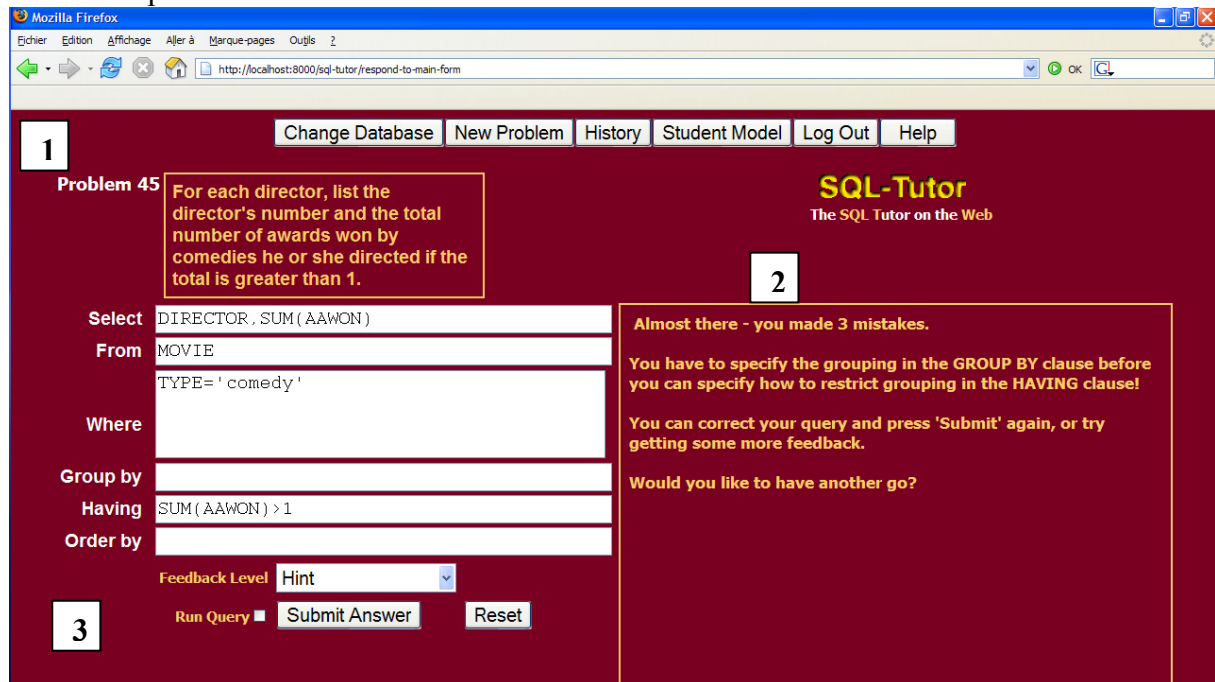


Figure 8 – Screen shot of the sql-tutor web interface

Technically, knowledge about the domain is expressed as a set of constraints in Lisp. An example of such constraint is shown below figure 9 (Mitrovic, 03). This constraint expresses a pedagogical expertise regarding a specific type of error. That is, if the user writes the ANY or ALL predicate in the WHERE clause, the system checks that the corresponding attribute in the SELECT is of the same type.

The feedback is visible on the second line and the conditional expression is detailed in the remaining.

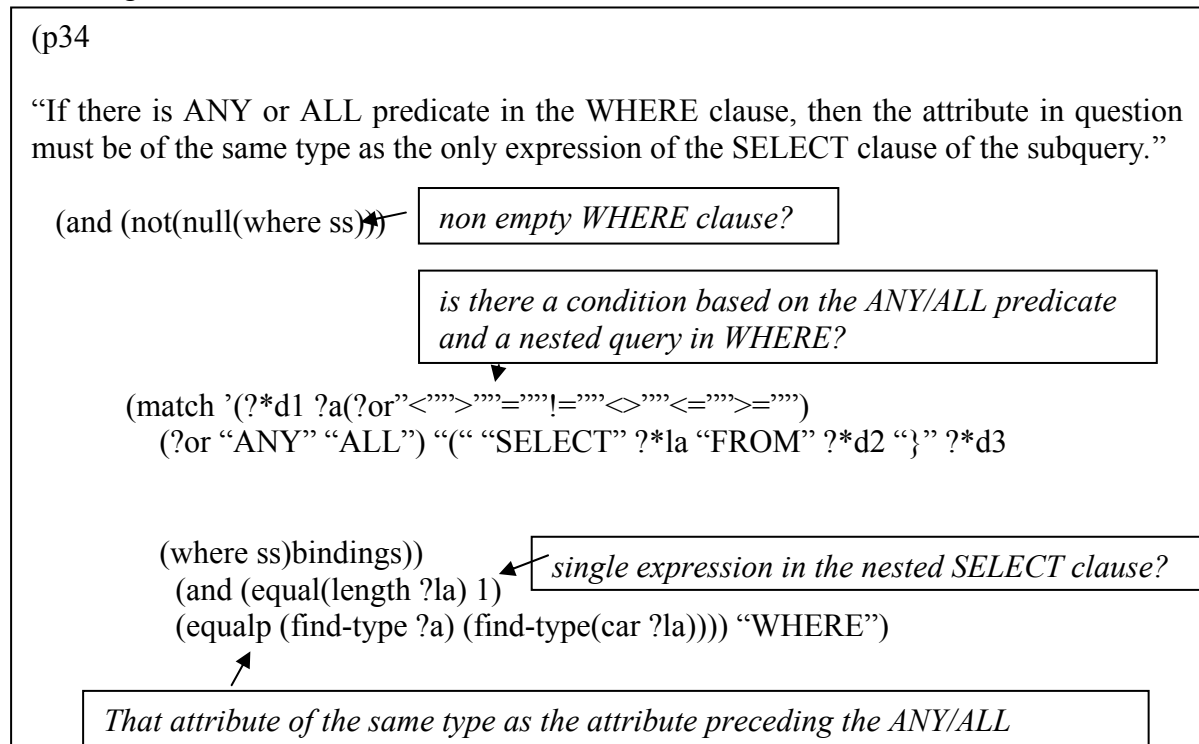


Figure 9 - An example of constraint in SQL-tutor (Mitrovic, 03)

The Adaptive Hypermedia Architecture

To illustrate Adaptive Hypermedia, we selected another system also freely distributed. It belongs to the category of systems, where knowledge and resources are expressed in proprietary XML formalisms. Other similar examples based on standards (HTML, semantic web standards, etc.) will be introduced later (see 3.4.2).

This application is called the Adaptive Hypermedia Architecture. It is presented by its creators (de Bra et al., 03) as a generic adaptation framework. It proposes to base all adaptive systems on the same principles:

- Rules help assign weights to the various resources,
- The visibility of a resource depends on its weight with regard to a threshold.
- User navigation progressively modifies the weights so that new resources become available or hidden.

Availability (or recommendation) of a resource is materialized by colors in a dynamically generated table of content. The classic “traffic-light” metaphor (Dolog et al., 04) with green for recommended and red for not recommended is used. Links in the content of the resources can also be “hidden”, a feature called “link hiding”.

The AHA model (called by its authors AHAM), is based on the definition of concepts. A “concept” may be a document (web pages, or fragment of web page) or an abstract notion. Concepts have user defined attributes that take boolean or integer values (access, knowledge, interest, etc.). Those attributes are used in the expression of rules. The effect of a rule is to update attributes of other concepts and thus decide on the visibility of the concept. The rules are triggered on event-condition-action basis. This means that on each access to a resource, corresponding attributes are updated. This update triggers the rules that refer to those attributes, and so on. The set of concepts is hierarchically structured into the table of content. Additional relations can be defined between concepts that will also affect the evolution of attributes values. Some rules are defined by the system but additional rules can be defined. Each user possesses his/her instance of the conceptual model with specific values for each attribute. It implements a typical example of an overlay model.

To give a complete illustration of the most common adaptation features, we deployed an instance of the AHA system. For this we used the same course, on signal analysis, as the one on which we base our work presented in the following chapters. Two types of adaptation must be distinguished:

- By using direct dependency relationships between documents
- By referring to an abstract conceptual representation of the domain.

In the first case, two types of adaptation may be proposed: the link recommendation and the document customization. It depends on how the course is structured (i.e. a set of independent documents or single document with customizable fragments). In this example, we examine the result of the definition of a dependency relation between the resource giving the definition of a sound card “defCarteSon” and an example of sound card “exCarteSon” in the AHA authoring tool.

Figure 10 shows, on the left pane the structure of the course material and on the right pane the prerequisite relation between the two documents. Each entry on the left can be dragged and included in the graph on the right. Different relations can be defined. In this example we defined a prerequisite relation.

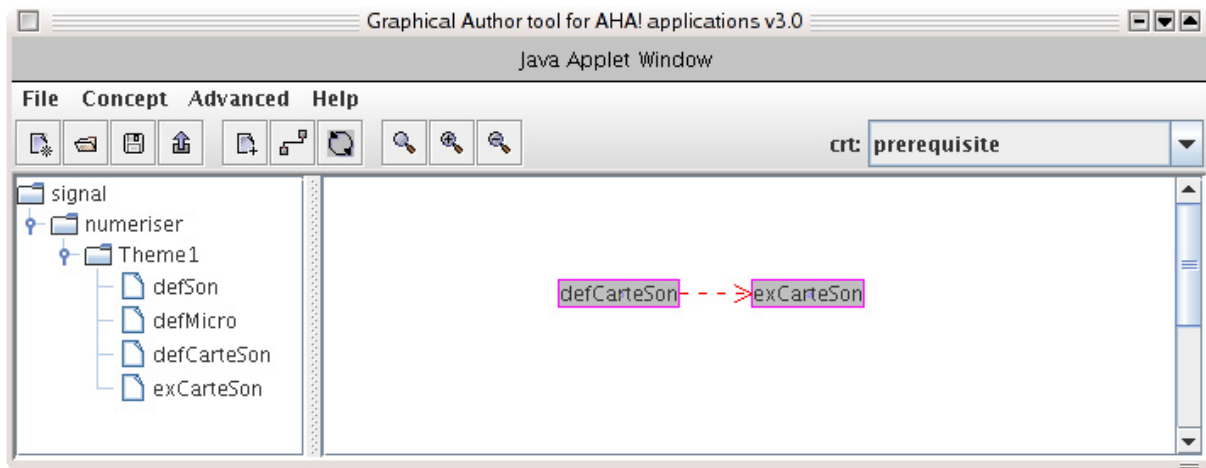


Figure 10 – Prerequisite relationship between two resources in the AHA authoring environment

In the learner interface (see figure 11) the same course structure appears on the left pane with explicit titles. White dots figure visited documents. The green dots show recommended documents and the red dots non recommended documents. The same distinction may apply for the color of the links: blue are recommended, violet are visited, black are not recommended. The right part of the interface shows the documents themselves.

On Figure 11, the red bullet “example” on the left remains red as long as the “definition” above is not visited.

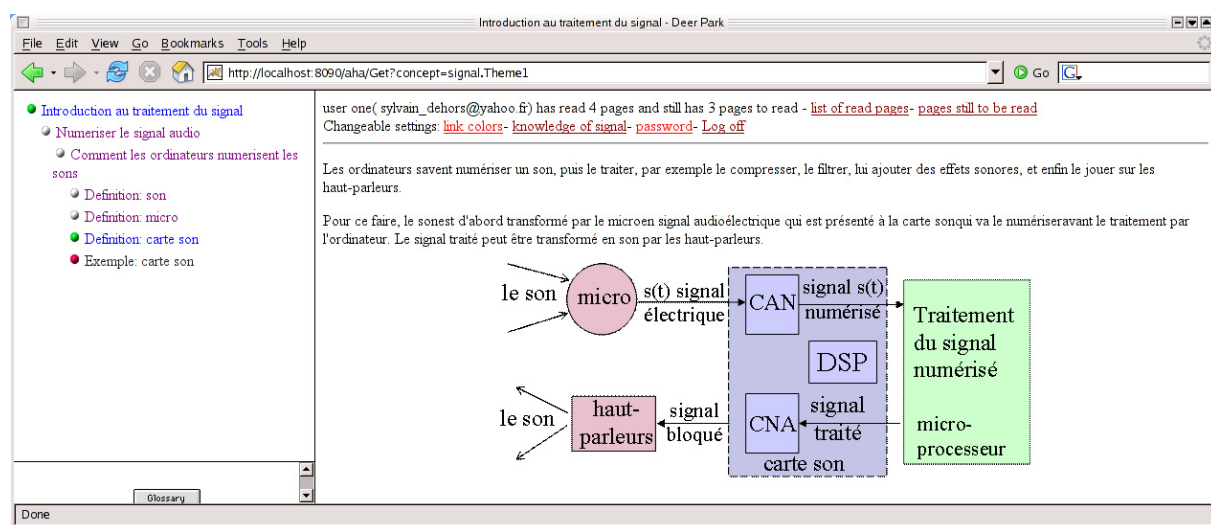


Figure 11 – Link recommendation in the AHA system

Adaptation can also consist in adapting directly the content of the document. For example if the two previous documents are in fact contained in a single document about “carte son” and a dependency relationship requires that knowledge of the document “defSon” is necessary to see the example. Then the content of the documents can be modified to reflect this adaptation as shown on figure 12.

On figure 12, the XML block corresponding to the example is surrounded by a conditional “if” block. The condition for the display of the block is indicated in the “expr” attribute. Here the knowledge attribute of the concept “defSon” must be equal to 100. That means the block will only appear when the “defSon” document has been visited.

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <!DOCTYPE html SYSTEM "/aha/AHAstandard/xhtml-ahaext-1.dtd">
3. <html xmlns="http://www.w3.org/1999/xhtml">
4. <p>Definition</p>
5. <p>La carte son ...</p>
6.
7. <if expr="signal.defSon.knowledge==100">
8. <block>
9. <p>Exemple</p>
10. <p>Sur la carte son ...
11. </p>
12. </block>
13. </if>
14. </html>

```

Figure 12 - Authoring of adapted content in AHA proprietary XML language

The result of the adaptation is displayed on figure 13. The first screenshot shows the document informing about sound card on the first visit. It only contains a definition. On the second visit, visible on the second screen shot, the example with the picture is included in the document.

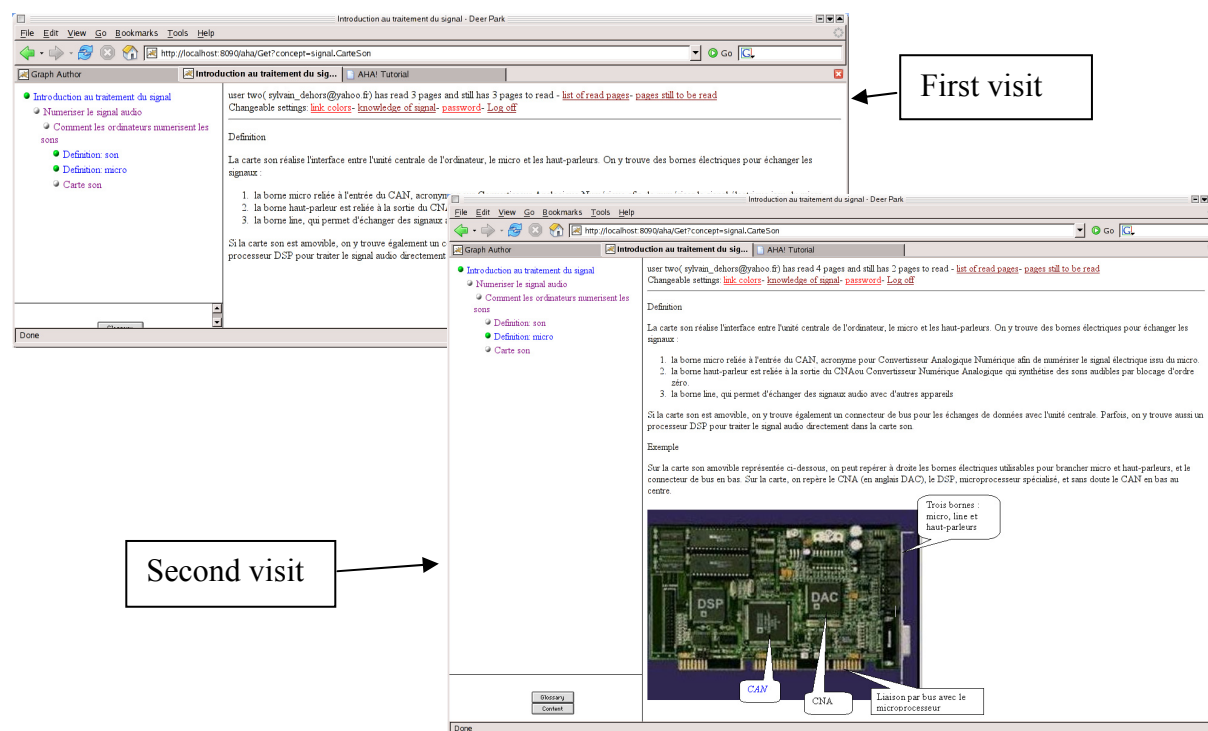


Figure 13 – Example of adaptation of the content

The above adaptations rely on basic dependency relationships between documents or fragments of them. They can be refined by introducing abstract concepts that are themselves linked to documents. In this scope, user-interaction increases the abstract concepts knowledge. Then, this influences documents appearance in the interface. Figure 14 shows how the concept of “son” (sound) that is introduced by the document “defSon” is a prerequisite for the concept of “CarteSon” (sound Card). The latter is introduced by “defCarteSon” and exemplified by “exCarteSon”. Adaptation rules will then guide the user to first discover the concept of sound through its definition before moving on to sound card through its definition and example. Using the abstract concept, we have generalized the dependency relation. The teacher does not have to define the priority of the definition of sound over both the definition and example of sound card. Working on the abstract level reduces the work by factorizing some information.

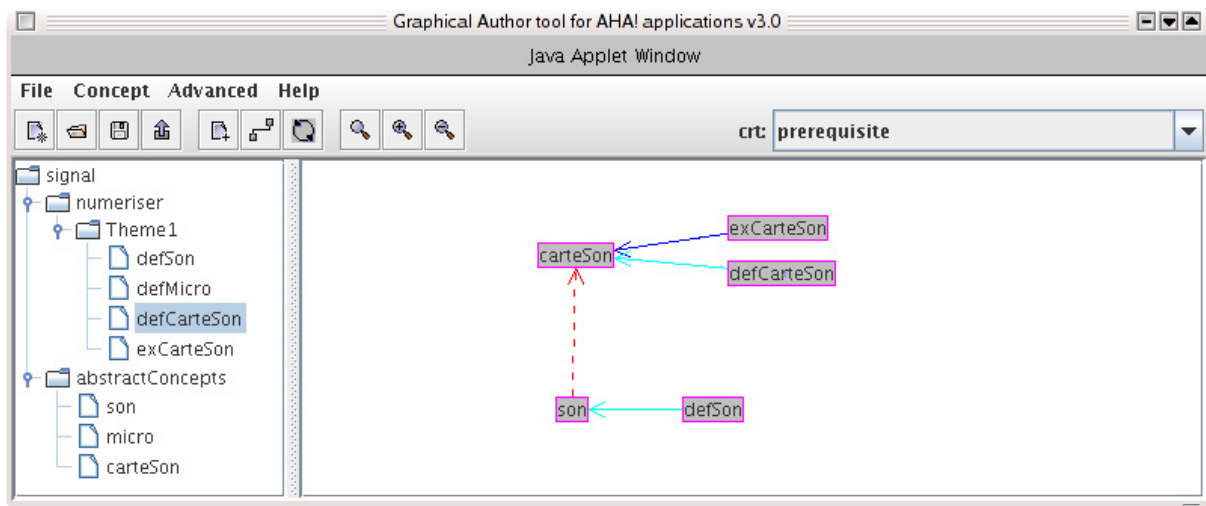


Figure 14 - Use of abstract conceptual representation, using abstract concepts in AHA authoring tool

Other examples of adaptive systems can be found with Interbook (Brusilovsky, 99), WHURLE (Brailsford et al., 02), and AHES (Hanisch et al., 06). We can notice that the last two, which are more recent works, are based on XSLT transformations, a very practical technology to modify and adapt XHTML documents.

One of the main issues about adaptive systems implementing the mechanisms presented above is the huge time required to specify each dependency relation between resources or concepts manually. In existing applications, the pedagogical content, relying on proprietary formats (see figure 12) has to be authored specifically. This requires an expertise not available among the vast majority of teachers. Even if graphical interfaces are provided like in AHA. Such solutions do not scalable for real world courses containing hundreds of resources. Up to now, adaptive systems did not reach a large audience.

3.2. Knowledge Management and E-learning

In this section, we present the potential analogies between knowledge management and e-learning. We particularly focus in the second section on one the main KM tool for e-learning: ontologies.

3.2.1. Organizational memory: inspiration and analogies

3.2.1.1. Organizational memories for learning

We explained in the previous chapter the convergence between e-learning and knowledge management practices (Maurer and Sapeer, 01). An interesting tool of knowledge management is the organizational memory. An organizational memory targets the growth, transmission and conservation of knowledge (Dieng-Kuntz, 04) often in a corporate context. This applies for example to the memory of industrial projects, the knowledge of a company, etc. This knowledge can be theoretical or practical. For companies we talk of “corporate memory”. They are helpful to improve the organization performance and help building on previous experiences. The term of “organizational memory” can be used for more informal communities or legal structures different from companies.

A classical academic teaching organization, such as a university or a school is one type of organization that may need to build an organizational memory. Distance education providers, like open universities, are often using such tool: Considering the scattered localization of its students and staff, standard solutions such as physical libraries and files are not suitable.

At a smaller scale, a pedagogical team of teachers might also need a memory to keep track of previous courses, assignments, etc. to improve teaching from one year to another. Such use of organizational memory for e-learning was introduced by (Abel et al., 04) for example.

3.2.1.2. Corporate semantic web

An analogy with the open World Wide Web can be build to propose a “corporate semantic web approach” (Dieng-Kuntz, 04). This approach deals with heterogeneous and distributed information like the semantic web (see 3.3) and tries to solve the problem of information retrieval relevance. In contrast, however, a corporate memory has a context, an infrastructure and a scope limited to the organization that allows semantic web techniques to be applied with greater impact. The resources of the organization can then be searched and retrieved based on knowledge representations.

Starting from this result, the same analogy can be made between the web and the “corporate semantic web” to jump from “educational memories” to “educational semantic web”. In the following, we shall describe organizational memories for education as the connection between three components:

- *Documents* or the expression of the educational material. Examples may be slide shows, assignment sheets, books, etc. Documents may contain interactive content (e.g. flash animation).
- *Ontologies*, a concept explained in detailed below, express the conceptual knowledge necessary for structuring the memory. For example, knowledge may express that “field” in the JAVA programming language are specific types of “variables”.
- *Annotations* express a different kind of knowledge. They link documents with concepts of the ontology or instantiate ontological concepts to formalize knowledge about the documents.

This generic, three poles, overview (see figure 15) is the fundamental organization for knowledge on which we base our understanding of knowledge management techniques for e-learning.

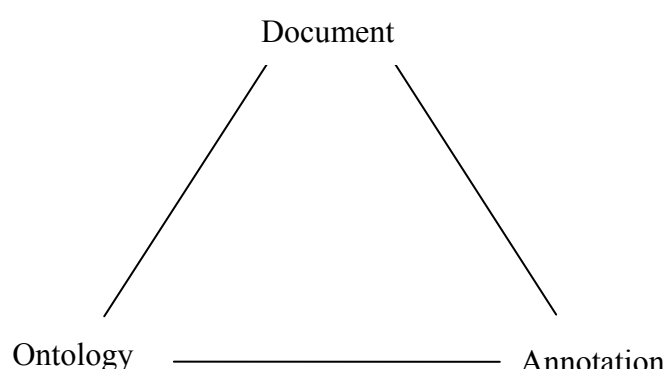


Figure 15 - The knowledge triangle

Given the important similarities in term of needs for formalized knowledge between the corporate memory scenario and an e-learning situation, the different methods and tools for corporate memories (Dieng-Kuntz, 04) can also be relevant in the context of the later. However if this translation across different domains might be quite powerful, there is major differences between e-learning and knowledge management (through organizational memories) that we need to stress.

- First, it is not sufficient to make knowledge available to learners for *learning* to occur. The complex human process of learning cannot be reduced to knowledge search, retrieval and

display. If providing the adequate knowledge is a necessity, it is not a sufficient condition for learning.

➤ The motivation and attitude of users is different. In the e-learning situation learners are motivated by an exam, a diploma, or a competency assesment. In the corporate knowledge management scenario users need to answer strategic choices, save costs, etc.

Given this differences we can also stress the common subgoal of finding and sharing information among stakeholders which completely justifies the cross-over we are investigating.

3.2.2. Ontologies

The LOM standard shows the interest of using metadata to annotate, select, organize or adapt learning content. However, metadata filled with free text are difficult to use automatically. In the LOM standard (see 3.1.3.1) propositions of standardized vocabularies are made but they are very limited. In ITS and AH systems, much richer models are used, and instead of vocabularies, networks of concepts are linked with semantic relations (Brusilovsky, 03). Ultimately, what those systems need is a conceptual representation of the domain being addressed. This representation must be generic and shared among all the stakeholders (i.e. learners, teachers, system, etc.). It must also be understandable by computer programs and may be coupled with implicit or explicit rules for automation like the graphs of AHA (figure 14).

In philosophy, the term “Ontology” refers to the subject of existence. It was borrowed by the artificial intelligence domain when it came to start formalizing knowledge. An ontology became a representation tool for knowledge engineers. In the context of knowledge sharing, an ontology identifies a specification of a conceptualization (Gruber, 95). An ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents.

The connection between e-learning needs and ontologies in AI is strong. The goal of conceptual representation in learning system and in ontologies is different. For learning, conceptual representations aim at specifying, sequencing and adapting courses using an abstract level of representation, like in the example of AHA developed figure 14. However, ontologies may help in that scope too. They possess a logical and a semantic structure that can be exploited. This section examines this tool by looking at the existing formalisms proposed to express ontologies and their associated benefits for automation (including inferences, reasoning, etc.). The last section is dedicated to the use of ontologies for e-learning, from pioneering work to most up-to-date realizations.

3.2.2.1. Practicalities and difficulties of ontologies

Characteristics of ontologies

Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. The ontology is an explicit representation of this conceptualization. Thesaurus, taxonomies, hierarchies of classes, etc. are flavors of ontologies. However, ontologies are not limited to these restrictive forms. Ontologies are usually composed of concepts (for example the concept of person) and relations (ex: two persons can be siblings). Binary relations are sometimes called “slot” depending on their original theoretical framework. Axioms can also express other kinds of knowledge. For example, if any two persons are siblings, then someone must be the mother of both of them).

Another facet of ontologies is that they express common representations. Ontologies are defined to be shared amongst different agents (whether human or machine). The semantics they carry is by definition the same for all.

Levels of expression

Depending on their usage, ontologies may have several levels of expression (Uschold and Gruninger, 96):

- Highly-informal: the ontology is expressed in natural language.
- Semi-informal: the ontology is expressed in a restricted and structured form of natural language to increase clarity and reduce ambiguity.
- Semi-formal: the ontology is expressed in an artificial formally defined language.
- Rigorously formal: the ontology is defined with formal semantics, allowing theorem and proof.

These levels of representation are in contradiction with the previous definition of a common description. A high level of formalism is needed for semantics to be machine processable, whereas human readers will prefer plain text. These levels can be viewed as different expressions of the same ontology. Another vision is that all these levels are just steps towards the rigorously formal state, in an ontology creation process. In this scope, ultimately all ontologies have to go for formal representations. An argument for this is that in formal ontology languages, places are provided to express natural language definitions that are not used by machines but by humans (e.g. comments).

The vision we prefer is that the level of formalism is dictated by the usage scenario and each level suits a specific need. Each representation is slightly different from one level to the other and it is not compulsory to go through each level to build a formalized ontology, especially if the formal representation is useless.

Examples

The philosophy of shared meaning has been pushed to its limits by the Wordnet project from Princetown Univ. WordNet is an online lexical reference system. Its design is “*inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets.*” (Wordnet, 06).

Even if the tenants of the project do not consider WordNet as an ontology, a W3C activity has started to propose a representation of WordNet using the Ontology Web Language (OWL, see 3.3.2.1).

Another example of a “world-class” ontology is SUMO (Suggested Upper Merged Ontology) which aims at providing definitions for general purpose terms and acts as a foundation for more specific domain ontologies (Niles and Piese, 01).

Apart from such generic ontologies, many domain specific ontologies have been developed, using different formalisms. The biological and medical domains especially are particularly active in this area with ontologies such as the Gene Ontology or UMLS meta-thesaurus and semantic network.

Ontologies about organizations are being developed in the context of more classical knowledge management applications, for example the O’Comma ontology (Gandon, 01).

3.2.2.2. Representations and formalisms

To ensure mathematical properties to ontologies they have to be expressed using a defined formalism. Formalisms are standard representations and the level of expressivity determines the possible inferences. Formalisms are used in a language, and languages themselves can be

physically expressed through serializations. The formalisms are abstract representations of the semantic of a language.

Formal semantics define representations based on logic. They can be processed by machines because of this logical nature. Various logics exist: First Order Logic, Description Logics, etc. Logics are interesting to perform inferences. Most classical inferences are transitivity (if a in relation with b and b in relation with c, then a is in relation with c) and subsumption (if x is of type a, or belongs to the extension of a, and b is a subclass of b then x is of type b, or belongs to the extension of b).

Decidability and complexity

An issue with formal representation is the decidability of the language. A language is decidable if and only if there is an algorithm that can determine if any sentence of this language can be proved valid or invalid. If no algorithm can solve this problem, the language is called “undecidable”. Propositional logic is decidable, because there exists for it an algorithm (truth-table construction) such that for every formula which combines M atomic formulas there is a maximum number $N = 2^M$ of steps such that after completing those N steps the algorithm will always decide whether the formula is valid or not. Here, a “step” of the algorithm has been (arbitrarily) defined as the completion of a row of the truth-table.

In practice, many languages are undecidable and still largely used in inference systems. Defining ontologies using a decidable language guaranties that an algorithm can validate any formula expressed from this ontology. However, there is no time limit. Information about complexity brought by the expressivity of a language is also an important aspect characterizing formalisms.

We describe below three formalisms for knowledge representation and manipulation that are of prime importance in the context of our work.

Description Logics

One of the “light-house” formalism for ontology is description logics (DL). Description logics are rooted in the AI work on Frames, Semantic Networks, Object-Oriented representations, etc.

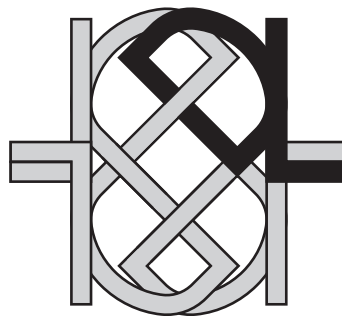


Figure 16 - A logo for description logics

DL separate between the A-box (assertion box) containing the knowledge statements and the T-box (terminological box) containing the ontological primitive used in the A-box (Baader and Nutt, 2002). Concepts define the classes or set of objects. Roles are binary relations on the objects. Concepts can be combined to define concepts that are more complex. For example, “an elephant is a mammal that is grey” may be expressed like this:

- $Mammal \wedge \forall color.Grey$ is defined in the Tbox,
- Babar:Elephant is an assertion of the A-box.

There is a whole family of DL, depending on their expressivity. The constructors of the language are used to determine the name of the logic. Table 4 gives an overview of a few classical description logics (Baader and Nutt, 2002).

Table 4 - Base description logic (corresponds to the multi-modal logic K) Defines ALC:

Constructor	Syntax	Example	Comment
Atomic concept	A	Mammal	The group of mammals
Atomic role	R	Color	The color relation
Conjunction	$C \sqcap D$	$\text{Mammal} \sqcap \text{Male}$	The group of male mammals
Disjunction	$C \sqcup D$	$\text{Tamed} \sqcup \text{Wild}$	Tamed beings are not Wild and vice versa
Negation	$\neg C$	$\neg \text{Carnivore}$	Non-Carnivore
Existential restriction	$\exists R.C$	$\exists \text{has-child.Mammal}$	Has a child that is a mammal
Value restriction	$\forall R.C$	$\forall \text{color.Grey}$	Has always a grey color

Other constructors have been introduced, like number restrictions (ALCN), inverse roles, transitivity over the roles, etc. DLs are used to compute subsumption and generate according hierarchies. They can also determine the satisfiability of a concept (can it exists given the existing knowledge?).

Conceptual graphs

Conceptual graphs (CG) are a system of logic based on the existential graphs of Charles Sanders Peirce and the semantic networks of artificial intelligence (Sowa, 84). They express meaning in a logically precise form. This form aims at being humanly readable and computationally tractable. The connection with natural language is stronger than in other logics. The readable, but formal, design is obtained through a graphic representation. The ISO Common Logic standard is based on CGs, and proposes a language CGIF (ISO CG, 06).

A conceptual graph is bipartite graph, which consists of two kinds of nodes called concepts and conceptual relations. Relations can be n-ary relations. Concepts have a concept type and a referent. Concept types are organized in a hierarchy between the universal type and the absurd type. Relations are also organized in a hierarchy.

Several languages can be used to express conceptual graphs: the graphical display form (DF), the formally defined conceptual graph interchange form (CGIF), and the compact, but readable linear form (LF). Logically CGs can be translated to predicate calculus and expressed in the KIF language. Examples of those various languages expressing the graph shown on figure 17 are presented below, (Sowa, 84):

- LF [Elephant:Babar]@ (hasColor)@ [Grey]
- CGIF [Elephant:Babar *x] [Grey: *y] (hasColor ?x ?y)
- KIF (exists ((?x Elephant) (?y Grey))
 (and (Name ?x Babar) (hasColor ?x ?y)))
- Predicate calculus (\$x:Elephant) (\$y:Grey)
 (name(x,'Babar') \cup hasColor(x,y)).

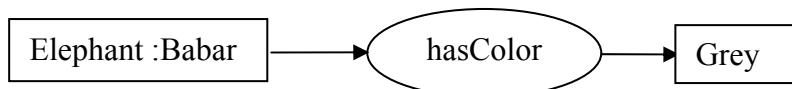


Figure 17 – Example of a conceptual graph, expressing that “Babar, an elephant, is grey”

In the last two, we see that the translation is not straight forward. The “name” relation had to be introduced, even if it does not exactly belong to the semantic of the above languages.

In CG, the deduction problem is performed by a graph homomorphism, called projection (Mugnier and Leclerc, 05). The existence of a projection from a graph Q to a graph G means that the knowledge represented by Q is deductible from the knowledge represented by G. This offers the possibility to query the graphs, by defining a query Q as a graph and looking for the projections G. All the Gs matching the projection are answers to the query. While possessing weaker semantics than some DLs, CGs present the advantage of being easier to read for humans. CGs have been implemented in a variety of projects for information retrieval, database design, expert systems, and natural language processing.

Extensions of simple conceptual graphs using rules and constraints have been proposed and studied in (Baget and Mugnier, 02). Rules can be expressed as bi-colored graphs. The first color indicates the premise of the rule and the second color its conclusion. Two kinds of rules are defined: the inference rules that explicit knowledge and the evolution rules that might violate existing constraints. For the conceptual graph formalism including both types of rules and constraints, the problem of deduction (is there a projection from any graph Q into the knowledge base) is undecidable. For formalisms without constraints and just rules (in that case, both type of rules can be merged) the problem is decidable for all the positive instances.

Topics Maps

A formalism and associated language that has already been standardized by the ISO committee is the Topic Map standard (Biezunski et al., 99). It originates from the XML community. Topic Maps have been designed from the beginning to express knowledge about documents. The basic idea behind topic maps is to formalize the topics describing a resource. Topics can be instance of other topics (their type) and have one subject. This is represented in the topic graph by t-nodes (topics), a-nodes (associations), and s-nodes (subjects). The topic is described by its occurrence and basename. Topics are related through association links. A scope defines the namespace for the topics.

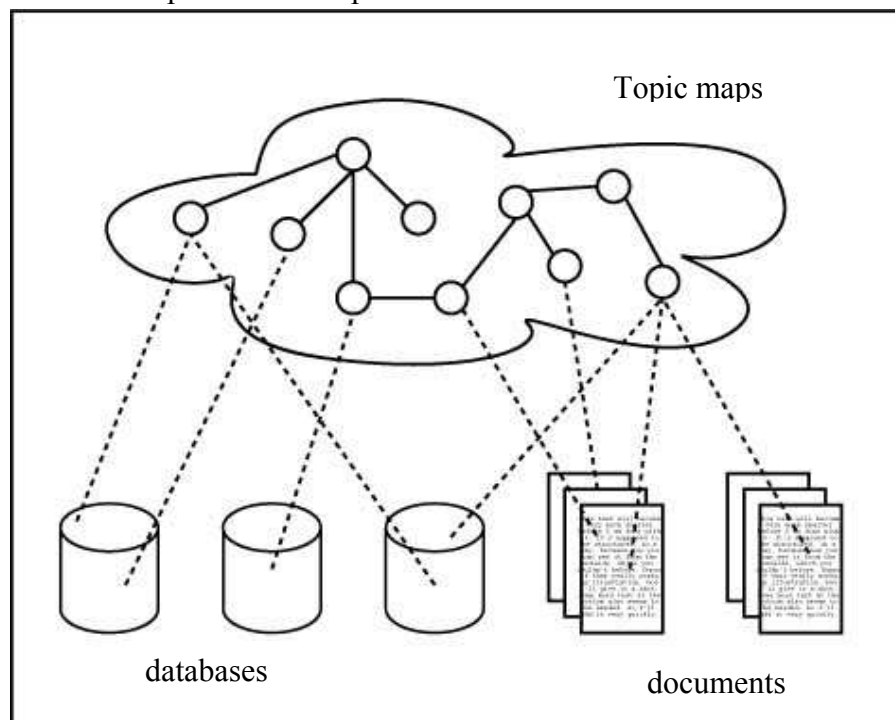


Figure 18 – The vision of document and data base indexation with topic maps (Garshol, 03)

The topic map model and associated formalism is oriented for the practice of indexing documents, it is often associated with works on digital libraries. Issues about reasoning, and inferences do not seem to be a primary concern, as compared to DLs. Inferences is seen as a property of systems and not of documents, thus it is not standardized in the topic map formalism.

3.2.2.3. *Ontology for e-learning*

Ontologies are useful tools whenever knowledge needs to be formalized and exchanged. E-learning is then an interesting field for the use of ontologies.

Formalization goal

Ontologies are of primary interest for learning in general: they are representation of knowledge and in a learning situation several types of knowledge are involved.

(Mizoguchi et al., 97) already pointed out the fact that Intelligent Educational System (IES) were already manipulating knowledge and that we would gain at formalizing the declarative representation of what the systems knows. The main goals pursued are:

- Making the conceptualization on which the system is based explicit.
- Standardizing the vocabulary.
- Facilitating the communication with humans.
- Making knowledge more reusable.

This is equivalent to the objectives of « conceptual knowledge » mentioned previously in examples of ITS and adaptive systems. Now, with the introduction of ontological engineering in education, the identification of the knowledge involved in the learning systems becomes crucial (Mizoguchi and Bourdeau, 00).

To start with, the formalization of the **domain** to learn may have several uses:

- First, it serves as an explicit representation of the knowledge that can be used directly to provide students with a correct vision of the concepts they have to learn. This approach is close to those inspired by the use of visual concept maps for helping or assessing learner's conceptual knowledge (see 3.2.3.2).
- Domain ontologies serve as a common representation to annotate resources. In an information retrieval perspective, queries can be performed using these common representations. They disambiguate user queries compared to text based search (Brase and Nejdl, 03). In addition, the structure of the ontology can be used to extend the search results to other concepts and thus improve the recall of the search mechanism.
- The domain ontology can be directly used to structure learning paths. Student follow subsumption links to learn one concept after another (see Memorae experiment (Abel et al., 04)).

The use of domain ontologies is supported by the subsumption theory of D. P. Ausubel (Ausubel et al., 78): *“A primary process in learning is subsumption in which new material is related to relevant ideas in the existing cognitive structure on a substantive, non-verbatim basis. Cognitive structures represent the residue of all learning experiences; forgetting occurs because certain details get integrated and lose their individual identity”*.

(<http://tip.psychology.org/ausubel.html>, 06)

The cognitive structure is made explicit through the domain ontologies. Ausubel clearly indicates that his theory applies only to reception (expository) learning in school settings. He distinguishes reception learning from rote and discovery learning, the former because it does not involve subsumption (i.e., meaningful materials) and the latter because the learner must discover information through problem solving. However, we shall see that domain ontologies can be used in these cases as well.

The **pedagogical** aspects can also take the form of an ontology. In this case, the following goals are pursued:

- First, it serves as a communication language between the actors (teachers/student). Like the domain ontology, defining the pedagogical concepts helps understanding the teacher's vision as well as the role of resources in the learning process.
- But pedagogical ontologies are also used as a communication language between teachers and programs. Pedagogical plans might be expressed using those concepts and later on understood by automatic systems to implement learning strategies. Here we connect pedagogical ontologies with learning designs (see 3.1.3.3). Ontological knowledge at this level is deeply connected to processes and actions.

Ontologies may express rules. Such rules are used to plan the reactions of ITS of AH systems. In (Jacquiot et al., 06), the "layered" approach illustrates the need to separate domain related knowledge from domain independent "meta rules". We understand this "meta level" as a pedagogical ontology expressing the navigation strategy.

Finally, programs need to model the **learning environment** to be able to interact with the learner. Information concerning the resources, and how to manipulate them, is also compulsory. Knowledge about the user must be represented in some way. Ontological representations may act at two levels: On an abstract level, disconnected with the domain and the application (stereotype models), and at a lower level, formalizing each user's knowledge (overlay models).

Examples

This section presents a few prominent examples of ontologies and their use in ontology supported learning.

A minimalist vision is proposed by (Brase and Nejd, 03), based on the metadata approach. The fields provided by the Dublin Core standard (DC, 06) are used to link resources to ontological concepts through the unique "dc:subject" relationship. Ontologies are limited to taxonomies of concepts from the domain. The LOM (see 3.1.3.1) standard is used to express the taxonomy and formalize the concepts using RDF (see 3.3.2.1). This simple vision figures what is most commonly realized in practice using ontologies to annotate resources. For example, this approach was chosen by the Edutella peer-to-peer network (Nejd et al., 02).

For (Mizoguchi et al., 97) the pedagogical knowledge classically formalizes the meaning of the following terms:

- Nouns: "Problem", "Scenario", "Answer", "Example", "Operation", "Hint", etc.
- Verbs: "Provide", "Show", "Ask", "Simulate", etc.
- Adjectives: "Unsolved", "Easy", "Correct", etc.
- Constraints: "Rationality", "Preference", "Condition", etc.

The role of the pedagogical ontology is then to provide shared vocabulary between the participants as well as a formalization of the processes of learning and tutoring.

The same distinction between domain and pedagogy is introduced by (Abel et al., 04) in the Memora experiment, but with inversed meanings: the pedagogical "application" ontology is relative to the domain to learn, whereas the "domain" in this case concerns the learning strategies. The pedagogical model also expresses knowledge about resources, like "Audio", "Video", etc. and about the agents: "student", "teacher". The domain ontology is used to guide learners by following the "ontological axes" along which the ontology is built.

Doing this the ontology defines the navigation possibilities in the set of resources that constitutes an “organizational memory”.

In (Breuker and Bredeweg, 99) three ontologies are introduced to cope with the problem of reusing existing ontologies. The generic part, allegedly reusable as part of the common understanding is called the *top* ontology. The *core* ontology provides the initial structure to distinguish the major categories of knowledge by setting up a top-down knowledge acquisition framework. Finally, the *domain* ontology relates to the standardized terminology used in the community.

In (Desmoulins and GrandBastien, 02), the emphasis is put on the document in the context of the exploitation of technical documentation for training. In addition to pedagogical and domain ontologies, an ontology formalizing the syntactical characteristics of the documents (technical documents) is introduced. Ontologies are used to describe existing documents and create from those documents a set of fragments that can be retrieved and used to construct specific courses. The ontological formalization in this work ranges from the “top” or generic ontologies to the “domain” or regional ones. The pedagogical ontology presents several facets:

- The types of descriptions
- The different types of knowledge (meta-cognition, declarative, perceptual, reasoning, procedural, performing skills and attitudes).
- The pedagogical uses of the fragments

Finally it formalizes in an ontology the pedagogical activities and proposes patterns of pedagogical scenarios using these descriptions.

In an approach closer to classical knowledge management preoccupations in companies, (Schmidt and Winterhalter, 04) divide ontological knowledge in four areas:

- Organizational ontology (roles, departments)
- Process Ontology
- Task Ontology
- Knowledge area Ontology

The Bloom’s taxonomy (see table 3) is presented as a major inspiration for pedagogical ontologies as it describes the different levels of learning outcomes. In the ontology presented on the left figure 19, the different types of learning resources are defined (Ullrich, 04). They can be related to the Bloom’s taxonomy.

A last example is the distinction proposed by (Leidig, 01) that identifies the following six dimensions that are quite complete:

- *Subject matter dimension*, which we call “domain” ontology
- *Competence dimension*, expressed for example in the Bloom’s taxonomy
- *Medium dimension* for technical description of the artifacts
- *Knowledge dimension*, which we call “pedagogical” ontology, this dimension is presented in figure 19 on the right.
- *Rhetoric dimension*, which identifies relations like pre-requisites between concepts.
- *Interactivity* to describe the types of interaction an object might propose.

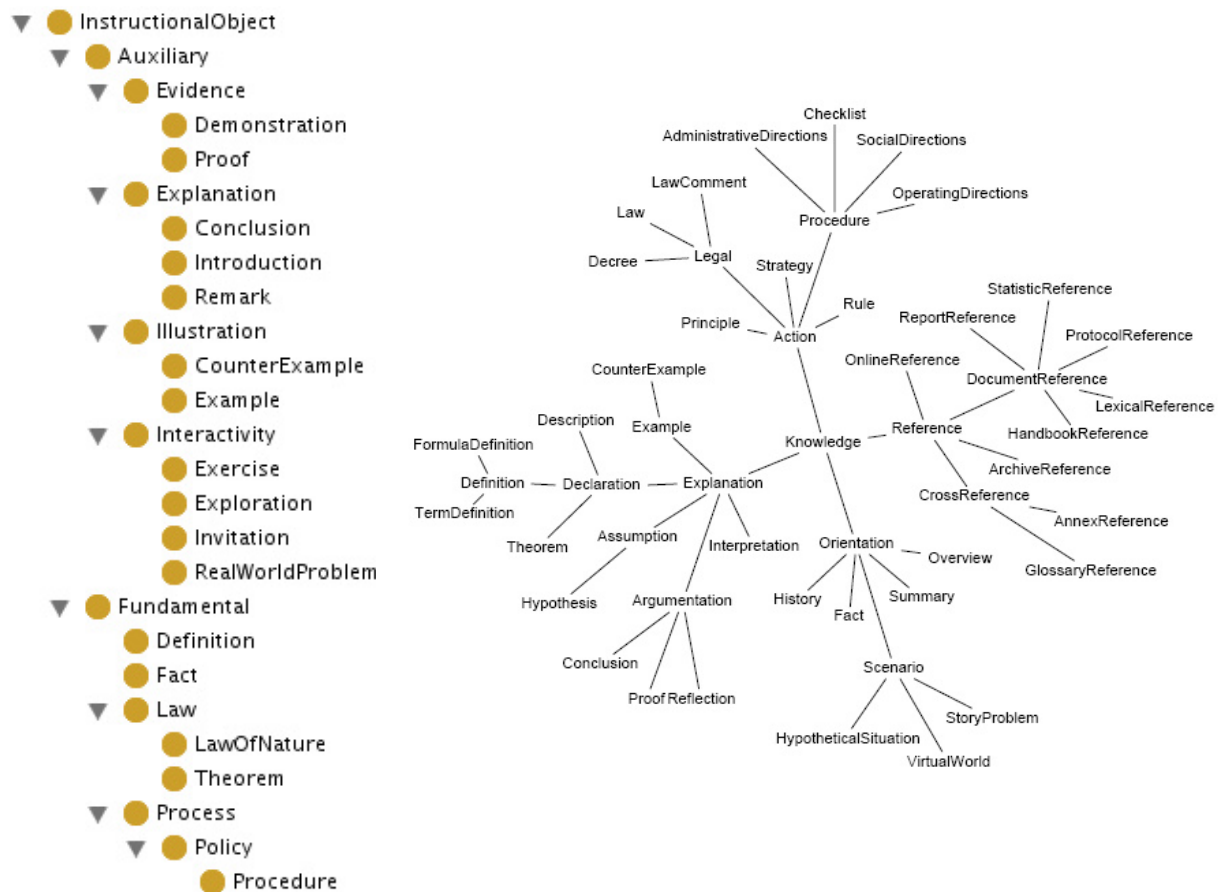


Figure 19 - Example of pedagogical ontologies (Ullrich, 04) (Leidig, 01)

To try to share and exchange all this different ontologies, we mention the initiative of (Dicheva et al., 05) that proposes to describe the domain of ontologies for education in an ontology and to build a portal that would allow practitioners to find appropriate models. Still up to now, such proposals have met little success and there is no real place to search for the various produced models.

What also transpires from this quick review on ontologies for learning is the variability of the representation modes. Even if the knowledge expressed is often quite similar, the point of view from which it is used depends on the application. This means that the process of reusing ontologies is certainly valuable but the modality of reconstruction and adaptation of an existing ontology to a new situation are still to be clarified.

3.2.3. Annotations

3.2.3.1. Definition

Defining ontologies is one important step for the creation of knowledge bases for learning. However, ontologies only define the concepts that are abstraction of the reality. Concepts must be instantiated or linked to real world instances. This connection is done through a process of annotation.

For (Azouaou and Desmoulins, 05) an annotation can be any object that a person adds to a document with a specific objective. An annotation on a document has the following attributes:

- an anchor, indicating where the annotation has been placed in the document.
- a visual form, ex: coloring, underlining, etc.

- episodic attributes related to the action of creating the annotation (author, date, etc.).
- semantic attributes that describe the content of the annotation and its meaning.

Different ontological categories are also proposed by (Azouaou and Desmoulins, 05) to identify the purpose or meaning of an annotation in a learning context (structuring, criticize, enrich, etc.).

3.2.3.2. Graphical representations

Annotations “link” concepts and instances together. There is a strong visual metaphor behind it. The use of graphical representations of knowledge through graphs has been formally described in a theory called “concept maps” (Novak and Canas, 06). Concept maps are tools for organizing and representing knowledge. They represent concepts usually enclosed in circles or boxes, and relationships between concepts. Spatial disposition of the concepts is also meaningful, the most general concepts being placed at the top. The similarities with conceptual graphs and ontologies are important, but they differ in their use. Concept maps just act as visual representations for humans, no computer manipulates them. As a learning tool concept maps link to Ausubel learning theory (see above) of presenting the knowledge into hierarchical form. The same outcome as for formal ontologies is expected from concept maps, except that the visual aspect is emphasized in concept maps.

Using two dimensional representations of knowledge is also proposed under the form of metaphorical geographic maps (Armani and Rocci, 03). In this case, learning is like discovering a new country. Different types of maps are identified, base on linguistic work on discourse analysis:

- The fil rouge map to suggest a path through documents.
- A cataphoric map to present the titles.
- An anaphoric map to detail the previous ones, and facilitate the understanding of concepts.

In (Stutt and Motta, 04) the same idea is applied to represent knowledge about argumentation in “knowledge charts”. The argumentation is modeled to be reused and exploited by machines in browsers. This visual representation of graphs has many connections with the ontology representation and the formalisms of the semantic web.

3.3. Semantic Web

3.3.1. The vision

The W3C home page for the semantic web activity states: *“The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming.”*. (W3C, 06)

In (Berners-Lee et al., 01) Tim Berners Lee presents his vision of the “Semantic Web”. The idea is that knowledge could be understood by both humans and machines, enabling programs to perform intelligent searches using various sources of information. The semantic web is seen as an extension of the actual web where information would have a well defined meaning. It builds on knowledge representation techniques, but in an “open-world” perspective, accepting incoherent knowledge and unanswerable questions, as opposed to centralized “expert systems”. Still logics and rules could be applied on the formalized knowledge.

The semantic web is founded on two major technologies: XML and RDF. XML brings structure to documents, while RDF supports the expression of meaning. RDF represents knowledge as triples, linking a subject to an object with a predicate. To conceptualize the meaning of RDF triples, reference representation are needed. This task is fulfilled by ontologies.

The semantic web components are usually represented in a “layered cake” (see figure 20). The layers figure the different levels of formalism necessary to reach a “Web of trust”, where not even the meaning of information is understood but also its “value” (universal, limited in time, space, etc.).

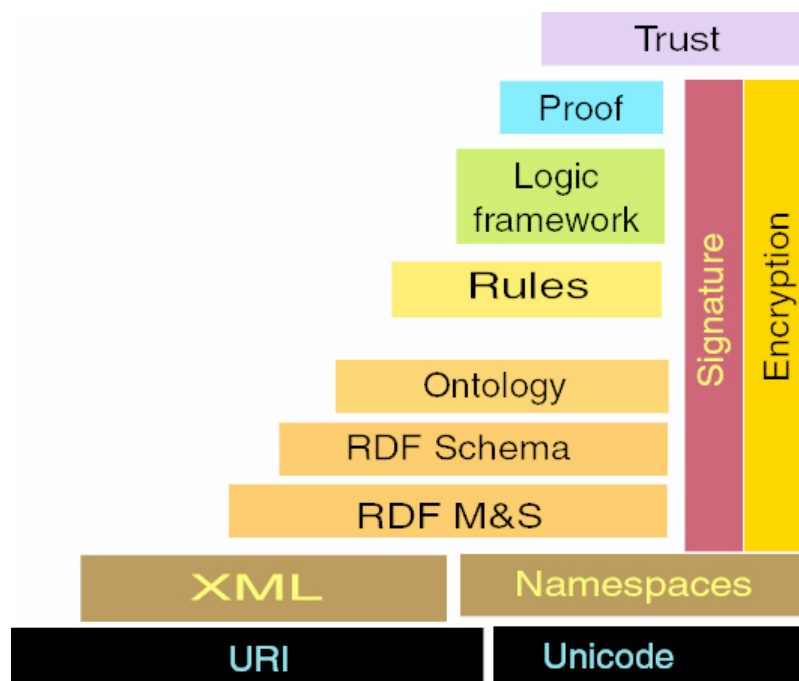


Figure 20 – The semantic web layer cake (W3C, 06)

3.3.2. Standards

The semantic web is supported by a set of standards and recommendations, developed and endorsed by W3C.

3.3.2.1. *RDF/RDFS/OWL*

RDF

The Resource Description Framework (RDF) is a general-purpose language for representing information in the Web (RDF, 06). It is an open-world framework allowing anyone to make statements about any resource.

RDF can be represented as graphs. An RDF graph is composed by triples. A triple is formed by with a resource, a property and a value. The resource is an entity defined by its URI (Unique Resource Identifier). A property defines a binary relation between resources or between a resource and a value. The property allows us to attach semantic information to resources.

RDF can be encoded using different syntaxes:

- The RDF/XML binding uses an XML syntax:

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:dc="http://purl.org/dc/elements/1.1/">
4.   <rdf:Description rdf:about="http://www.inria.fr/acacia/TheseSylvain">
5.     <dc:description>The thesis written by S. Dehors</dc:description>
6.     <dc:title xml:lang="en"></dc:title>
7.     <dc:title xml:lang="fr"></dc:title>
8.   </rdf:Description>
9. </rdf:RDF>

```

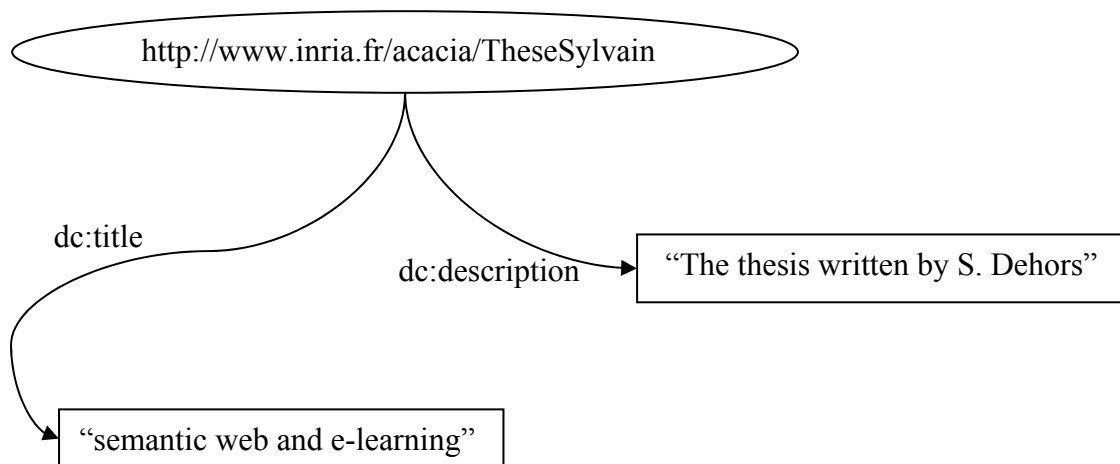
- The N3 notation is closer to the triple interpretation of RDF

```

http://www.inria.fr/acacia/TheseSylvain dc:description "The thesis written by S. Dehors"
                                         dc:title "semantic web and e-learning"

```

- The graphical view of nodes (resources) linked by properties is also considered in the standard.



RDFS

RDFS uses RDF to describe RDF vocabularies. It defines a meta-model for representing classes and relations. It also indicates which classes and properties are expected to be used together. This meta-model can be instantiated to represent vocabularies in general and thus ontologies. In other words, RDF Schema provides a type system for RDF.

RDF Schema allows resources to be defined as instances of one or more classes. In addition, it allows classes to be organized in a hierarchical fashion; for example a class `ex:Dog` might be defined as a subclass of `ex:Mammal` which is a subclass of `ex:Animal`, meaning that any resource which is in class `ex:Dog` is also implicitly in class `ex:Animal` as well. However RDF class and property descriptions do not create a constraint model into which information must be forced, but instead provide additional information about the RDF resources they describe. Triples cannot be inferred as wrong information. They can only express additional information. For example if a resource is said to be a definition and is linked by a relation valid for examples, then this resource is inferred to be both an example and a definition.

The RDF/S (RDFS+RDF) language has been connected to the Topic Maps representation. Both formalisms share many similarities (Garshol, 03). For example, RDF resources can be mapped to subjects and nodes of the RDF graphs and are close to the idea of topic. Some structures like associations can be easily transposed from one standard to the other. It is also envisioned that higher level of semantics provided by RDFS and OWL (see below) can map

to the TMCL (Topic Map Constraint Language). Finally, the topic map formalism possesses more primitives than RDF/S but shows no striking advantage over the W3C recommendation.

The RDFS model has been compared with Conceptual Graphs. A mapping is proposed by (Corby et al., 00). Classes can be modeled as CG concept types, and properties with domain and range match CG binary relations with signature. CGs are also seen by (Berners-Lee, 01) as a potential base structure for the semantic web language. Doing this RDFS can take advantage of existing inference mechanisms on CG like projection for example (Mugnier and Leclerc, 05).

OWL

The Ontology Web Language (OWL) is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of web content than what is supported by XML and RDF/S. It does so by providing an additional vocabulary along with formal semantics. It relies on RDF to express its primitives. Three levels of expressivity are distinguished: Lite, DL and Full. The DL level is proven decidable whereas the Full is not. For this it restricts part of the expressivity, especially classes cannot be used as individuals, a restriction not present in RDFS. In general, OWL builds on RDFS expressions, but only the OWL full level is fully compatible with RDFS (W3C, www).

RDFS semantics are defined with “if” conditions whereas OWL has “iff” conditions. It results in a “stronger” semantic. The conclusions that can be drawn on the expression of the language (entailments) are stronger. For example in RDFS if *c* and *d* are classes, then RDFS requires that if *c* is a subclass of *d*, then the extension of *c* (i.e. the set of instances of *c*) is a subset of the extension of *d*. On the other hand, OWL DL and OWL Full require that *c* is a subclass of *d* if and only if the extension of *c* is a subset of the extension of *d* (Horst, 05).

3.3.2.2. Querying on the semantic web

Representing knowledge is important but the capacity to query this knowledge is equally important. In this scope, query languages for the semantic web have been proposed.

In the context of the Edutella (Nejdl et al., 02) peer-to-peer network, a family of query languages (RDF-QEL-i) based on Datalog has been defined. This family ranges from purely assertional capabilities (RDF-QEL-1), to complex levels allowing recursive definitions. The queries are expressed in RDF syntax. Aggregation functions like count, average, min, max, are included in the language group whose name is appended with ‘-A’ (ex: RDF-QEL-2-A). This family of languages can wrap around existing query languages typically supported by Edutella peers: RQL, TRIPLE, SQL, XPath, etc.

SPARQL (SPARQL, 06) is the query language proposed by W3C. The SPARQL query language is based on matching graph patterns. The simplest graph pattern is the triple pattern, which is like an RDF triple, but with the possibility of placing a variable instead of an RDF term in the subject, predicate or object positions. Combining triple patterns gives a basic graph pattern. To fulfill a pattern an exact match to a graph is needed. The query language integrates specific tags for optional graph pattern, like union of patterns, unbound variable, or specific filtering. Figure 21 shows an example of a SPARQL query, asking for the titles of the resources which price is less than 30.

```

1. PREFIX dc: <http://purl.org/dc/elements/1.1/>
2. PREFIX ns: <http://example.org/ns#>
3. SELECT ?title ?price
4. WHERE { ?x ns:price ?price .
5.         FILTER (?price < 30) .
6.         ?x dc:title ?title . }

```

Figure 21 – A simple SPARQL query

3.3.3. Architectures for the semantic web

The semantic web does not define a specific architecture for its applications. It is supposed to build on the actual web. Then it is worth looking first at existing concepts in this domain to better identify the different architectures currently deployed.

The most common setting on the web is the *client-server* architecture. In the context of the semantic web, it means that all the semantic data and resources are stored on a single server. It may be replicated for large systems, but this is transparent to the user. Users access and use the system from “clients”, usually web interfaces. All the computational operations are carried out by the server. This is well adapted when the number of users is limited. Otherwise, it implies very powerful servers and large communication bandwidth. The characteristic of storing all material and processing it in one place offers more possibilities for maintaining coherence and guaranteeing safe evolutions of the application. In addition, both content and metadata (or annotations) can be managed by a single agent. Such centralized approaches reduce the possibilities of free contributions to the base, and may face scalability issues. In the e-learning domain this approach is typically adapted for ITS in small communities of learners.

Another orientation, more in line with the semantic web idea as an extension of the actual web, is to use search layers to independently crawl semantic data and offer search interfaces that can combine results from the various sources of the whole network. Using engineering terms, this might be called a “*semantic middleware*” (Stojanovic et al., 01). It allows us to deal with very large sets of resources, and a large number of users.

Semantic data collected from different sources must be “aligned” in some way. This can be performed by relying on standard ontologies, in which case the alignment problem is deported on the annotation task, or using matching algorithm. Such algorithm can be potentially automatic.

In the *peer to peer* architecture there is no server centralizing the information, each participant (peer) holds, and is responsible for, part of the resources and knowledge shared among all the peers. This obviously increases the scalability. Adding new material is also very easy. However, ensuring quality and trust soon becomes a hard problem. The Edutella project (Nejdl et al., 02) proposes an implementation of this architecture using semantic web technology and a peer-to-peer network infrastructure. Each node of the network acts as a provider of resources as well as semantic information. Metadata are expressed in RDF and queries, in different levels of RDQL can be asked and propagated through the network. Even though a very interesting example by its size and its real implementation, the Edutella project fails to address crucial problems like interoperability of the metadata and of the material. The allowed annotations are a subset of LOM, so most of the useful information is in fact in the form of free text (see 3.1.3.1).

3.3.4. Tools for the semantic web

One of the key ideas of semantic web is interoperability of formalized knowledge. Knowledge is represented in standard formalisms with defined semantics that can be exchanged, and used,

in different applications. All the agents can rely on the standard formalisms to implement this interoperability. It is especially important for tools to respect those standards. We distinguish five categories of tools, or standard components, necessary to set up a “semantic web: first, for the formalization of ontologies, then for the expression of knowledge and its exploitation through visualization, search and navigation. The order in which the tools are presented follows a classical development cycle for an “organizational memory” (Dieng-Kuntz, 04).

3.3.4.1. *Ontology editors*

To be exchanged and manipulated on the semantic web, knowledge must be formalized through shared ontologies. Ontology editors for OWL, like Protégé (Protégé, 06) or SWOOP (Swoop, 06), offer dedicated interfaces to manually author ontologies and generate OWL files.

Those ontology editors share the following characteristics:

- They allow browsing the ontology structure, usually making use of hierarchical tree views.
- They offer full edition of the ontology.

The screenshot on figure 22 illustrates this in the Protégé ontology editor. The hierarchical structure of the classes in the ontology is visible on the left. The different characteristics of the selected class (“Fundamental”) can be edited on the right. This application can be customized using many plug-in, to create forms, include reasoning programs, etc.

Another example is the Swoop ontology editor from university of Maryland. It offers the same basic functionalities, but is a lighter application, inspired by web interfaces (presence of a back button, hyperlinks). Both mentioned tools are generic editors. They require a good knowledge of OWL formalism and they can only be used by experts, or so-called “knowledge engineers”. When the ontologies have to be built directly by other categories of users, such as domain experts, dedicated interfaces and editors are necessary.

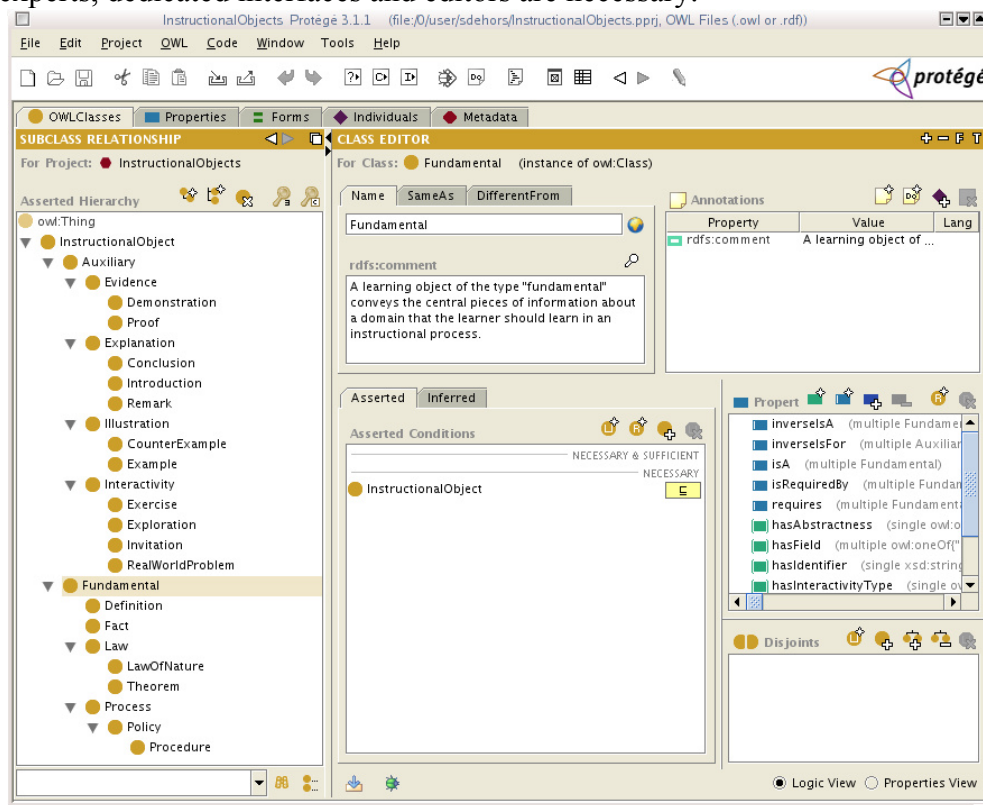


Figure 22 - Screen shot of the protégé interface

3.3.4.2. *Annotation tools*

On the semantic web not only documents can be annotated, but any resource identified by a URI. That means agents, places, concepts, etc. In order to perform this annotation of the resources, a number of tools are again necessary. We regroup them under the generic term of annotation tools. A review of annotation tools for KM in general is proposed by (Uren et al., 05). Seven key points are identified:

- The editor must provide standard format for input and output.
- It must be integrated within the environment where users manipulate the documents, and in the global collaborative process.
- It should support multiple ontology references.
- Multiple document formats must be supported.
- Evolution of the document must be taken into account.
- Annotation can be stored either in the document itself or in a separate container.
- Facilities for automation have to be provided.

Some of these points are still hot research topics, but they give a good overview of the complexity of the task. For a list of exiting annotation tools, please consult (Uren et al., 05).

3.3.4.3. *Visualization*

Ontologies are commonly represented by graphs structures. Tools like IsaViz or plugins for Protégé, allow visualizing the graph nature of RDF. However, RDF does not specify physical coordinates for displaying the nodes and arcs is encodes. Tools must rely on automatic algorithms for the placement of nodes and arcs on a two dimension level. As explained in the concept map theory (Novak and Canas, 06) the physical location may carry some semantics that is not encoded in RDF.

Another difficulty raised by RDF is the definition of the RDFS language in RDF itself. When flattened on a graph view it might be difficult to separate the different levels of modeling. In the case of ontology editors like Protégé the class hierarchy is clearly separated from the instances. It facilitates comprehension. Globally specific presentation patterns must be adapted to efficiently visualized RDF graphs. This is illustrated by the “stylesheet” approach of Isaviz (IsaViz, 06)

3.3.4.4. *Search engines*

One of the main functionalities of web applications for knowledge management is information retrieval (IR). IR on the semantic web takes advantage of ontologies and annotations to allow users to ask ontology-based queries. Semantic search engines are necessary to perform this task. Those search engines take ontologies in RDFS or OWL and annotations in RDF to answer queries expressed in query languages like RQL, SPARQL, etc. Semantic search engines are not standalone applications but provide the semantic middleware that needs to be connected with dedicated interfaces. We mention the following existing platforms:

- Ontobroker (Ontobroker, 06), which is now a commercial product, and KAON2 (Motik and Satller, 06) based on DL reasoning.
- The Corese semantic search engine (Corby et al., 04), internally based on conceptual graphs. It has been embedded in a complete web platform called sewese (SEmantic WEB SErver).
- Other tools that include the Jena framework, developed by HP (Jena, 06), Sesame (Sesame, 06), Triple (Triple, 06).

3.3.4.5. Semantic browsers

Another way to exploit formalized knowledge and annotated documents is to follow the analogy with the actual web and offer browsing of the available knowledge. This is called “conceptual browsing”, and finds a direct expression in the context of the semantic web.

The idea is supported by a number of “conceptual browsers” with different navigation paradigms.

“Conceptual navigation” as presented by (Naeve, 01) may consist in visualizing graphs representing the conceptual knowledge and accessing related documents. In the Konzilla browser, users visualize what is called a “knowledge manifold”. Figure 23 below shows a screenshot of the Konzilla interface. On the left part, the conceptual representation is visible using some UML-like graphical representation (it proposes also an RDF model). On the right part, the content of the document linked to the concept is displayed. Such navigation paradigm is also described by (Stutt and Motta, 04), but without any associated implementation.

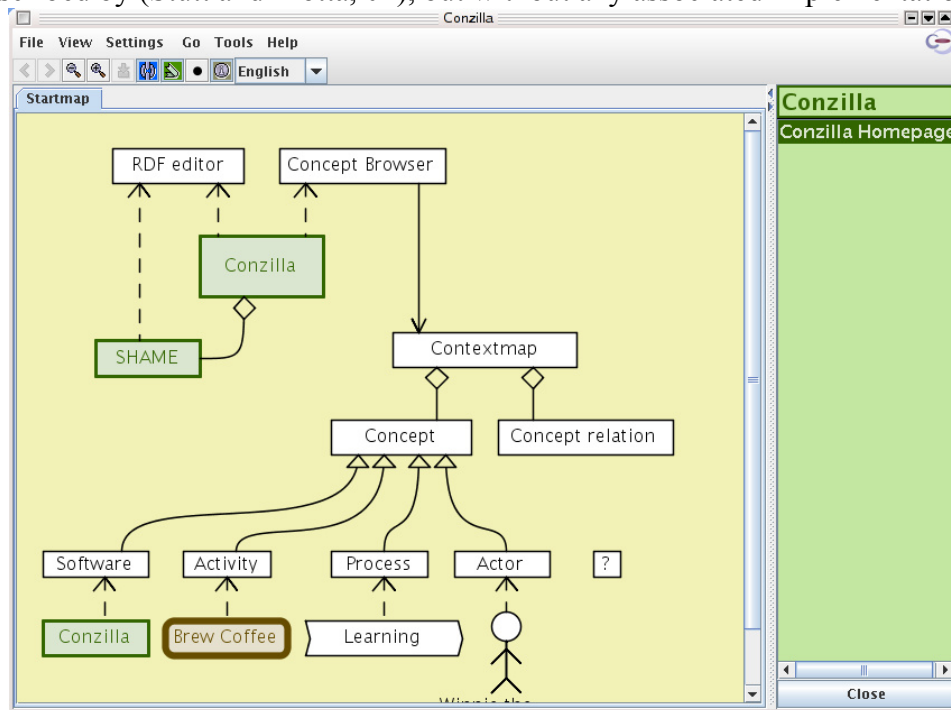


Figure 23 - Screenshot of the Konzilla conceptual browser

Another type of browser is proposed by the Magpie plug-in (Dzbor et al., 05). Instead of visualizing a conceptual structure and the associated annotations, the browser relies on classical hypertext navigation, enhanced by automatic annotation. Terms on a web page identify concepts from a selected ontology. By right clicking on the term a list of available services relative to this concept are proposed. Navigation is then driven by the concepts found in the pages and not only by hyperlinks. Figure 24 shows a web page enhanced with the identification of the concepts from an ontology about semantic web topics. For a spotted concept a list of possible services is proposed in a menu appearing when right-clicking the concept.

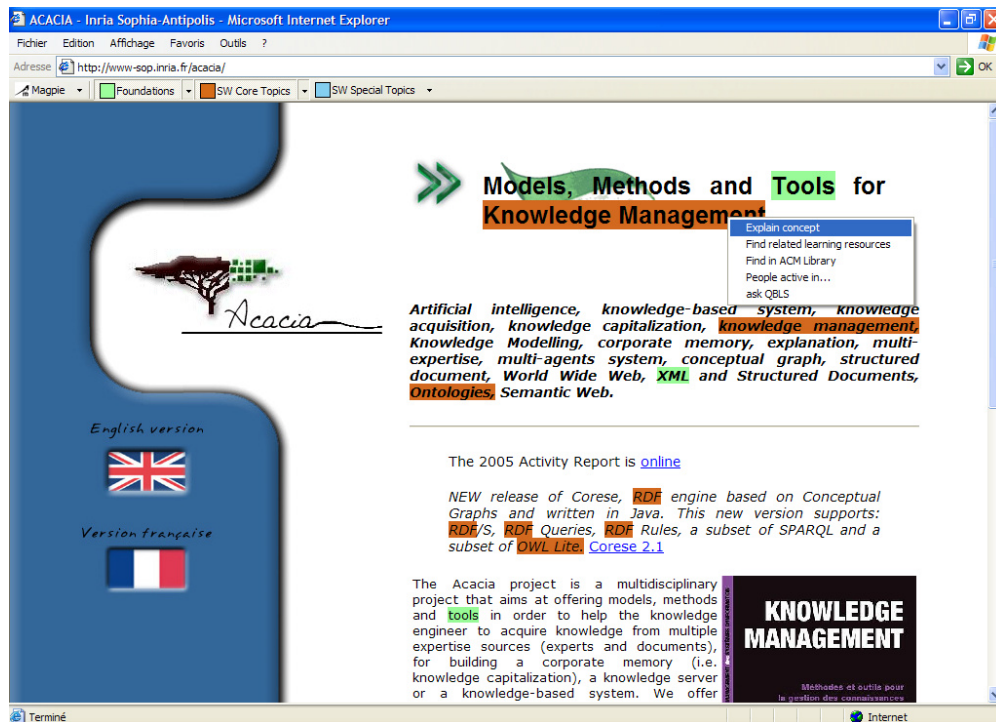


Figure 24 - The magpie plug-in

The tools and associated use presented above are generic. Most of these ideas can be very interesting in the context of e-learning. For example, as experimented with earlier “semantic hypertexts” (Murray, 03), “conceptual navigation” may help learners in exploiting a course document. In the next part, we present the actual state of the research on semantic web that actually focuses on learning.

3.4. Semantic web for e-learning

The semantic web proposal as it stands today, poses key research issues that are largely shared by the e-learning community. Problems of generation of semantic annotation or metadata, interoperability of expressed knowledge and assessing the domain of validity of this knowledge are amongst the most interesting for us, considering the knowledge management orientation we are focusing on.

3.4.1. Visions

The idea of applying the semantic web paradigm to e-learning counts numerous contributors. Some of them try to describe the implications and outcomes on a purely hypothetical basis, while others have developed and implemented real systems. We discuss in this section the proposal of some of the thinkers, who entitled their work with generic stances like “E-learning in/and the semantic web” or “Semantic web for e-learning”.

In 2001, (Stojanovic et al., 01) proposes to rely on the semantic web to perform search and retrieval of learning resources on the web. It is envisioned that on the semantic web, learning materials are semantically annotated. Using ontologies would solve the problem of variability with free text annotations met with the LOM standard. The learning context would be also precisely defined. The structure of learning material can be understood by machines if the resources are described using a structural ontology. Thus relying on all these ontological representations, course reuse and customization is presented as a direct benefit of using the semantic web.

In (Devedzic, 04), the introduction of the semantic web annotations in educational material is said to provide both an interactive environment by allowing students to query for material, and a framework for pedagogical agents to collaborate and propose tailored course material. The importance of relying on standards is stressed and difficulties of authoring ontologies and annotations are mentioned. It is also interesting to point that the author described the lack of early adoption of W3C standards in the AIED community. For intelligent tutoring systems and adaptive hypermedia, this move has not been really done yet.

Considerations of the interoperability aspects addressed by the semantic web are developed in (Aroyo and Dicheva, 04). Ontologies, communication syntax and service oriented architecture are seen as the key to enable interoperability. Authoring tasks must be supported and facilitated to increase the mass of interoperable material. Authoring regroups three tasks: authoring of the content, of the instructional process and of the personalization. It largely encompasses what we previously called annotation.

An original position is proposed in (Naeve, 01). In this work, “metadata” or annotations are seen as subjective views of resources. The distributed aspect of the semantic web allows systems to combine several points of view and get better semantic search results than with traditional fixed metadata schemas.

Globally, all the above contributions give very broad overviews of possible use of the semantic web. They stress the positive existing efforts, mostly standardization ones, but fail to address the real problem of the practical realization of their ideas. Major limitations such as creating annotations, sharing ontologies, or managing evolution are crucial problems not answered yet. In the next section, we now take a closer look at existing implementations and the potential solution they offer. Then we discuss specific problems one by one.

3.4.2. Inspiring projects

Several projects have led to effective implementations, we present four of them here. Only two of them actually rely explicitly on semantic web technologies but they all deal with using semantic representations of knowledge for organizing and navigating courses in a web environment.

3.4.2.1. Karina/Sibyl

(Crampes et al., 00) describe a pioneer work using conceptual graphs. It exploits ontologies to offer conceptual navigation in a closed set of annotated resources (Karina). First resources are indexed using a vocabulary and an ontology. Then, according to a user’s profile, relevant resources from a pool of annotated materials are selected and ordered relying on prerequisite relationships with intermediate concepts. A web interface allows learners to navigate this course structure. Annotations are represented in simplified conceptual graphs and compatibility with RDF is mentioned but not achieved. The mechanism also includes the functionality of weighting graphs to differentiate important statements from others.

In this system, two strategies can be implemented:

- Backward conceptual course construction: given a course objective, expressed with weighted statements, and a learner profile, a resource is selected based on the proximity of its description with the objective. The objectives fulfilled by the resource are removed from the objective set and added to the learner profile. Then this process iterates until the objectives are empty. The succession of resources then constitutes an adapted course.
- Forward conceptual navigation exploits conceptual relations to offer free ontology guided navigation.

In a different application (Sybil), navigation exploits the domain ontology to determine the learning path (conceptual navigation). In addition, a pedagogical ontology is added to order the resources that refer to the same topic (enforcing pedagogical rules like “an explanation must precede an example”).

Finally, the authors acknowledge the difficulty to pedagogically annotate resources and confine them to a single pedagogical role. This contribution illustrates the main possibilities of using ontologies for exploiting learning resources, the Sybil system is further described in (Leidig, 01). Even if the tool does not directly use semantic web technologies, they are mentioned as potential alternatives and taken into account for the functional principles.

3.4.2.2. *COHSE*

The COHSE project (Goble et al., 01) considers that the semantic web realization will be done through hypertext, like the original web. Thus, this project tries to improve existing linking between pages, using semantic web technologies. The term “conceptual hypermedia” is introduced to describe this navigation framework. On a classical web page, terms identifying concepts from a chosen ontology are spotted, following the same idea as the magpie example (see 3.3.4.5). This requires a specific browser, or a proxy between the original web server and the client program. For each concept, a selection of links is proposed, this selection exploits the metadata (annotations) and the ontology to find the potential resources that can be linked to the concept. The query is adjusted, for example by querying narrower concepts if the number of results is too high. COHSE internally relies on OIL, one of the ancestors of OWL, and may take any web page as a starting point. It has been tested on the Java Sun tutorial (Java, 06) with a very complete domain ontology about JAVA. The annotations on web pages are manually generated using a dedicated side bar in the Mozilla browser.

3.4.2.3. *Custom Course System*

A realistic example of the use of learning objects is presented in (Farell et al., 04). Custom courses are built from a base of annotated LOs. The LOs originate from a collection of books from IBM, which have been previously translated into DocBook XML. Learners connect to the system and can generate lessons on technical topics. LOs are retrieved regarding their pertinence with the topic. A path is then built among them depending on the experience necessary and time allowed for the course.

The homogeneity of the base is very high in this example. It shows the necessity to work with coherent content to perform such personalized course construction. One of the major interests of the system is that it has been used effectively for internal training at IBM.

3.4.2.4. *Personal Reader*

The Java tutorial from Sun (Java, 06) is a very common example of a learning resource freely available on the web. It has also been chosen as the experimental ground for the personal reader tool (Henze, 05). Unlike the previous examples, this tool does not build dedicated web pages nor modify them. It appears as a separated window, advising the user on additional navigation links, while he/she browses a limited web space like the Java tutorial. Links are recommended based on contextual and personal information.

The current page indicates a topic of interest. For this topic and according to a representation of user’s knowledge, resources are proposed both inside the navigated corpus and outside (directing to the API or the FAQ). The resource selection relies on an implementation of the TRIPLE (Triple, 06) rule based query language running on an RDF base of metadata and an RDFS ontology about Java.

Potentially it can use external semantic web annotations through the Edutella peer-to-peer network. Matching mechanisms can also be defined to use external ontologies. The semantic

relations used to annotate the tutorial page indicate the type and the subject of each page along with classical metadata like title and description. The Dublin Core standard is used in the RDF expression of those relations.

Figure 25 shows an example of a rule used to determine recommended resources: “a learning resource LO in the local context [...] (is) recommended if the learner studied at least one more general learning resource (UpperLevelLO)” (Henze, 05). In the following TRIPLE rule LO is a variable for any resources (Learning Objects) and U identifies the user.

1. FORALL LO, U learning_state(LO, U, recommended) <-
2. EXISTS UpperLevelLO (upperlevel(LO, UpperLevelLO) AND
3. p_obs(UpperLevelLO, U, Learned))

Figure 25 – Example of a rule in TRIPLE

Unfortunately this work does not consider the generation of annotation, or the specification of matching rules for external ontologies.

3.4.2.5. TM4L

Based on the Topic map formalism the TM4L project (Dicheva and Dichev, 06) exploits topic maps to foster exploration practices in learner tasks. The creation of the topic map is supported by a standalone tool. A viewer and a web portal offer to navigate the topic map and access the attached resources. This experiment, quite inline with other experiments on Topic Maps (see (Abel et al., 04), shows that Topic Maps are essentially used as navigation structures without dynamical aspects, but naturally supports “conceptual navigation”.

3.5. Conclusion

3.5.1. Open Vs Closed worlds

In the original vision of the semantic web (Berners-Lee et al., 01) the resource space is virtually open, like the web. Anyone, anywhere, can make his/her own digital information accessible. This is the “open world” assumption. On the contrary, the “closed world” assumption is characterized by a finite set of resources known in totality and cannot be changed without going through a specific process. This is what typically happens for pedagogical resources used in ITS, and in most of the examples presented in this chapter. Even when applications use external web pages, they rely on a closed knowledge base.

Authoring time in “closed world” is important because at the start everything must be created from scratch. Each time a new domain is addressed new resources must be created to populate a new base. Opening the system brings the possibility to reuse and exchange content and knowledge. The authoring efforts would benefit many others, and the cost of setting up the systems would then decrease.

However, working in the open world is quite problematic. The quality and coherence of the material cannot be guaranteed. On the web, resources may change without notice. If ontologies and annotations are also distributed and opened, problems with coherence arise. This open/closed trade-off is at the heart of the novelty brought by the semantic web idea. It also constitutes one of its major challenges.

In the following we only address closed set of resources and consider the open

3.5.2. Standards

Standardization bodies for e-learning did not directly consider taking the turn of the semantic web. Still, important standards like Dublin Core inscribe themselves in the semantic web

initiative by using RDF as their language for expressing the standard. The Dublin Core relations are now used almost in every semantic web system we reviewed.

Standardization for LOM using RDF has also been proposed (Nilsson et al., 03) but this is still to be recognized as a standard.

Globally semantic web standards have little impact on the e-learning community despite the number of issues shared by both communities. However, the power of existing frameworks (see section on tools) should increase the number of projects switching to this representation formalism.

In the following, we shall focus only on semantic web standards and will not consider e-learning standards because they do not answer the problems we want to address.

3.5.3. Reusing resources

The research efforts described in this chapter ultimately target the reuse of distributed sources of pedagogical information (web courses, pedagogical annotations, learning strategies, etc.). This implies to consider to what extent those sources of information may be interoperable.

The syntactical level is now handled by XML quite naturally. We can safely argue that authoring formats are converging towards this unified syntax (XHTML, Microsoft Word, Open Office suite, etc.). However, semantic interoperability is also necessary. As shown in this chapter it may be obtained through the definition and use of ontologies.

RDF/S and OWL provide the means to reach this interoperability with common knowledge representation formalisms. However, experiments shows the difficulties humans have to create coherent and interoperable annotations. The problem of formalizing, extracting and collecting knowledge from documents and experts remains an important challenge.

Learning Object repositories build over the past years (ARIADNE, MERLOT, etc.) also show the unavoidable heterogeneity of the content. If technical solutions might solve the problem up to a certain point (Zimmermann et al., 06), coherence and context are crucial aspects that must be carefully understood when trying to reuse content.

Surprisingly enough, lots of contribution presented in this chapter by-pass this problem, except those coming from the topic maps community, where indexing issues are given a much more importance. We will particularly focus on this issue in the following.

Other issues like evolution, confidence and trust, scalability, etc. are problems the semantic web has still to address and that we do not mention here, as they are quite out of the scope of the work presented in the following.

4. DESIGN AND OVERALL DESCRIPTION.....	65
4.1. <i>Analysis framework for the design approach.....</i>	66
4.2. <i>Scenarios for learning with web resources</i>	67
4.2.1. Motivations for using and creating web resources	67
4.2.2. Reusing digital content	68
4.3. <i>Question Based Learning Strategy.....</i>	69
4.3.1. Addressed problems	69
4.3.2. Strategies based on questions	69
4.3.3. Theories and contributions justifying the strategy	70
4.4. <i>Using semantic representation in QBLS</i>	71
4.4.1. Enhancing the browsing of course content.....	71
4.4.2. Conceptual navigation	73
4.4.3. Semantic representation for knowledge sharing	76
4.4.4. Conclusion.....	77
4.5. <i>QBLS Overview.....</i>	77
4.5.1. A system build around three components	78
4.5.2. System Description.....	78

4. DESIGN AND OVERALL DESCRIPTION

We detail in this chapter the original choices and design characteristics of the system at the heart of this work. As mentioned in the beginning of the manuscript, we want to target real-world application and practical problems. Up to now, the domain of semantic web for e-learning has been the subject of much theoretical contributions. We shall rely on them or discuss them as we describe in the following chapters the details and outcomes of the development and use of our system.

In this chapter, and in the following ones, the references to existing “learning systems” focuses on the area of on-line course consultation systems that make use either of semantic web technology or at least rely on close paradigms to help learners in their learning task.

On the practical side, research on the semantic web benefits from the standards and recommendations published by W3C. Several generic implementations for RDF are available (see 3.3.4.4) giving us solid back ends on which to build real applications.

Still, practical examples using semantic web in the domain of e-learning rarely get pass the “toy-example” level, or are only designed as experimental tools (see the “inspiring projects” in 3.4.2). The integration of such tools in real day-to-day situations is a question often eluded. Compared to other domain of application of semantic web (knowledge management, etc.), this is especially true in the e-learning field because experiments are easy to realize but changes of the practices in educational organizations are difficult. The design of the system we are proposing, without offering specific answers for this difficult problem, always consider practical issues in its choices.

First, we present the analysis directions that allow us to structure the presentation of our design choices. The real-world scenario in which the study takes place is exposed in the second section. It precedes the introduction of the chosen learning strategy and its pedagogical foundations. The presentation then largely focuses on the use of semantic representations for e-learning courses. Existing navigation paradigms that have guided the design of our system in this domain are detailed in the forth section. Finally, the last section introduces the architecture of this system.

4.1. Analysis framework for the design approach

The analysis grid proposed by (Tchounikine et al., 04) gives dimensions for research in the domain of the design of e-learning systems. Analysis dimensions are divided in four groups:

- The definition of the research project,
- The theoretical framework,
- The results of the research,
- The life cycle of the research.

[A] Definition of the research project [A.1] Research objectives [A.2] Constraints on the creation of the learning environment [A.3] Goals of the artefact [A.4] Actors implied in the design [A.5] Social grounding	[B] Theoretical framework of the research work [B.1] Reference to the knowledge being taught [B.2] Refereed theories [B.3] Relation between theory and the targeted problem [B.4] Role played by theory in the design [B.5] Ways of referring to theory in the design [B.6] Design Theory
[C] Research results [C.1] Nature of results [C.2] Generic aspects and generality of results [C.3] Validation types of the results [C.4] Analysis of research results [C.5] Research impact	[D] Research life-cycle [D.1] Initial context of the research [D.2] History of the research

Figure 26 – Analysis dimensions for conception work in e-learning, after(Tchounikine et al., 04)

Without strictly following this framework, we try to inform the reader on each of the specific dimensions mentioned on figure 26.

The context and history of the project [D.1][D.2] have been presented in chapter 2 of the thesis. In a nutshell, the project aims at providing a working example of a learning system based on semantic web technologies. Doing this we want to demonstrate the interest of the “corporate semantic web” approach for learning with on-line courses [A.1]. This should be a contribution to a better understanding [A.3] of the benefits and drawbacks of latest knowledge management techniques to the field of e-learning.

We consider the specific learning situation [A.2] where students use an on-line course. The pedagogical approach [B.2] is based on questioning learners. The system investigates the use of knowledge representation for learning [B.1][B.3][B.4] with on-line courses. Results of the design and implementation of the system [C.x] will be discussed specifically in the following chapters.

For the design theory [B.6], an engineering method: MISA (Paquette et al., 97), proposes to codify the development of pedagogical applications. It applies cognitive sciences principles to the domain of pedagogical design. In the context of our study, the situation is somewhat different as our engineering and design work is pursued in a research perspective. Still we can identify the relevant aspects of this method in the context of our work. Especially this method identifies four axes that we tend to follow in the remaining sections of this chapter.

- *The media involved.* In 4.2, we extend this concept to the description of the learning situation where web resources are used. The “media” itself is the web.
- *The pedagogical conception.* The pedagogical strategy adopted is presented and discussed in 4.3. Even if we do not present a research on didactics, the importance of the pedagogical strategy makes it an unavoidable design parameter.
- *The knowledge model.* The MISA method supposes the explicit representation of the knowledge involved in the design process. We present the principles of using formalized knowledge (in particular ontologies and annotations) to exploit pedagogical resources in 4.4.
- *The delivery aspects of the system.* The global architecture allowing on-line access to courses is detailed in 4.5, along with the different services of the system.

While covering those different facets, the following sections give an overview of the design choices and addressed problems.

4.2. Scenarios for learning with web resources

Generally speaking, our project inscribes itself into the recent practices of using digital courses for education. This does not only concern distance education but also classical academic teaching where computers will soon take the lion share among learning tools. However, the motivations and goals for teachers to adopt this practice are heterogeneous. In particular, we focus on the use of computers for transmitting course material to the students.

4.2.1. Motivations for using and creating web resources

Without considering ITS and AH systems, today the web offers a lot of pedagogical material accessible under the form of HTML pages, downloadable PDF files or slides shows. Depending on the initial motivation for putting those resources on-line, the quality of the material and its suitability for reuse will vary. We present the main use-cases where learning material is made available through the web:

- In a first situation, authors of material (teachers) just want to distribute their course to their students. Putting the material on-line reveals to be the most convenient way, provided the students have sufficient access to networked computers. For example, this type of communication is very interesting when direct handing of paper handouts is made difficult by the “physical” situation, like in distance learning, during strikes, etc.
Rather than using complex course management systems, with restricted access and monitored by administrative staff. Teachers communicate material to students through the worldwide web. Student can connect to a given URL through any browser and download the available material provided by the teacher. To avoid any right access problem, the page is often made visible to the whole web. Hence, Availability on the web does not mean that the material has been specifically authored for this purpose. It might be a side-effect of new practices of communication in education, involving more and more nomadic and digital support. The content meaning in this case is contextual, and irrelevant to anyone outside the targeted school.
- In a different perspective, teachers, often personally attracted by the web, freely author web pages for anyone interested in their matter. Courses are then created to be directly viewed in the browser (HTML) and the necessary contextual information is given to the reader. Motivations for this effort are multiple: philanthropy, desire of recognition as a reference in the matter, university policy, etc.
- In between those two extremes, most of the material is often put on the web, “just in case” someone might be interested, but such material remains in the original form and context for which it was created in the first place.

4.2.2. Reusing digital content

4.2.2.1. Issues with reusing content

Given the amount of material made available on the web, the idea has emerged in the e-learning community to reuse it, to save time and energy. However, many issues are not yet solved to perform such a reuse:

➤ First, appropriate search tools are required to find the relevant resources on the web for a given learning context. Common text based search engines give quite poor results compared to the amount of material existing.

Repositories like Ariadne, Merlot, EducaNext, etc. propose to store pedagogical material along with metadata facilitating search. However, the creation of such metadata revealed quite problematic

➤ Finding a resource and retrieving its content is only one aspect of the problem. Content obtained from the web or through a repository, needs to be re-contextualized, tailored, etc to fit a specific course objective. For the moment, this task is left to the sole responsibility of the teacher, without any particular support. Some research approaches address the problem of delivery of web resources to end-users or learners in context. They take a different orientation by considering that Learning Objects are available at a finer grain than complete documents as found today on the web or in repositories. They also assume that “someone” already annotated them with sufficient information to enable an intelligent construction or navigation of the course (ex (Crampes et al., 00) (Henze, 05)). Both questions of generation of the resources and annotation are mostly eluded.

➤ The registration of a course is often made on a voluntary basis, but it nonetheless has a cost. Economic models might eventually be necessary, see (Quemada and Simon, 03).

4.2.2.2. Reuse scenario

There is a missing link between repository approach, offering material to reuse but no means to exploit it and the “intelligent” systems, exploiting material but not reusing much. The question of how to “migrate” the content from existing material found on the web to a working on-line course system dedicated to a specific teaching is still largely open as illustrated figure 27.



Figure 27 – From repositories to intelligent systems

In this scope, the realistic scenario we want to address is the *reuse of a course found on the web, adapted using external knowledge and used for learning by a real class*.

We believe that scenarios around course content exploitation, whether in the academic domain or the professional environment, have not given entire satisfaction yet. If many theoretical proposals exist, their application in real settings is still limited to small experiments. Our application scenario will then consider the situation of an on-line course where students access the course through a dynamic web interface.

In this scope, we contacted several teachers of Nice Sophia Antipolis University that agreed to take part in the realization of this scenario. Two different attempts are described in the following. In the first one, the teacher reuses a course he authored for the previous year session, on introduction to signal analysis. For the second experiment, the teacher reused a course found on the web and introducing Java programming. In both cases the goal was to provide on-line resources to help students performing labs.

4.3. Question Based Learning Strategy

Prior to exploiting a resource in a hypertext system, the question of the pedagogical strategy must be raised. Our work does not aim at discussing pedagogical concerns about a specific strategy, but “*one of the most important tasks of the designer is to decide on the intended activities that learners will undertake as they use the resources*” (Dalgarno, 98). This is even more important when dealing with digital artifacts as the teacher will not present and comment the material himself/herself as he/she would for a slide-show. Learners will access it through a computer interface, and such interface cannot reproduce the contextualization and adaptation brought by the teacher’s discourse.

Learning systems already possess a rather strict embedded strategy. That strategy must fit the use of the material. In most cases, it must be authored on purpose. When reusing a resource, the pedagogical strategy that will be applied using the content must be carefully defined. It must offer both an interesting contextualization for the material and, at the same time, fit the existing presentation.

4.3.1. Addressed problems

To define a correct strategy we looked at the specificities of web based courses displayed as hypertexts. One of the identified problems of using an on-line course is that learners sometimes get “lost in the hyperspace” (Brusilovsky, 03) and loose focus and motivation to further read the course. The non linear nature of hypertext is seen as a great benefit and improvement compared to books because it offers freedom in the navigation. However, too much freedom leads to disorientation. Freedom is interesting if learners have an activity close to what hypertext has been defined for, i.e. navigating to find information.

In this scope, we defined the “QBLS”, standing for *Question Based Learning Strategy*. The rationale behind it is that students are being motivated by practical questions they have to answer. As explained in (Schneider et al., 03), learners must have « learning tasks » in order to be motivated. In QBLS, the task is to use the course as a knowledge base where to look for answers.

Apart from motivation, the questioning makes learner active in a constructivist vision. Just reading through a course is not enough. Some activities must be performed to help the “construction” of knowledge. Such consideration applies for any static content. Reading a course from a book never made learners fully competent in the domain; there is no reason why courses in digital format would do it.

4.3.2. Strategies based on questions

We envision two types of questioning, supporting two different learning activities:

➤ First, the motivation can be introduced by presenting the domain through large practical problems addressed by the course. For example in the signal analysis domain, the learner could be asked how to digitalize a sound. This is a “thematic” questioning. It may serve as a canvas for lectures, or personal reading. Such questions rely more on learner’s curiosity and interest, than provide direct motivations.

➤ A second type of identified questioning is to confront the student with a real problem, and ask him to find the answer within the content offered to him/her. The means to answer the question are given in the course. To reinforce the personal involvement, answers may be made available at some time to allow students to compare their results with the given answer. There is no evaluation of the learner. Questions are only used as an artifact to support motivation.

4.3.3. Theories and contributions justifying the strategy

In first approximation, the two types of questioning match two different learning paradigms.

➤ The first approach can be considered as cognitivist, the objective being the discovery and acquisition of the principal notions of the domain.

➤ The second questioning is closer to constructivists' ideas, by letting learners experience and answer by themselves.

However, this distinction is not clear. As explained by (Allert, 04), the distinction cognitivism / constructivism does not apply well to such learning situation. We rather refer to an established pedagogical practice, called the ternary rhythm, introduced in (Blanc, 01) as a key for pedagogical success in this situation. According to this rhythm, training must respect a three fold structure:

(1) Creating the need for knowledge (the heuristic step).

(2) Providing the required information (the demonstration step).

(3) Exploiting and assimilating the received information (the application step).

By providing a set of well defined questions at the beginning of the learning path, the heuristic step is realized. It triggers a new need for knowledge and offers a goal to pursue while navigating the hypertext. The extent of the question must be sufficient to guarantee a good coverage of the material. The demonstration step is covered by hypertext navigation and the application step by the activity of answering the actual question.

The QBLS teaching strategy thus falls into the category of Problem Based Learning, or PBL (Savin-Baden, 00) (Barrows, 86) (He, 02), as it makes the learners more active and lets them discover and evaluate the useful information. Doing this, they create their own models and conceptual representations. However, the analogy is not complete. In the case of QBLS, questions are well defined and clearly describe problems, contrarily to "ill formed problem" of PBL. The necessary resources are provided as well as the expected answers. This strategy does not emphasize the problem solving activity but rather the "active reading" one. Motivation appears in this strategy because learners observe the gap between their personal conceptualization and what would lead to the answer. This gap is filled by browsing the course and finding the appropriate information.

We can also compare our QBLS strategy with the proposition of (Naeve, 99) who proposes the QBL, or Question Based Learning. QBL is seen as good motivator and as an alternative to either lecture based learning or problem based learning. The major difference is that questions originate from students, a very ambitious goal, whereas we limit ourselves to the questions that might interest learners according to the teacher.

The QBLS strategy does not constitute a research proposal in itself. It was chosen to answer classical problems faced with hypertext course (mainly motivation, disorientation) and despite its obvious interest, other paradigms might be envisioned (PBL, participative learning, etc.). What we want to emphasize is the key role played by the chosen strategy as a design choice, and thus the importance of defining one first (like mentioned in the MISA method).

The following section deals with our design principles for navigating the reused on-line course, especially by defining semantic representations of the course. Please note that from

now on the QBLS acronym standing for the strategy, will also stand for “Question Based Learning System”.

4.4. Using semantic representation in QBLS

The QBLS system is primarily developed to investigate the possible realization of navigation in on-line courses using semantic information. On the technical side, we propose to use semantic web technologies for this.

We present in this section the existing principles and models that drove the design choices of QBLS. We particularly focus on our choice of semantic representations for guiding learners in the course.

From a methodological point of view, this section roughly matches the phase of knowledge models design mentioned in the MISA method.

4.4.1. Enhancing the browsing of course content

In QBLS, the course is browsed in a standard web browser. There are several ways to browse a digital course, depending on the nature of the course and on the tool used to perform the navigation. For example, if the course is in PDF format, with a table of content, navigation using a PDF browser will consist of accessing chapter pages from the table of content and reading one page after the other. If the course is structured in a hypertext document (in HTML for example), a web browser will allow the learner to navigate from one page to the other following the hyperlinks. Learners may also use the browser history, press the backward button, use bookmarks, etc.

4.4.1.1. Limitations of hypertext courses

Due to the actual convergence towards “web” applications, we focus here only on courses in HTML. In the following, we talk indifferently of HTML pages, hypertext and hypermedia, as we only envision hypertexts on the web.

In comparison with books, on-line hypertext documents possess a great number of advantages, like availability and portability, potential richness of the content, freedom in the navigation compared to a linear format, etc. If such characteristics are generic, they are particularly desirable in the learning domain. However, they also present several major drawbacks:

- As already mentioned in 4.3.1, the “lost in hyperspace problem” (Brusilovsky, 03) causes a loss of focus and motivation. Specific pedagogical strategies may answer this problem, but nonetheless hypertext navigation implies a cognitive effort for learners.
- Classical means of browsing a hypertext course are criticized as static and do not provide enough guidance for learners.
- The authoring of hypertexts requires a lot of time. Dedicated editors have been developed (for example (Nanard and Nanard, 03)) but without convincingly reducing the workload in our opinion.

4.4.1.2. From browsing to navigating

When using hypermedia courses, most of the activity consists in browsing and reading (Murray, 03). Apart from motivation and attention issues discussed above, a potential improvement of hypertext would be to help learners better understand the structure and organization of the course. In this sense, we shall now talk only of “navigation” implying a more conscious activity in term of localization than “browsing” or just “surfing” the web.

For traditional educational material like books, the sequencing of the pages takes into account potential pre-requisite knowledge. When navigating a hypertext some paths may not respect

this pre-requisite order. Guidance might be provided by either adaptive techniques or a careful construction of the navigation structure. Without rejecting the interest of dynamic adaptation, we decided to emphasize the second direction, which is much more realistic in term of the amount of work required from experts (see section 3.1.7 on adaptive systems in the literature review).

The work on hyperbooks (Murray, 03) is quite complete on the subject and shows a wide range of potential improvements for navigating a hypertext structure. Hyperbook refers to courses in hypertext format that are enriched with navigation features obtained by adding the following functionalities to the navigation interface:

- annotated table of content, pictorial table of content, search tool
- custom depth control feature (e.g. explain more, next page, return buttons)
- “go to parent” button, “go to next /previous sibling”, “go to page number”, “go to sub-topic”, “go to related page”, etc.
- integrated glossary, links in the page to glossary entries
- annotated history

These functionalities exploit different sources of knowledge about the course to guide navigation. This confirms our first design choice to *use explicit representations of the knowledge involved in the course to understand it better*.

4.4.1.3. Using explicit knowledge for navigation guidance

In e-learning the main goal of using conceptual and semantic representation in courses, is to facilitate understanding. Experimental evidences tend to show that if the knowledge structuring the course is explicit, comprehension and use of the document is facilitated (Murray, 03):

- Such knowledge includes the hierarchical structure in chapters and subchapters.
- The identification of the *learning topics* covered by each document might also improve the readability and usability of the course. Such structure must be defined by a domain expert and it forms a learning resource in itself. The formal knowledge it expresses is just as important as the expertise encoded in the text. Rather than topic, we prefer the more focused term of *learning objective*, which better identifies the goal for formalizing this information. Later on, we also use the term *concept* closer to the formalization of this knowledge in ontologies.
- Other domains might be described as well (pedagogy, curriculum, etc.).

Generally speaking, the knowledge necessary to be formalized is described extensively in the literature. For in-depth discussion on the different aspect of knowledge involved in learning, see the literature review in 3.2.2.3.

4.4.1.4. Different types of navigation

The question of the potential types of navigation that can be offered has already been quite well investigated by works on adaptive hypermedia or adaptive web based systems (see chapter 3). In the domain of “hyperbooks”, three “epistemic forms” of reading may structure the course for (Murray, 03):

- The *narrative* form follows a linear path. It mimes the affordance of classical books.
- In the *network-like* category, navigation is guided by links between documents. Theoretical justification for this builds on the constructivist vision.

➤ The *hierarchical* form follows the topic/subtopic organization of the documents. The benefit of this structure is grounded in Ausubel subsumption theory as already explained in 3.2.2.3.

User evaluations show that these different epistemic forms complement each other's. However, they only highlight one way of exploiting the expression of the underlying knowledge. We show in the next section that more complex navigation spaces can be obtained using an explicit representation of the knowledge contained in the course, leading to the definition of a conceptual navigation”.

4.4.2. Conceptual navigation

4.4.2.1. Definition

In the field of learning systems, the expression “conceptual navigation” is used to describe learner navigation among digital learning resources, guided by a network of concepts (Brusilovsky, 03). Given the scenario, strategy and research focus presented above, the main motivation of the QBLS system is to provide learners with a conceptual navigation in a document, in particular in a reused document taken from the web. We discuss below this idea, which constitutes the *fil rouge* for our research, presented in the next chapters.

➤ The term “conceptual” means that an explicit representation of concepts is involved. Ontologies and associated annotations are convenient to express such conceptual representations.

➤ The “navigation” idea points at the activity of reading/watching/listening content and acting. On the web, actions are often limited to mouse clicks. Those actions lead to “move” virtually from one resource to another.

4.4.2.2. Graph based representations for conceptual navigation

Graphs are commonly used to represent conceptual structures and their link with other artifacts. The graph expression of the RDF language is a perfect example (see 3.3.2.1). Despite its flaws, like arbitrary positioning and the lack of visual expression for rules, the graph vision is quite commonly used to represent the mechanisms at work in conceptual navigation. It is intensively used in existing works, for example in (Brusilovsky, 03) for categorizing systems, and as an authoring tool in (de Bra et al., 03).

Typically, graphs are used to illustrate two types of objects involved in conceptual navigation:

➤ *Conceptual structures*, where nodes are concepts and edges represent semantic relations between the concepts. Ontologies as “specification of a conceptualization” (Gruber, 95) can be visualized as graphs and thus often play the role of conceptual structure for navigation. In adaptive hypermedia (see 3.1.7), the conceptual representations are not generic and thus are different from ontologies. In learning applications in general (Mizoguchi et al., 97), such knowledge is separated in two broad dimensions: The *domain knowledge* presenting the concepts that need to be learned, and the *pedagogical knowledge* expressing what is specific to learning and its associated strategies. Both can be represented as graphs but pedagogical knowledge is often completed by rules, to support logical operations, which are difficult to represent in a graph.

➤ *Hypertexts*, where each node is a resource and edges represent hypertext links between documents. By document, we mean the smallest entity that can be addressed by its URL. A single web page with anchors for example contains several documents (or resources). In this scope, we can identify each resource to a Learning Object. The different paths create the different possible “assembly chains” of LOs (see theory on Learning Object in chapter 3).

The two graphs (conceptual structures and hypertext) can be connected by linking the resources to the concepts through semantic relations, like subject or type relations.

In the following sections, we describe the different ways to organize navigation among resources using ontologies as conceptual representations. Both resources and concepts are represented on the same graph structure. User paths on such a “map” can be drawn. To avoid confusion between the conceptual level (concepts) and the resource level (instances of “physical” objects) we adopt the following conventions:

- ellipses are concepts.
- corned rectangles express resources
- each relation or link has its type indicated by a specific style (dotted, plain, large arrow head, etc.).

This original representation gives us an *abstract representation of the conceptual structure*, where we identify parts of the semantics that are used for navigation.

When appropriate, the distinction between domain and pedagogical knowledge is made. The presented models are designed after internationally published major experiments, implemented using semantic web technology (Henze, 05)(Goble et al., 01)(Dzbor et al., 05).

4.4.2.3. Static domain guided navigation

The most straightforward “conceptual navigation” consists of using the domain ontology axes (subsumption relations) to navigate between the concepts of the ontology. For each concept, relevant resources can be accessed. They are usually linked to the concepts by relations like “subject”. The domain ontology totally structures the navigation and understanding is eased by this explicit domain model. Example of such navigation can be found in the Memora experiment (Abel et al., 04). Figure 28 shows a graph illustrating this mode. A User path, shown in dashed arrows, starts at the root concept of the ontology and follows the semantic relationships back and forth to navigate the structure.

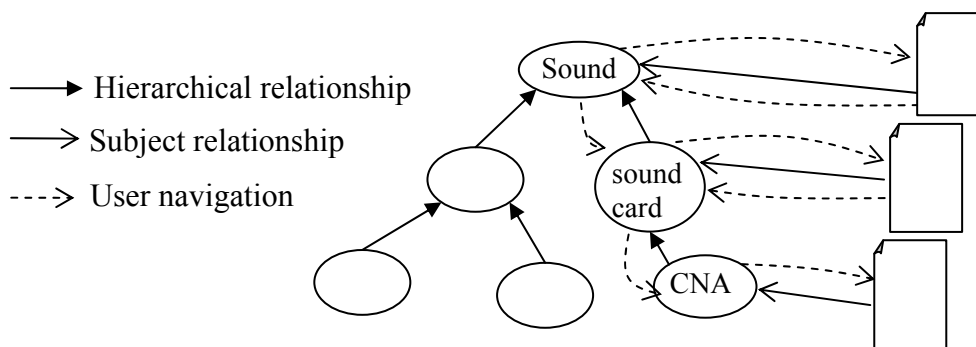


Figure 28 - Navigation model based on domain ontology

Another direct approach consists in using a resource as a starting point instead of a concept. Terms are recognized in the content of the resource, and linked to concepts of a domain ontology. For each identified concept, relevant resources are proposed.

This is what is done in the Magpie tool (Dzbor et al., 05), where terms that identify concepts of a selected ontology are highlighted in the page. For each concept, resources are selected via services, called with the concept id as parameter. Automatic identification of the terms, based on linguistic technologies, suppresses the tedious manual annotation of documents. Pages can be taken directly from the web (for choosing the accessed documents manual annotation remains necessary). The structure of the domain ontology does not play any role in this case. The ontology can be replaced by a list of concepts or a glossary.

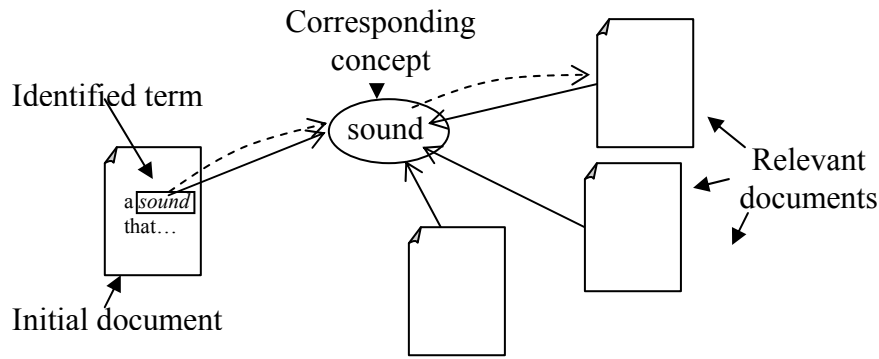


Figure 29 - Navigation through concept identification in documents

4.4.2.4. Navigation based on static ontological reasoning

Logic inferences are powerful means to complete and enhance semantic descriptions. Transitivity and subsumption properties bring ways of exploiting a domain ontology for navigation using logic inferences. For example in the Cohse project (Goble et al., 01) the system uses an ontology and annotations (expressing relations between concepts and documents) to generate new interesting links using the description of the ontology. Typically, resources that are linked to a subconcept of a concept linked to the current resource should be proposed as potential navigation direction. This differs from the first example (see figure 28) as the navigation in the ontology is not explicitly performed but *inferred*. Figure 30 shows a generated hyperlink (thick arrow) between resource *a* and resource *b*. This link was generated because concept “CAN” that describes resource *b* is subsuming concept “Sound” that describes resource *a*.

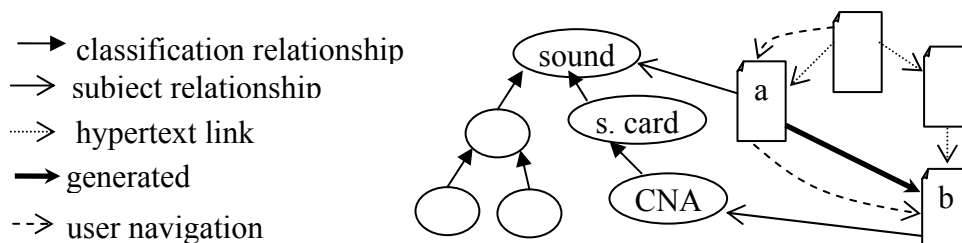


Figure 30 - Link addition based on domain knowledge

Some systems might want to determine exactly the navigation path, to better “guide” the learning activity. In this range of systems, resource selection is performed to build specific/personalized paths. Pedagogical constraints or rules are formalized to generate the complete course automatically. Semantic information involved might include user profile. This is very close to the Learning Object idea of constructing courses from small chunks but we still consider it as a case of hypertext navigation because learners are often given some freedom regarding the static generated path. For example in (Colluci, 05) a course flow creation algorithm is presented, it uses the definition of a domain ontology in OWL-DL to compute an acceptable path to reach some assigned learning goals. Figure 31 shows two generated learning paths respecting the following constraint: a resource relevant for a concept must follow a resource relevant for a subconcept (like resource *b* and *c*).

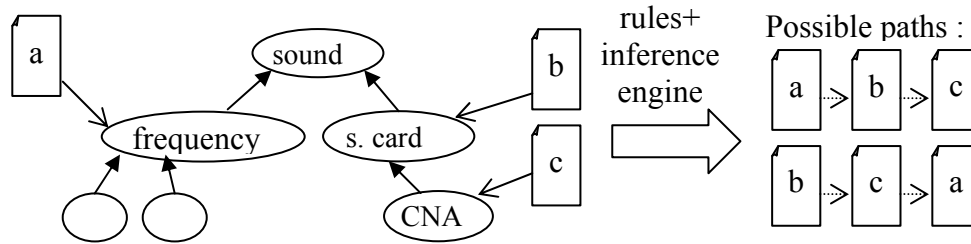


Figure 31 - Path creation using ontological knowledge and reasoning

4.4.2.5. Navigation based on dynamic ontological reasoning

The previous category is planning the whole path at once, whereas “adaptive systems” perform reasoning at each step of the path. They compute and propose navigation directions to guide the learning progression. This is equivalent to add links, as shown on figure 30, but performed dynamically. In the Personal Reader (Henze, 05), the recommended resources are computed depending on the user visits. Rules, interpreted by Triple (Triple, 06), apply to the hyperlink structure, or use the domain ontology and annotations. For example, if a concept is marked as known (because all the relevant resources have been visited) resources relevant for subconcepts become recommended. Figure 32 illustrates it: when a resource *b* is visited, then resource *c* becomes recommended. In term of navigation, this translates to: a hyperlink from *b* to *c* is proposed.

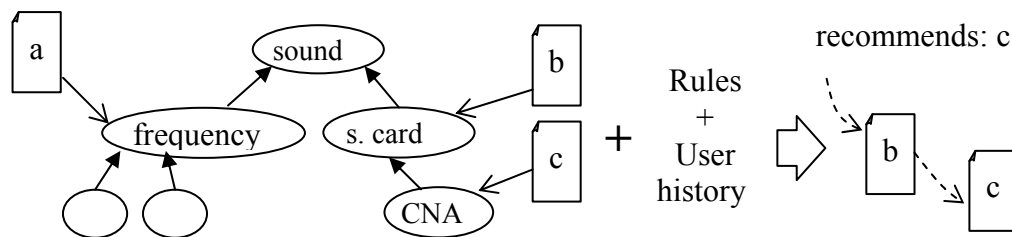


Figure 32 - Dynamic resource recommendation (or linking) using user's path

A closer look at existing use of domain knowledge models shows that the distinction between pedagogy and domain is not clear. A tight coupling exists between both.

- For example, the definition of “requires” relations between domain concepts expresses a learning path among the concepts. This is a form of pedagogical knowledge.
- In addition, pedagogical rules may interpret the subsumption links between concepts of a domain as “requires” links (Henze, 05), following a top down approach in the conceptual navigation.

In the end, the status of the domain classification is unclear as it possesses a pedagogical value and forms an ontology that was built based on this vision. We will discuss this point further when reporting our experience of defining domain knowledge for QBLS (see 5.2.5).

Our design choice for QBLS is to rely on a formal language and a knowledge manipulation tool that will allow us to deploy *all these models in a single navigation system*.

4.4.3. Semantic representation for knowledge sharing

Apart from the objective of facilitating navigation, another objective of semantic representation in general might be to share pedagogical content, representations of objectives, pedagogical models, etc.

The examples of existing Learning Object repositories (LOR) show how standard classifications like ACM computing classification system (ACM, 98) might be interesting to describe resources. Such classifications are restricted kinds of ontologies. They allow us to

envision resource exchange at the scale of the web. However, on a LOR the granularity of the resources is quite large compared to our focus. For the type of resources we want to deal with in QBLS (mostly pages and subparts of a single course), sharing at a worldwide level is far too ambitious and almost pointless.

However, we see a great interest in using semantic representation for sharing knowledge at a smaller, “community” scale. We have observed that teachers often work in small pedagogical teams on the same course at university. Communicating and sharing the course material among this small community is crucial. Knowledge representations like ontologies may be interesting tools to establish a common understanding of the goals and structure of the course.

In the MISA method (Paquette et al., 97) too, common representations are used in the conception phase of a learning system. For example, knowledge modeling is said to improve communication between experts, making it faster and more objective.

In QBLS, we chose to use *ontological representations of the content for supporting navigation*. Community and sharing aspects will not be emphasized. However, such a representation will also help manage the knowledge base, facilitate discussion among teachers, provide an overview, etc.

4.4.4. Conclusion

The review of existing conceptual navigation modes leads us to question the potential genericity of the domain ontologies used in the different examples. By looking at existing experiments, it is clear that either the ontology is being built for the sole purpose of its use with the associated course content (Henze, 05), or that its structure is barely used for navigation (Dzbor et al., 05).

- In the first case, logical relations are interpreted as navigation links (prerequisite, etc) but then the ontology does not really model a domain and may even contain obvious modeling errors (e.g. “sound card” being a subclass of “sound”).
- In the second case, if the ontology is generic, following subsumption links for example might lead to connect resources that do not fit together depending on the way they are written. The structure of generic models might not always indicate the best way to learn.

We think then that two aspects must be considered:

- The identification of the important *concepts* for describing the course. In learning environments, this role is often played by a glossaries, vocabularies, etc., which are unstructured kinds of ontologies. The generic aspect of such representations is high.
- The ontological *structure* that gives additional information by formalizing the relations between concepts. This information is defined at a logical level but it must identify prerequisite, related concepts, etc. It heavily depends on the context.

The following chapters will emphasize the importance of those design choices.

4.5. QBLS Overview

Implementation details of the QBLS system will be presented chapter 6. We give here a broad overview of its functionalities through a general description. This section intends to present the characteristics of QBLS that support the scientific contributions presented in the next three chapters.

The QBLS system is built around a web server hosting a semantic knowledge base. The learning resources and the different interfaces for interacting with users are also managed by the server. Two categories of users are supposed to interact with the system:

- Teachers provide courses, pedagogical expertise and formalize the necessary knowledge.
- Learners use the system, either freely or in the context of labs.

4.5.1. A system build around three components

Three main services are offered by the platform. One for the learners and two for the teachers:

- *Course consultation.* The main functionality of the system is to give on-line access to course content. Learners can connect to a specific URL where they have to authenticate themselves. Dedicated interfaces are provided to navigate the course. Navigation relies on the different models (domain and pedagogy). They help learners in understanding the structure and the organization of the content. Adaptation of the interface is performed based on the learner's profile and navigation history. The content of the course is displayed in web pages and navigation links, dynamically generated by the system, are presented on the same page. An in depth presentation of the implementation using semantic web technologies is given chapter 6.

The course is fully “on-line”, which means learners can access it from anywhere, at anytime through a standard browser and internet connection. Even if several configurations can be thought of, this is primarily designed for interaction of a single user with the system. Each student possesses his/her login and is intended to use the system by him/herself.

- *Course Management.* Teachers can access another interface to manage the different resources offered to learners. Resources can be uploaded as files in a standard format (OpenOffice Writer and Microsoft Word) provided that the files have been previously annotated through a specific method detailed chapter 5. The management of the course also includes the edition of the knowledge base (see chapter 6 for technical details on edition).

- *Log analysis.* Learners' interactions with the system are recorded and presented through a dedicated interface. The teacher can analyze the actual use of the course content using a graph representation of user paths. Statistical measures about resources usage are also provided. The analysis of the generated logs is discussed chapter 7.

4.5.2. System Description

The main part of the system runs remotely on a distant server. In normal use, it does not require the intervention of any technical staff. This is very important in the perspective of using the system in a real environment. The system does not need to be reset during normal use and all teachers' tasks can be performed on-line using the web interface. Thus, a teacher can decide to use QBLS, knowing there is a running server somewhere on the web.

From the learner point of view, the system is purely web based and runs in most web browsers.

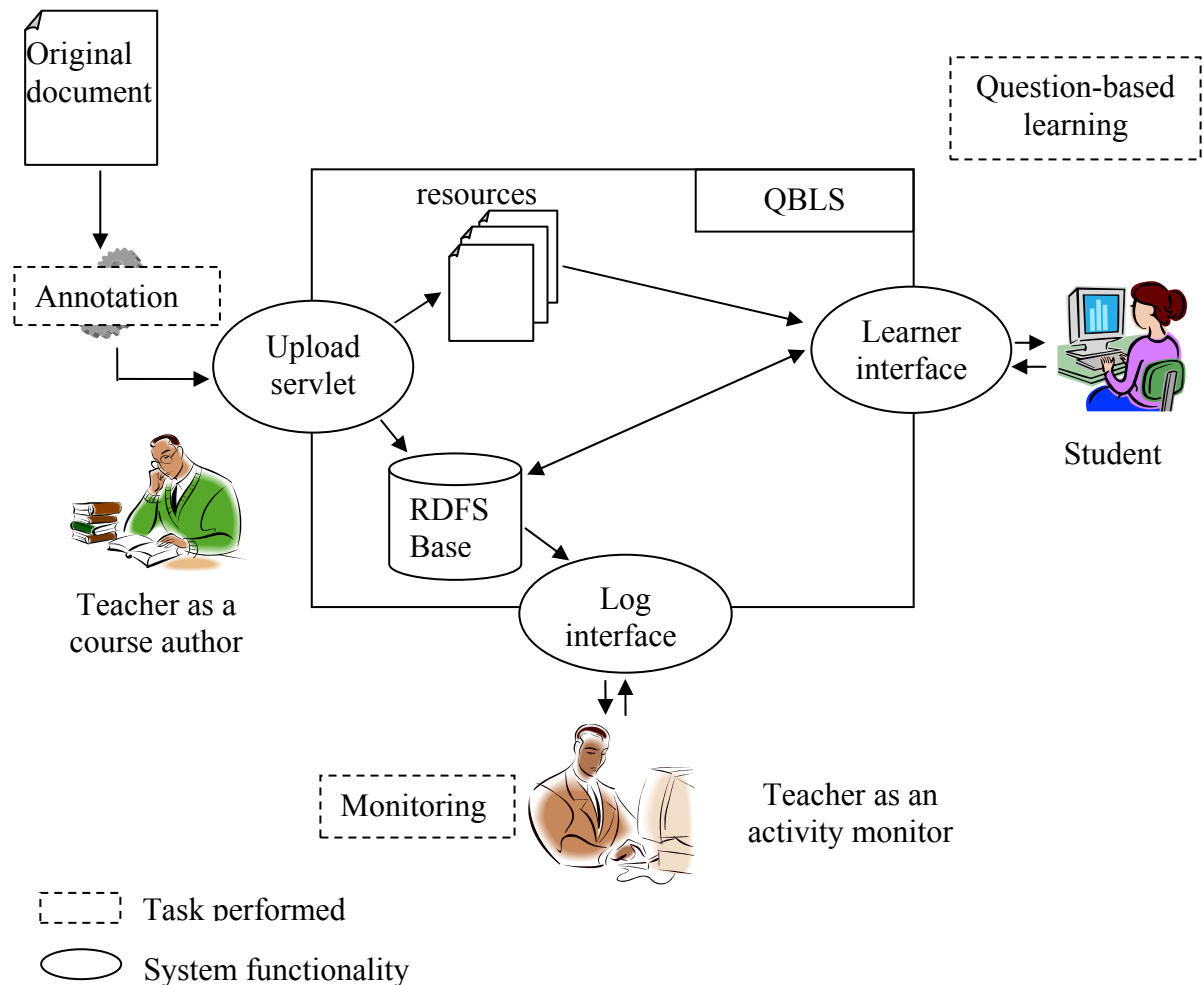


Figure 33 – the global architecture of the QBLS system

The exact nature of this pedagogical tool can be examined according to (Michau and Ploix, 03) classification:

- This is a resource oriented system as it places the use of a course document at the centre of its usage scenario.
- It possesses a high level of organization brought by the formalization of knowledge relative to the course into explicit ontologies. Organization is both structural (navigation is considered inside a structured document) and semantic (abstract concepts are introduced for the domain and the pedagogical aspects).
- The supervising function in the system is ensured by both auto evaluation of students (answers to questions might be checked by students) and analysis of activity traces or logs.
- Communication and programmatic aspects are not developed.

This architecture is very similar to existing approaching realizations (Kunze et al, 02b)(de Bra et al., 03)(Aroyo and Dicheva, 04). However, the monitoring service is less frequent. The originality lies in the internal characteristics of the system, built around a “semantic middleware” (Stojanovic et al., 01) accessing an embedded semantic search engine (Corby et al., 04).

The specific innovations we propose for each functionality are introduced and detailed in the next three chapters.

5. SEMANTIZING COURSE DOCUMENTS 81

<i>5.1. Overview of semantization</i>	82
5.1.1. Existing problems and actual solutions	82
5.1.2. Necessary appropriations by the teacher	83
5.1.3. Separation of content and presentation	84
5.1.4. Characteristics of the semantization task	85
<i>5.2. Proposed semantization method</i>	88
5.2.1. Step 1: Content modeling	90
5.2.2. Step 2: Exploiting semiotic markers linking to semantics	94
5.2.3. Step 3: Automatic annotation	96
5.2.4. Step 4: Manual annotation.....	97
5.2.5. Step 5: Production of formalized knowledge	100
<i>5.3. Evaluation</i>	104
5.3.1. Reusing existing material	104
5.3.2. Qualitative view of the produced annotation	105
5.3.3. Quantitative data on two experiments	106
5.3.4. A graph perspective for domain model	108
5.3.5. Difficulties met.....	111
<i>5.4. Impacts</i>	115
5.4.1. Semantization cycle.....	115
5.4.2. Ontology interoperability	116
5.4.3. Teachers' role and learning technologies	120
5.4.4. Redefining conceptual navigation	121
<i>5.5. Conclusions</i>	123

5. SEMANTIZING COURSE DOCUMENTS

We exposed in the previous chapter how various semantic representations can guide learners through a set of annotated resources. Expected benefits from such techniques are high. However, the problem of expressing the necessary knowledge remains difficult. Strong specificities are linked to the learning context. Difficulties arise both when describing the conceptual model and associating it with documents.

In this chapter, we propose a solution to annotate content with semantic markers directly by teachers. It focuses on the creation of annotations in exiting tools and targets realistic usage scenarios. The proposal covers all the necessary steps from the expression of knowledge and its capture to its formalization in standard machine processable languages.

In the first section, we present the characteristics of annotation for pedagogical documents and formulate the necessity to provide an innovative solution. Then, we describe the solution we propose, called “semantization” for course content, and show its application in a practical annotation method. The third section evaluates the results of this process applied in the context of experiments with the QBLS system. The last section discusses the impact and novelty of the proposal.

5.1. Overview of semantization

5.1.1. Existing problems and actual solutions

5.1.1.1. *Little diffusion of the tools*

To our knowledge, of the five inspiring projects detailed in the literature review (see 3.4.2), none of them has reached a “production” stage, where they would be used outside the context of their original creation.

This should not be interpreted as a lack of interest for modern technologies among teachers because the field of education is very active and is looking for such technological solutions. Research projects often know a limited diffusion of their tools because they were not developed with quality and stability in mind but more as “proof of concept” for innovative ideas. Even when technical quality is met, usability issues (like inadequacy of the tool interface with regards to user qualifications) might also limit its diffusion.

However, such explanations do not fully justify the poor impact of presented e-learning applications in real teaching practice. If the claimed benefits for learning are real, and we believe they are, the problem must have deeper roots that involve the design, and functionality of the tools themselves.

5.1.1.2. *Identified key issues*

According to our literature review and critical analysis, three points have to be considered for the diffusion of an e-learning tool giving on-line access to course material:

➤ *Instructional user independence*: Many experiments involve several roles that must be played by human agents surrounding the teacher. For example, a pedagogical engineer often takes charge of the technical side of the e-learning application, a knowledge expert annotates the content, etc. In regular teaching situations (schools and universities) teachers are often alone to teach a classroom. Eventually a small pedagogical team of teachers deals with a larger group of students. However, on a regular basis the palette of competencies available is restricted compared to what research projects envision.

In companies the situation is somehow different. Experts may appear as technical consultants, but the price of creating a training or a course then increases tremendously. Companies tend to turn towards more stable products than advanced tools.

The recommendation that emerges from these observations is that instructional users (university teachers in our case) must be fairly independent in their interaction with a learning system. This includes course creation as well as deployment processes. This has been understood by systems providing administrative and authoring tools for teachers (de Bra et al., 03)(Aroyo and Dicheva, 04). Nevertheless, this direction needs to be much further developed.

➤ *Content management*: In most adaptive systems, as well as recommended by the MISA method (Paquette et al., 97), the course is built from scratch. This is obviously a good way to ensure coherence between annotation and content but the associated cost of authoring courses specifically is high. Reuse of content as envisioned by the Learning Object approach is implemented in few systems (Farell et al., 04), (Rigaux and Spyrtatos, 03) (Buffa et al., 05). The type of reused resources considered is then very specific (mostly books in specific format). A crucial problem to solve for the diffusion of advanced on-line pedagogical applications is to enable the reuse of existing material by lowering the cost of the necessary “preparation” of the material to include it in a specific system.

➤ *Annotation support.* Manual annotation by filling electronic forms takes a lot of time and is often rejected by practitioners. An important result obtained in the past years is the certainty of the inadequacy of form-based annotation for annotating course material (Brusilovsky, 03). Annotating by filling forms is definitely not acceptable on large corpora. When addressing large realistic amounts of resources like a complete university course, solutions must be found to ease the process.

The alternative “Mark-up” based technique seems more scalable. The term “markup” comes from the use of an underlying “markup” language (typically XML) for storing content. In this approach, annotation is added on the material itself. It solves contextualization problems but many issues subsist. Solutions must be provided to help users generating annotations faster and more intuitively.

Generic annotation tools already exist, and key characteristics for knowledge management annotation tools have been identified (see 3.3.4.2). It has also been observed that generic annotation tools, like (Annotea, 06), are too rich and complex to fit a specific learning scenario. For example, they do not match the first requirement identified by (Uren et al., 05): the integration of the annotation tool “*in the one in which they create, read, share and edit [documents]*”.

The above issues were thought at first to be only economical ones, and there is still a strong belief among the research community, like (Duval et al., 02), that by spending enough time and providing adequate competencies, they would be overcome. We think that some of the problems mentioned have deeper roots (especially the annotation method). In addition, one of the promises of e-learning was to lower the cost of authoring to leave more space for interaction. Then, spending more is not a solution.

The examples of e-learning systems actually in use (LMS or learning platforms) show that the efforts required to set-up the proposed advances go well beyond current practice. In return existing applications offer only simple functionalities.

5.1.2. Necessary appropriations by the teacher

Despite the above difficulties, we believe in the possibility to reuse pedagogical material, annotate it and include it in educational software as proposed in the reviewed examples. For this, we must consider (1) an independent user, (2) the reuse of existing material and (3) an efficient annotation method.

The teacher, that selects and “prepare” the material, must perform a triple appropriation:

➤ *Appropriation of the document.* For the reused documents, an appropriation phase is necessary. First, it targets the adaptation of the content according to the learning goal and the system requirements. It also aims at convincing the teacher of the quality of what he/she is ready to put on-line under his/her responsibility towards learners. This phase is compulsory even if the teacher reuses a course he/she has authored himself because he/she may have changed his/her mind from one year to the other. This appropriation implies the recognition of the pedagogical goal, the domain covered, etc.

➤ *Appropriation of the tools.* The system behavior must be well known. It is important that the teacher trusts the mechanisms at work and that he/she can manage the complete tool on his/her own. No external intervention should be necessary to put the course in place. From the point of view of the teacher, he/she should be as independent as possible in his/her dealings with the system.

➤ *Appropriation of the pedagogical model.* Teachers must accept the pedagogical paradigm proposed by the system (e.g. learning by questioning, navigation through concepts, etc.). The

benefit of this paradigm must be recognized as well. Such acceptance may sound trivial but it reveals crucial and generally speaking, it should take place in the process of using any learning system. When the system relies on an explicit conceptualization, like an ontology for example, the teacher must adhere to the proposed ontology. In this sense, we can speak of “ontological consensus” in the sense of (Gandon, 01) and (Mizoguchi and Bourdeau, 00).

For (Verbert et al., 05) these appropriations are defined by the term “*repurposing*”. It presents similar requirements but envisions the appropriation of resources in an authoring process for new courses, whereas we focus on reuse of complete courses with as little authoring as possible.

Once these preliminary requirements are fulfilled, a method for reusing course documents can be constructed based on a full cooperation of our main actor.

5.1.3. Separation of content and presentation

5.1.3.1. Principle of separation

More and more digital documents are encoded in languages that separate content from presentation. The typical example is the XML language: It does not contain any presentation information, but expresses the content or internal semantics of a document. Various “visual” rendering can be generated from XSL templates from the same source.

From a conceptual point of view, this technology separates two concepts that constitute a document: the content on one part and the presentation on the other part. We also envision this separation as the distinction between the *semantic* level and the *semiotic* level.

In a formatted document the presentation, or layout, informs the reader on the role of the different components (titles, paragraphs, words, etc.). Reader’s interpretation is guided by the rule that if two elements play the same role, then they should look the same. Reasoning the other way round, we conclude that in a well written course, *if two elements look the same, then they should play the same role*.

The semantization idea is to extract semantic information from the semiotic level. It uses visual markers, whose description belongs to the presentation, to derive semantic information about the document. If the markers are used in a coherent way, it is possible to extract information automatically, doing “reverse engineering” on the author’s intention. This strategy is exposed on figure 34.

This strategy relies on the crucial hypothesis that the material offers a coherent presentation regularly expressing the semantic organization. We will discuss this difficulty in 5.3.1.2.

5.1.3.2. Illustration with HTML

The HTML standard simply illustrates the above principle. Some semantic information is included in the markup language. For example, the document hierarchy is expressed using specific tags: H1, H2, H3, etc. In a browser, those tags look differently from one site to the other. Using CSS, for cascading stylesheets, the appearance of the title can be completely customized to reflect this hierarchy visually in the context of the page. Figure 34 shows a typical example of the use of CSS stylesheets. The HTML defines a title and content (in a div tag). The appearance of each of those tags is specified in the CSS. The final result combines both levels of information.

The semantization process tries to do the opposite: from the visual representation, it determines the layout characteristics (or styles) and the semantic structure of a document.

Classic use of content/presentation separation:

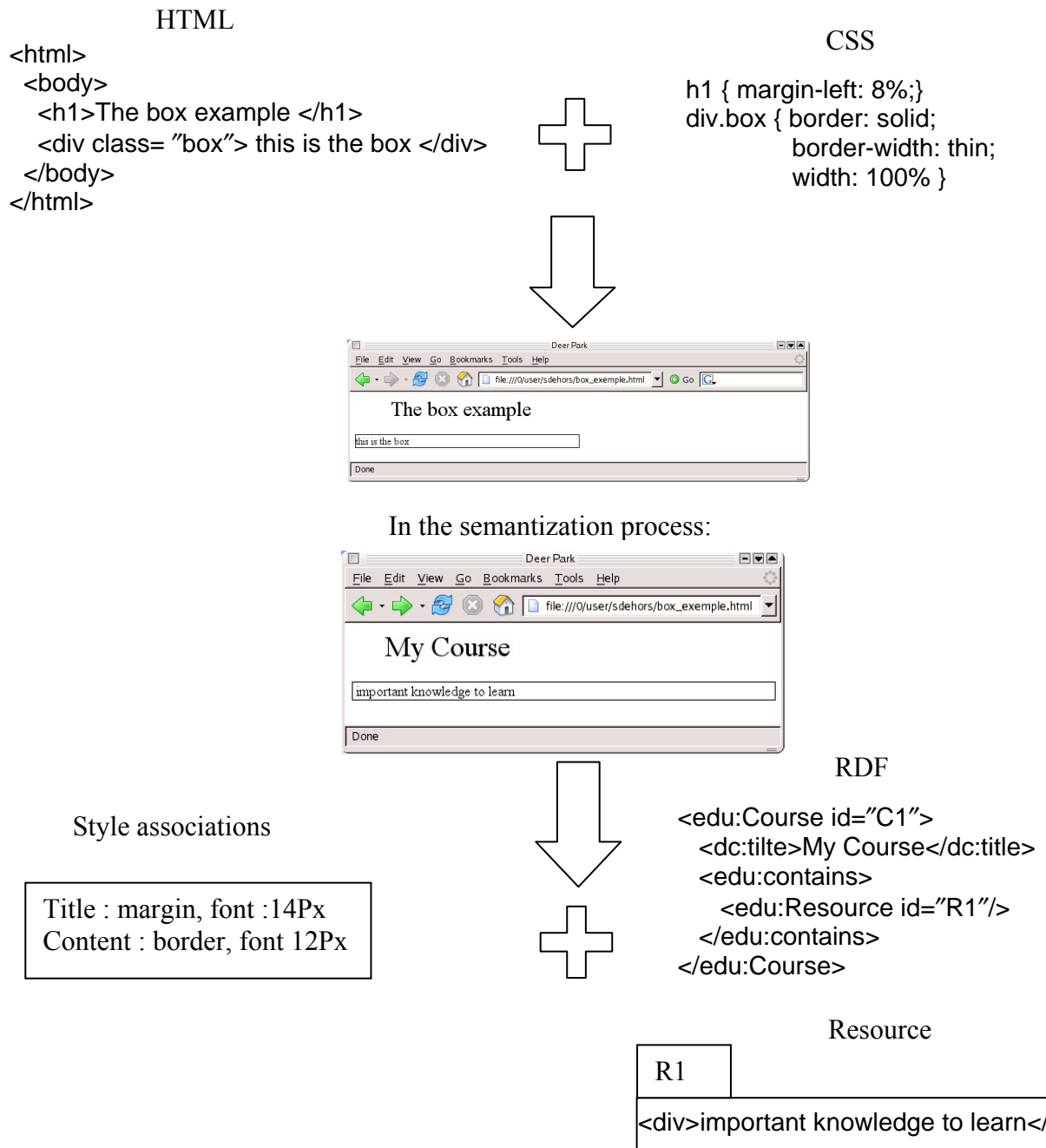


Figure 34 – Dissociation of content and presentation in the semantization process.

5.1.4. Characteristics of the semantization task

In this section, we propose a theoretical description of the semantization method that targets the integration of course material in a learning system. Four aspects are described: the identification of knowledge in the content of course documents, the possible automation, the role of knowledge references like ontologies and the principles driving this task.

5.1.4.1. Knowledge identification

In an e-learning system, several types of knowledge must be distinguished: domain knowledge, pedagogical knowledge, knowledge about the document structure, the user, etc.

The first step of the proposed process is to cut down the course into small components and identify the knowledge they contain. This is very much inspired by the Learning Object approach but in this case the “objects” come from the same source and they will be used in a single system.

The process is called “semantization” because it aims at expressing semantic information. That means information precisely defined in non ambiguous models. The type of semantization we propose, deals with the knowledge that can be associated or extracted from the resources. In particular, this task is based on the identification of the terms found in the text with regard to reference models.

Additional knowledge, compared to what is explicit in the document content, must be expressed and formalized for each object or “resource”. This additional knowledge represents the teacher’s expertise in a classical teaching situation. Its extraction represents the major challenge we are addressing. For example, it aims at characterizing the components of a course with regard to pedagogical models.

The task philosophy is close to the “slicing” idea introduced in (Buffa et al., 05). However, the content is described (or annotated) while considering the surrounding resources. What differentiates the act of semantizing from annotation in general is the final objective. For (Azouaou et al., 03) the action of annotating means linking a document to a concept. Our approach is similar and the produced “annotation object” possesses the same aspects of anchor and attributes. But the semantization aims at making explicit, in a formal way, the pedagogical added value brought by the teacher with the objective of using the resource in a specific system. A semantized document necessarily envisions a pedagogical goal and a targeted system, which is much more restrictive than classical annotation.

The focus is put on access to the material, the teacher first performs the annotation task in the specific goal of offering enhanced access to the content for its students and not for a potential reuse of the resources through large exchange frameworks (like Edutella for example),

5.1.4.2. Definition and use of ontologies

The semantizing process involves the use of reference models: domain, pedagogy, etc. It identifies concepts behind terms in the content of documents. For this, the “conceptual” models are expressed in the form of ontologies.

Reference ontologies may exist before the beginning of the process (they could be reused), but the possibility to update, or even rebuild them, must be taken into account. Each important domain term must find a match in the domain ontology.

On the contrary, content is not intended to be modified. This approach is opposed in this sense to the ones based on the creation of content given a specific ontology. In the proposed method, the ontology should be built or adapted given a specific material.

The idea of building the reference models from scratch, through expert experience (without reference to a specific document) must be rejected in this scenario. The final goal is to put a course on-line, so it is necessary to keep a strong link between the targeted material that will

be visible at the end and the conceptual model. We consider that at least part of the ontology construction is included in the semantizing process.

For the different aspects (domain, pedagogy, document, etc) to be modeled, we rely on the result of previous research, and identify a domain, a pedagogical and a document model. (see 3.2.2.3):

- Domain description is expressed by an ontological representation. The impact of the representation, for example its structure on the subsequent use of the semantized material drives the modeling choices for the domain.
- For the pedagogy, we rely on the hypothesis that every course is based on a learning or pedagogical model, which includes some pedagogical strategy. This strategy can then be represented by concepts and rules of an ontology.
- The document model is defined specifically to match existing content as closely as possible.

We shall see that ontologies offer a large expressivity. Limitations soon come from the teacher's ability to identify and express knowledge and not from the expressivity of this knowledge representation tool.

5.1.4.3. *Potential for automated support*

Document annotation, with regard to the concepts of an ontology can be limited to the identification of the terms used in the content. Annotation consists in associating one or several terms with a concept from the ontology. Concept labels in ontologies identify relevant terms in the content. This kind of annotation is supported by annotation tools, like MnM (Vargas-Vera et al., 02) and may be largely carried out by automatic annotation processes based on linguistic techniques.

In the domain of learning, the application of linguistic techniques raises difficulties. The precision of the annotation must be perfectly accurate. Mismatches are not acceptable in the context of annotations for learning because they would strongly disturb learners' fragile understanding process.

Teachers also place a strong intention dimension in their annotation. They identify knowledge for a specific goal. This is difficult to reproduce with automatic annotation.

These characteristics clearly discard the use of fully automated techniques for annotation. The scenario of semantization requires a large manual intervention, even if some degree of automation can always be introduced.

5.1.4.4. *Driving Principles*

In the remaining, we use the term "annotation" to describe teacher's work on documents, but it must be understood as a part of the global semantizing process.

This process is built around the following principles:

- Bring *coherence* to pedagogical content. This coherence targets a specific pedagogical strategy. It means that the content must support the envisioned strategy. For example in a question based approach, the teacher/annotator must make sure that answers are clearly contained in the material. In addition, any rhetoric formulation linked to linear reading must be removed or adapted. For example, expression like "as explained previously" or "based on the previous definitions" will not make sense in a conceptual reading.
- Make the course structure *explicit*. In a course document, information is often implicit. Making it explicit plays a double role: (1) it favors the reuse of content by the annotator

himself (from one year to the other for example). This is close to building a personal memory. And (2) it allows for different actors to exploit the content, including computer programs.

➤ Ensure the *homogeneity* of content. If resources use different conventions for colors, text styles, notations, etc. annotating and processing them automatically will be difficult. This is the most visible aspect of the process because it changes the appearance of the content. Such standardization is not often supported by annotation tools, which consider it a prerequisite. However, it is compulsory for any automatic exploitation and the cost of this process is not to be underestimated. It is anyway a requirement when learners start using the content.

➤ Use a *convention* for knowledge representation through visual signs (semiotics aspects). The code materialized in the annotation interface by the colors, forms, etc. must be well defined and unambiguous. The objective is to be able to distinguish visually the presence of annotation and potentially their meaning. This includes pedagogical as well as domain knowledge and the more obvious document organization information.

After presenting this theoretic view and its basic principles, we present a detailed method, based on real examples, to bring an existing linear document into some form of organizational memory accessible by a computer system.

5.2. Proposed semantization method

We detail in this section a practical method to semantize digital course documents according to the previous requirements. The starting point or input of the method is a course document found on the web, on a web page or on repositories of LOs. To bring such course document into a reusable form, a number of transformations and additions are necessary. In particular, depending on the targeted system, some additional knowledge must be expressed and formalized to enable a better exploitation of the document content.

Given the problems identified above (5.1.1), this process must be carried out by the teacher as independently as possible. In our basic assumption the task must be *straight-forward* for teachers and must not impose them to use any underlying formalism directly (ex: XML or RDF). Formalisms are necessary for persistence and automatic manipulation, but should not be manipulated by such non-experts. Hence, the method proposed below generates annotations in RDF, while only presenting the main actor, the teacher, with existing largely available tools for editing textual documents.

The proposed method is divided in five main steps:

1. First, the teacher defines the pedagogical strategy for the course and decides on the models (pedagogical, domain, document) that will be instantiated.
 - a. During this phase, the teacher may be interviewed by a “pedagogical engineer.
 - b. He/she may use and adapt existing models.
2. The annotation task consists in identifying, in the layout of the document, the markers of the concepts of these models expressed in ontologies.
3. Once the models are in place in the layout, an automatic extraction mechanism can prepare the manual annotation phase. It takes advantage of the existing defined ontologies to spot corresponding concepts in the content.
4. The teacher manually checks the annotations and ensures that they are coherent all over the document. Doing this he/she constructs the conceptual space that will be navigated by learners
5. The final step consists in formally creating the instances of the concepts identified in the original documents. This is done through a complete automated process through stylesheets (XSL) generated using the ontologies obtained from the previous steps.

Depending on the resources and type of deployment envisioned, some of those steps might be emphasized or skipped. Figure 35 below illustrates the five steps process.

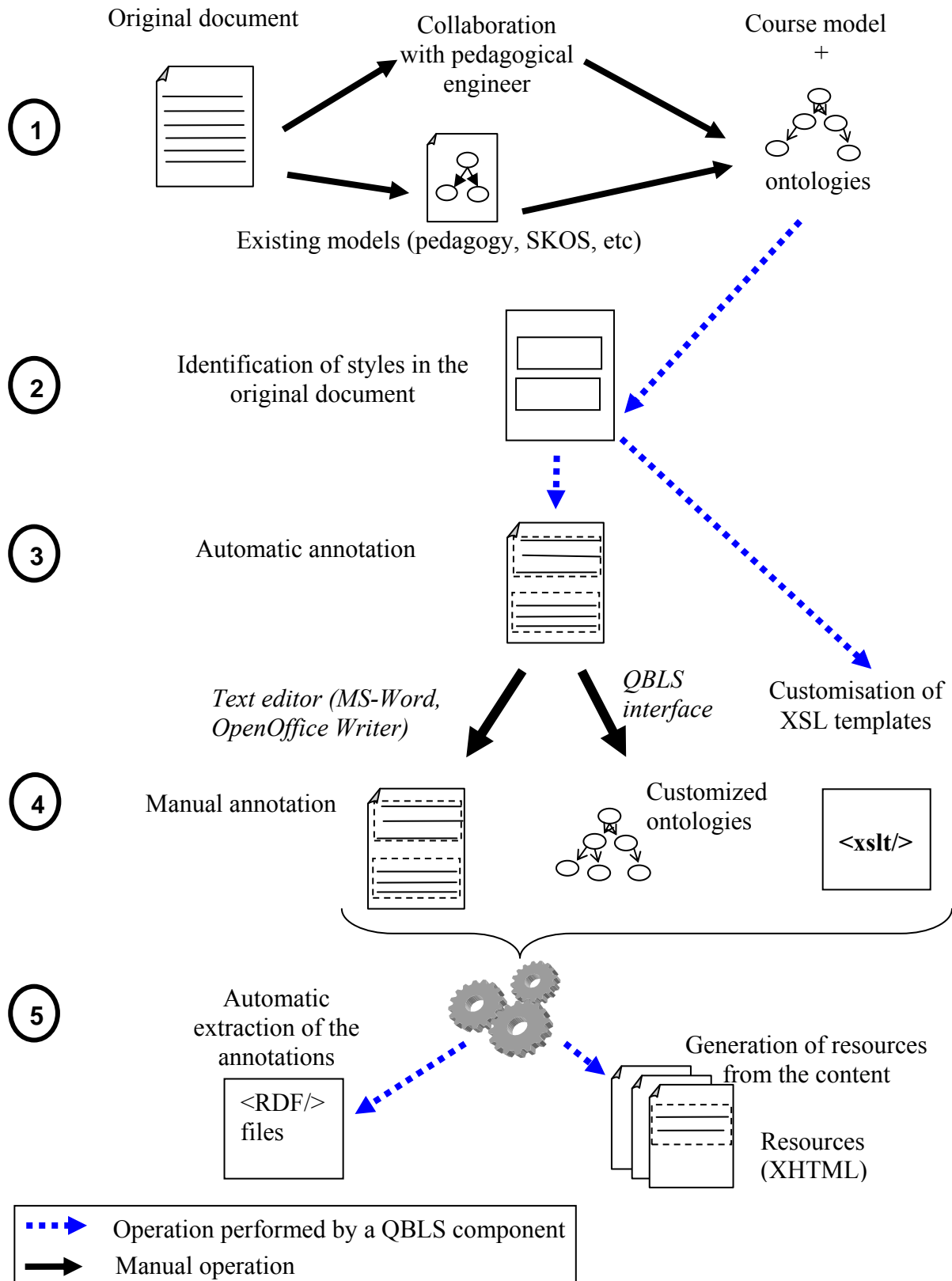


Figure 35 – Overview of the annotation process

This method and its implementation were developed with the QBLS system. It complies with most of the requirements identified by (Uren et al., 05) for semantic annotation processes. However, it does not address the problem of evolution as explained later on (see 5.3.5.1).

5.2.1. Step 1: Content modeling

The method starts with the choice of an existing material. In order to reuse material and thus save the cost of creating new content, all the process must be articulated around this central artifact. In this section, we show how the content can be analyzed and modeled. Illustrations are taken from the experience we conducted on two reused slide presentations:

- A course on signal analysis (later on called QBLS-1) used the previous year by the same teacher as a support for oral teaching and as a hard copy course reference.
- A document supporting a course on Java programming (QBLS-2) titled “Objects First with Java a Practical Introduction using BlueJ” and written by David J. Barnes and Michael Kölling. It targets the use of the BlueJ programming environment to learn Java. Students perform labs with this tool.

For skeptical minds, this method has been applied to several other resources. This is presented in the last chapter (see 9.3.1).

The first step of the method presents two facets: the analysis of the exiting material and the generation of the models. Before this, the very first action of the teacher is to select the material and state the pedagogical strategy of the course. For example in our case study, we focused on a question-based approach motivating learners to read the course (see chapter 4). Then, several modes of interaction with the teacher can be put in place:

- The teacher is interviewed by a “pedagogical engineer”, aware of the extraction mechanism and the system features. It ensures that the objectives are acceptable and establishes the role of the system within the strategy. In accordance with the strategy, they agree on a coherent model of the document, and on the necessary annotations that must appear in the content. As the discussion takes the original document for starting point, the model is somehow hidden but already implicitly present in the document.
- On a day-to-day basis most teachers are alone to prepare their course. Rather than defining a new document model each time, the teacher might be proposed a set of existing models. He/she would choose amongst them the closest one to the reused material. From our experience, when working with slide shows, the range of models is quite restricted. With a bit of experience the teacher may become autonomous in this task, which represents the long term goal of the proposed method.

5.2.1.1. Analyzing existing material

Analyzing the document should help in defining a model of the document that reflects the pedagogical strategy that will be carried out. In our case, this strategy has been described as the “QBLS” strategy. Taking a closer look at the existing document leads to the definition of the different concepts composing the course and the implicit model it is build on.

In our first experiment (QBLS-1), the original document is a unique Microsoft PowerPoint file, supporting one hour of formal lecture. The curriculum objectives are explicitly written at the top of every slide and a set of relevant questions are given at the beginning and during the course to motivate students. Questions help focusing on the curriculum content, and allow students to self-test. Figure 36 shows a slide taken from the original course.

La qualité de numérisation d'un son croît avec f_e et B

Curriculum objective

definition → **contrainte de Shannon : la fréquence d'échantillonnage f_e doit être au moins deux fois la largeur du spectre du signal à échantillonner** ← concept

example →

- le spectre est la composition fréquentielle du signal (voir le chapitre Spectre et TFD)
- par exemple, le signal $s(t) = \cos(2 \cdot 440 \cdot \pi \cdot t)$ contient la seule fréquence 440 Hz (ou LA3 musical) et doit être échantillonné à $f_e = 880$ Hz au moins.
 - Hertz (Hz) : unité de fréquence, comme radian par seconde (rd/s), on a : $1 \text{ rd/s} = 2 \cdot \pi \text{ Hz}$.

question →

- **Application de Shannon :**
 - comment faut-il échantillonner le signal $x(t) = \cos(2 \cdot 880 \cdot \pi \cdot t)$ (LA4) ?
 - L'appareil auditif humain sait percevoir des fréquences allant au mieux de 20Hz à 20kHz. Justifier en conséquence la fréquence d'échantillonnage de 44100Hz des CD audios.
 - Le téléphone grand public transmet sur la bande de fréquence 300 Hz, 3300Hz, proposer une fréquence d'échantillonnage en conséquence

S.S.I., ESSII, le 8 février 2004 Page 1

Figure 36 – Screenshot of the original course for QBLS-1

In a second experiment (QBLS-2), we repeated the process. The questioning approach was kept as underlying strategy. The material itself builds on a very classical scheme: a set of slides separated in chapters. Chapter introductions contain a list of “fundamental concepts” that are further developed in the subsequent slides. The title of the slides often identifies the concepts subjects of the resource (see figure 37).

The original material does not contain any questions. The team of teachers in charge of this course would provide the questions in external web pages authored collaboratively by the teachers (using a wiki). As those pages would be static and students would have to follow a sequential order, there was very little interest to consider them as resources to be included in the system.

Generally speaking, the analysis of the document consists in identifying in the layout of the document the role played by each graphical elements (boxes, titles, colors, etc.). The organization of document in small separable resources with potentially several hierarchical levels (e.g. chapters) must also be identified.

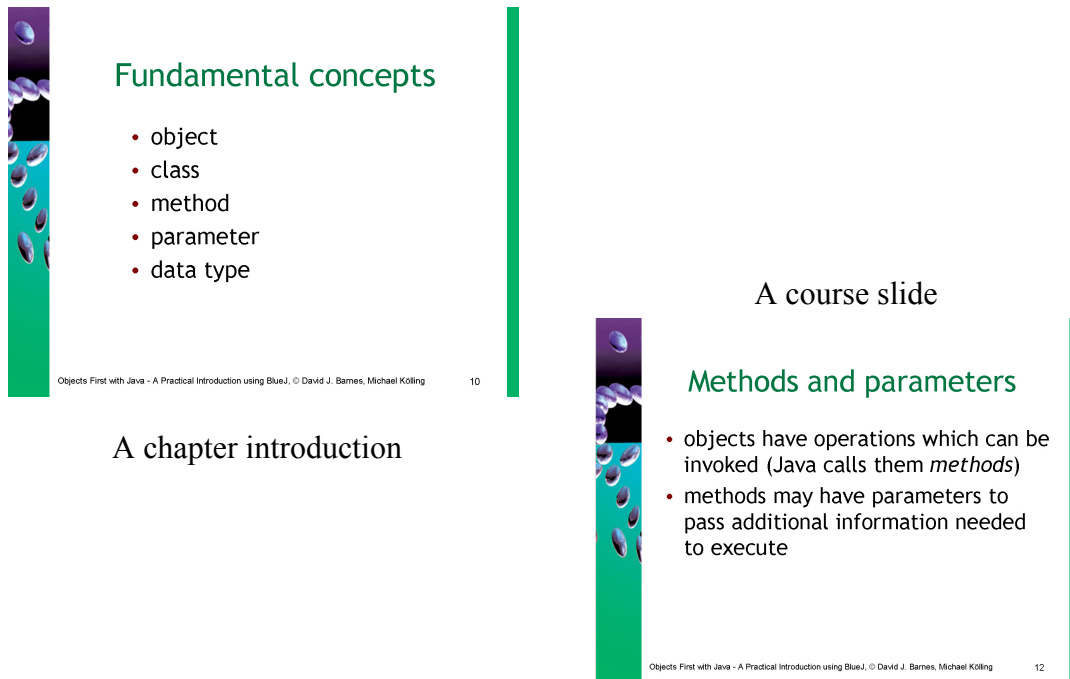


Figure 37 – The original slides of the course on Java programming

5.2.1.2. *Generating models*

A first formalization of the concepts identified above (titles, roles, etc.) must be performed in order to use this model in the following steps. The first ontological representation of the document is then created at this early stage.

We describe below the models we ended up in the experiments:

QBLS-1

On the behalf of the teacher, the final model does not exactly fit the implicit original material as he decided to “upgrade” his strategy and course content in the process of defining them (see figure 40 for the final layout of the course). In the end, the document model shows an aggregation of paragraphs, called cards. Each of these cards contains a title and content, which is the actual text and graphics of the corresponding paragraph.

Each paragraph describes an important concept from one point of view among several possible, e.g. “example”, “definition”, “formalization” or “precision”. For some of the paragraphs this was quite easy to spot because one of those terms was appearing in the title, for example “*Définition du son*” (definition of sound), “*Exemple de carte son*” (sound card example), etc.

On the structural level, the course has two levels of granularity. It is divided in seven parts, each of them containing a certain number of paragraphs. By discussing with the teacher we established that the first paragraph of each part is on a higher level of detail (expressing basics important to know about) than the rest which deals with more specific topics. Three types of concepts express this: “*Course*” is on the highest level and identify the subject of the whole document, “*Theme*” (some key knowledge) is at an intermediate level and stands for the beginning of each major part, finally “*Notion*” (very close to what (Brusilovsky, 03) calls “concept”) represents the specific topics of the domain. Questions, contained in the course, are represented by another specific concept: “*Question*”. The paragraphs attached to the questions are specialized depending on their content: statement, procedural hints and solution.

Physical resources (called here pedagogical resources) are linked to abstract representations through a classical subject relationship. That means each resource has for subject one, (and exactly one in this case) abstract subject. For a review of the possible ways to encode subject relationships using semantic web formalisms, we refer to (Noy, 06).

Concepts may be mentioned in the content of a resource. It expresses a relation from the resource to the concept. However, we could not specify its type (see discussion 5.3.5).

The pedagogical engineer's task consists in formalizing this underlying model that we claim exists in any pedagogical document of reasonable quality. The formalized concepts can then be hierarchically organized in an ontology. Figure 38 shows the main organization of this ontology.

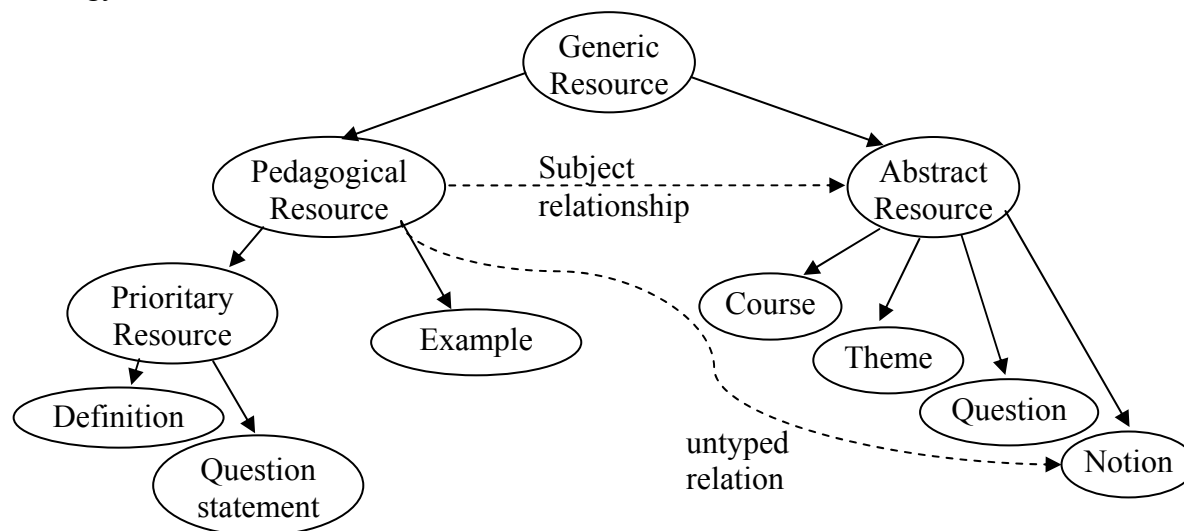


Figure 38 - The QBLS-1 Ontology (excerpt)

This model has no ambition of being generic, as our rationale is to save time and effort for teachers. It is faster and easier for them to define their own small model for documents, than to adopt an existing one as proposed in (Aroyo and Dicheva, 04). If the competency of defining and encoding such model is not available to the teacher, existing models will have to be reused and eventually adapted to fit the original document. The worst case is met when the document needs to be modified to fit a specific model. In this situation, the benefits of reuse are disputable.

QBLS-2

In the second experiment, we tried to reuse existing models as much as possible. We decided to enrich the conceptual vision, compared to the first experiment, by relying on the ontology proposed by (Ullrich, 04). Using an already defined conceptualization is typically what a teacher would do instead of defining his/her own model.

For the definition of the domain concepts, we stuck to the Course/Chapter/Concept distinction. We reused the SKOS meta-model to express and organize the domain concepts and specialized some of them. For example, we defined the concept type “case studies” to fit the specificities of our course model, which explicitly referred to specific case studies in the content of the course.

An excerpt of the final model is presented on figure 39. It is somehow very close to the first one, except that we striped the top concepts that were not useful for the system, based on our previous experience. For example, we observed that having a top concept for the ontology presented little interest because the system would never refer to this concept. The separation illustrates the natural distinction between a pedagogical model (on the left) and a domain

meta-model (on the right). Organizing concepts hierarchically makes the ontology look nice from an abstract point of view, but the interest of the model must be envisioned in its context of use. All the relations used or defined by the model are exploited in practice (subclass relations, title relation, subject, etc.). We also removed the constraint of having a single subject for each resource because this was not compatible with the material.

In figure 39 the left part of the model illustrates the reused pedagogical ontology.

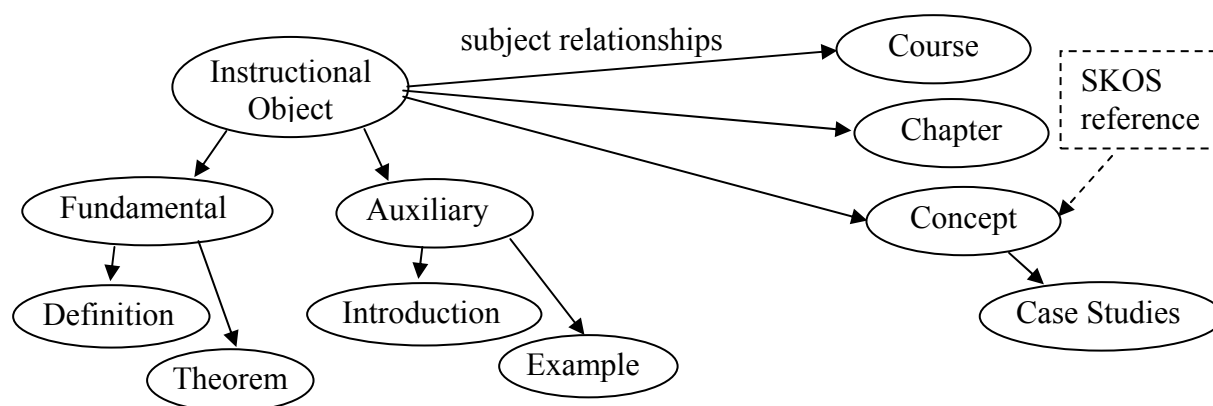


Figure 39 – The QBLS-2 ontology (excerpt)

5.2.1.3. Conclusion

In the end, the definition of the model is inspired by both the existing content and the envisioned scenario of use. This aspect is important for the success of the method. Teachers can only be motivated into annotating if this effort has direct benefit for their local and immediate use of the material. Models must be stripped of any useless conceptualization that would only discourage the annotator.

5.2.2. Step 2: Exploiting semiotic markers linking to semantics

Defining the knowledge model is only one step of the semantization process. The principal task of identifying the corresponding knowledge in the resources is left. For this, we propose to identify the semiotic markers that identify the elements of the models.

For example, if the model states that the smallest block of information is the paragraph, then the layout must clearly distinguish paragraphs (for example using the “non printing characters” mode in MS-Word). If several paragraphs constitute a resource, as often, some kind of visual delimiter must be defined (e.g. horizontal lines). In addition, if the model defines the concept of “important notion of the domain”, it is likely that the corresponding words will appear in bold in the content, and so on.

This kind of “visual” information must be gathered and standardized taking into account the teacher’s preferences in term of layout. Technically this standardization is performed using styling features of classical edition tools like MS-Word or OpenOffice Writer.

Styles allow us to assign a role to the different components of the document. Figure 40 shows, on the right, the list of styles in MS-Word. Styles in OpenOffice Writer are presented on figure 43. This feature is mostly similar in both tools. It realizes the link between the semiotic level and the semantics of the defined model.

We worked on slide show presentations and unfortunately, the proposed method cannot be applied directly on them because the style features of the tools supporting such formats (Power Point, Impress) do not offer sufficient functionalities (or an XML format for Power Point). A preprocessing operation must take place to bring content into more powerful editing tools, like MS-Word and OpenOfficeWriter.

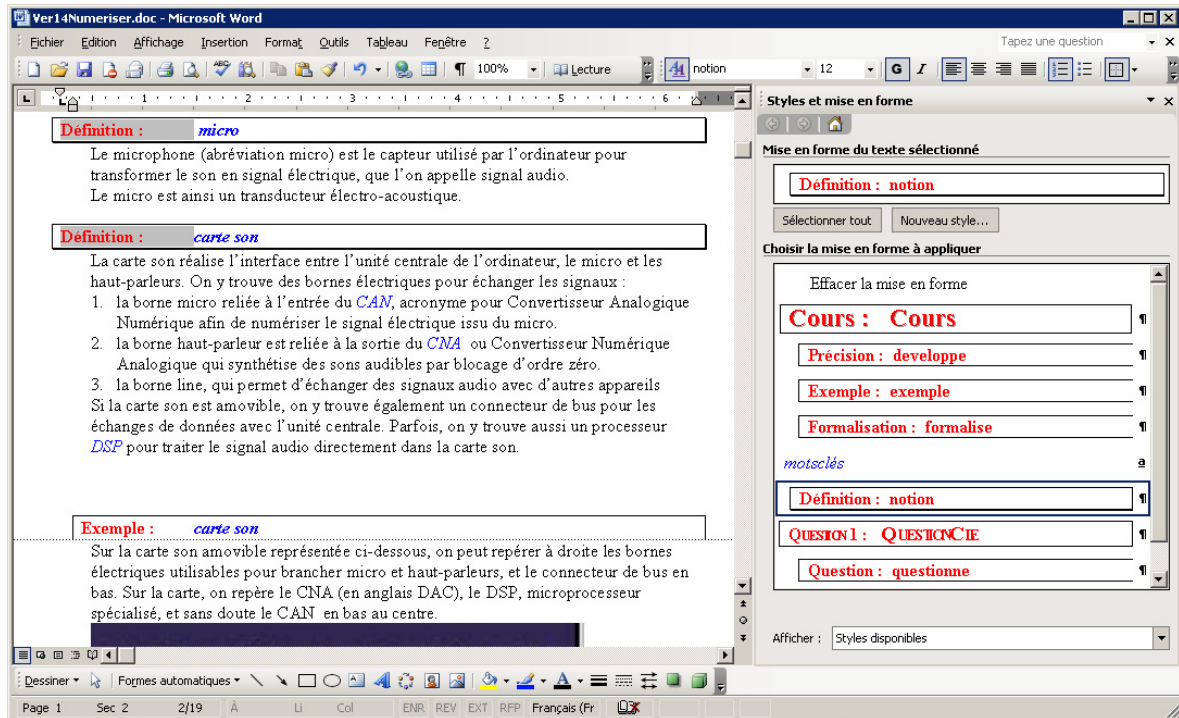


Figure 40 - Layout of the final document in MS-Word

Internally styles are configured in the document file itself using a specific XML representation. The configuration of the styles for a specific model can be performed by an automated process. Styles can be generated from the ontology in RDFS and inserted into the original XML using XSL transformation. Figure 41 shows the definition of styles in OpenOffice 2.0 format (OASIS standard) and in MS-Word on figure 42. In both cases the list of styles reflects the ontological concepts and hierarchy.

1. ...
4. <style:style style:name="InstructionalObject" style:family="paragraph"
5. style:parent-style-name="QBLSObject" />
6. <style:style style:name="Fundamental" style:family="paragraph"
7. style:parent-style-name="InstructionalObject" />
8. <style:style style:name="Auxiliary" style:family="paragraph"
9. style:parent-style-name="InstructionalObject" />
10. <style:style style:name="Definition" style:family="paragraph"
11. style:parent-style-name="Fundamental" />
12. <style:style style:name="Fact" style:family="paragraph"
13. style:parent-style-name="Fundamental" />
14. ...

Figure 41 – Style definition in OpenOffice 2.0 (Oasis format)

```

1. <w:style w:type="paragraph" w:styleId="PedagogicalResource">
2.   <w:name w:val="Pedagogical Resource" />
3.   <w:basedOn w:val="Normal" />
4.   ...
5. </w:style>
6. <w:style w:type="paragraph" w:styleId="PrioritaryResource">
7.   <w:name w:val="Prioritary_Resource" />
8.   <w:basedOn w:val="PedagogicalResource" />
9.   ...
10. </w:style>
11. <w:style w:type="paragraph" w:styleId="Definition">
12.   <w:name w:val="Definition" />
13.   <w:basedOn w:val="PrioritaryResource" />
14.   ...
15. </w:style>
16. <w:style w:type="paragraph" w:styleId="Example">
17.   <w:name w:val="Example" />
18.   <w:basedOn w:val="PedagogicalResource" />
19.   ...
20. </w:style>

```

Figure 42 – Style definition in MS-Word

5.2.3. Step 3: Automatic annotation

5.2.3.1. Pre-processing content

Linguistic operations can be applied to recognize concepts through their different forms (like plurals, synonyms etc.). If the domain learned deals with technical concepts, it will present fewer problems with homonyms and ambiguous words than when addressing more generic corpora. If an existing ontology for the domain is available, automatic annotation of the concepts can be performed in a preliminary or parallel step to manual annotation.

Using an XSL transformation, the content of the document can be analyzed and words can be assigned a specific style before the manual annotation process has started. For example, in the experiment on Java programming (QBL5-2) we found different ontologies on Java and applied this process. Mismatches appeared because of keywords like “if” or “object” belonging to both the usual language and the specific concepts of the domain. However, the error rate was acceptable, from the annotator’s point of view. The silence (concepts not matched) was important because some of the needed concepts were not defined in the reference ontology at that time (see discussion 5.4.2). The next step then involves the necessary evolution of the ontology.

Given the fact that automation builds on the existing ontologies, it could even be a parallel process, applied on desired parts of the course as the manual enrichment of the domain ontology goes on.

5.2.3.2. Contextualization problem

Automation is definitely a time saver to identify concepts. Yet each annotation must be checked manually. Looking at the corrections made by the teachers to the automatic annotations, sometimes proposed annotations were rejected whereas the highlighted word was truly identifying a concept of the course: the teacher judged that this concept was not important in the context of the resource and thus the resource should not be linked to this concept.

This shows the importance of the contextualization of the annotation that is very hard to reproduce using automated processes. As the resources are annotated inside the original document, the position of the resource in the document gives important contextual information. This is taken into account by the teacher, whereas it is not visible when looking at the resources alone. This position implies choices in the concepts that are highlighted by the teacher. For example, in a chapter close to the end of the document he/she will not highlight the concept of “object” (in Java) whereas he/she did so in the beginning. In the annotator’s point of view, this concept is supposed to be known by the time the student reaches this resource, and thus it does not need to be mentioned. However no rule can be drawn. For a different resource, the subject was a technical detail about objects in Java placed quite far from the beginning. In this case, the annotation with the concept “object” was very relevant even though the resource was located in an area where the concept is supposed to be known. A major principle of the semantization process, stated at the beginning, transpires here: *coherence of the annotation must be ensured with regard to the pedagogical strategy.*

Annotation related to the pedagogical knowledge can also be generated automatically. Such annotation is based on the occurrence of terms giving information on the pedagogical nature of the resource. For example, a resource containing the word “definition” in its title possesses a fairly high probability of being assigned to the “Definition” concept later on. This idea is relayed by the promoters of automatic annotation but it will always remain limited as mistakes are always possible and the effect of such a mistake on user’s disorientation cannot be tolerated in an e-learning content.

5.2.3.3. Conclusion

Finally, an automatic annotation does provide a significant help, but it must be checked manually afterwards, especially to contextualize the spotted associations. This contextualization cannot be performed by an automated process. The process of *semantization can only rely on a semi-automated mechanism.*

5.2.4. Step 4: Manual annotation

5.2.4.1. Using styles

After defining the standard styles, and semi-automatically annotating the content, an authoring phase must be performed to ensure that all the visual clues are present and they identify each component’s role according to the model. This represents the largest effort in teacher’s work, because it must be performed manually, and the whole course must be reviewed. Automated support can also help here to alert the teacher on potential problems (resources not being recognized, potential synonyms, etc.).

To add the missing annotations, or correct them, the teacher applies the styles on the document using the editor (MS-Word/OpenOffice Writer):

- First he/she selects the paragraph or highlights a word
- Then he/she clicks in the style list to assign the selection with the corresponding concept.

The resulting annotation is then carried by the layout. Figure 40 shows the final layout of the document (in MS-Word) used in the QBLS-1 experiment on signal analysis:

- The list of styles is visible on the right (Course, Precision, Example, etc.). The teacher has customized the appearance of each style according to his personal taste: each resource title is in a box, with its type indicated as a prefix to the actual title (the word “Definition”, for

example in the first two titles, is part of the style “Definition” itself and not written by hand by the teacher).

➤ The screenshot also shows three resources. The first two are definitions of the domain concepts (called *Notion* in the original model) labeled “micro” (microphone) and “carte son” (sound card). The third one is an example of the same concept of “carte son”. The word “micro” in bold italics and blue, identifies the concept this paragraph refers to. The same applies for the other paragraphs. MS-Words in blue italic, like “CAN”, “CNA” and “DSP” refer to other existing domain concepts in the course. The teacher decided that those concepts were related somehow to this paragraph as they appear in the content. The paragraph is itself primarily related to the concept of “carte son”, thus we can establish a link between “carte son” and “CAN”. Usually systems express this relation by “pre-requisite” or “see after” links, but the teacher participating in our experiment was not able to specify what exactly the relation was, because the use cases of this resource in the QBLS strategy were multiple. We opted for an “information link” which semantic is very much contextual. This is discussed further in the section on the difficulties met (5.3.5).

In the second experiment, we used Open Office to annotate the content of a Java course similarly. Figure 43 presents a screenshot of the OpenOffice Writer window where four resources are displayed, separated by horizontal lines. In this experiment, the pre-processing phase was purely automatic and the connection with the original document can be found by comparing this view with figure 37.

The third resource from the top is currently being annotated by selecting the style “Definition”, and applying it to the title “Methods and parameters”. The words “Methods” and “Parameter” are in italics, they have been assigned the style of “primary subject” (for this resource). The word “object” has been manually annotated by selecting another special text style. It indicates a reference to a different concept that is not the primary subject of this resource. As a result, the “annotations” in their graphical form state: *this resource is a definition for both the domain concepts “method” and “parameter” and it is related to the domain concept “object”*.

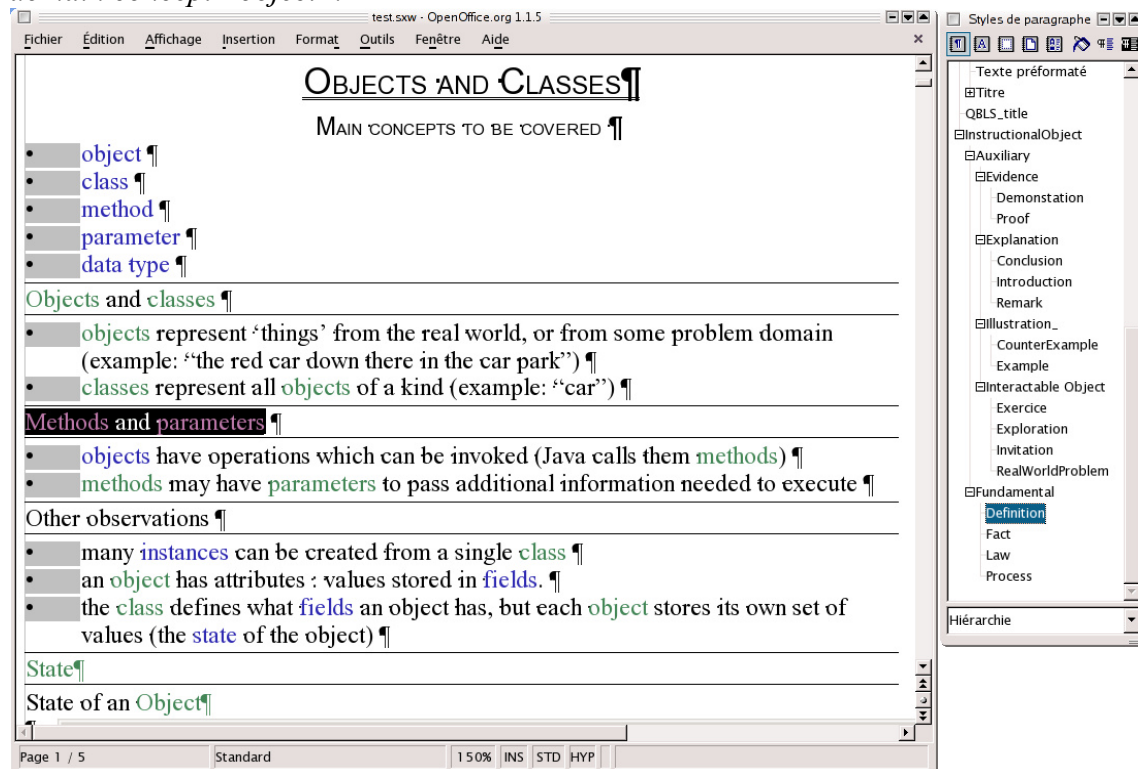


Figure 43 - Annotation of learning resources with OpenOffice Writer and its style hierarchy

The content of the questions, available in a wiki, was not included in this process mainly because questions were authored in parallel. Links to concepts in the questions had to be manually added. The syntax of the macro used the wiki page is presented below. Such syntax simplifies the task of adding links, an activity that can be described as annotating the assignment content with domain concepts.

[[Access(java:Accessor_Method, "Accessor Method")]]

In this syntax, the first argument identifies the URI of the concept and the second indicates the string that appears in the wiki page and materializes the link.

5.2.4.2. Defining and updating the domain ontology

The annotation of the content identifies concepts of the domain, the pedagogy and the document structure. Pedagogical and structural concepts are expressed in the styles. However, there is no explicit definition of a domain ontology in the tool interface itself. Concepts are identified through the terms contained in the textual content and annotated by specific styles ("keyword" or "primary concept" and "external concept"). The link with the ontological concepts is solely based on the identification of a term with the concepts labels. Syntax operations are performed to match upper-case with lower case words and a simple heuristic matches singular and plurals. For any homonyms or irregular plurals, the corresponding labels have to be defined in the ontology.

The identification of terms in a reference document is a very classical method for identifying concepts in an ontology creation process. By choosing to rely only on terms, the teacher has the freedom, while marking words, to define his/her own concepts "on the fly" and thus "adapt" the domain ontology as the annotation goes on. If a term cannot be matched to an existing concept, an alert is raised when processing the annotated file. The teacher is then asked to add the new concept "manually" in the ontology, or to attach the term to an existing concept. This process is specific to the importing mechanism of QBLS. It is supported by an interface disconnected from the annotation tool.

The domain ontology can express more than a simple list of concepts with various labels. An ontology may define hierarchies of concepts (for example with "subClassOf" or "broader/narrower" relations). This connects concepts together and offers "conceptual navigation" paths (see 4.4.2). Relations can be defined to ensure that all the resources of a chapter are linked in such a graph (i.e. a complete coverage of the material can be obtained through a conceptual navigation).

The QBLS system creates the graph and alerts the teacher when disconnected resources are found. The teacher can edit the domain ontology to add the required links connecting the resource and its associated concepts to the graph.

Instead of relying on terms in the content of the resources and modifying the ontology using a dedicated tool, the domain ontology could be automatically included in the right pane of styles like we did for the pedagogical one. Usability of such a device would be quite poor if manipulating a lot of concepts (up two hundred in our case). In addition, it is a much faster process to annotate words with a single style than having to select different ones for each term highlighted.

5.2.4.3. Conclusion

Once the layout is enforced throughout the document, the teacher's task is nearly achieved. By submitting the document on the server, the automated extraction phase proposed by QBLS can take care of the content and the annotations. The stylesheets performing this extraction

mechanism depends on the tool (Microsoft/OpenOffice) and on the ontologies (defining the style names). They can be automatically customized to take into account a new ontology, and its associated style definition.

It is important to notice that from the teacher's point of view this work is mostly performed with the sole use of the usual MS-Word/Open Office program he/she is familiar with. The QBLS interface is only used to modify the domain ontology according to the results of the extraction phase. The edition interface is presented in the next chapter.

Because the method adapts the knowledge models to the document, the domain ontology must be edited. However, the pedagogical ontology should not follow the same process for two reasons: (1) a large variability of this model is not suitable as coherence must be ensured all along the document, and (2) the number of concepts is normally quite small. If the teacher starts using too many different concepts, the value of annotations in term of guidance for the learner might decrease. Still if the ontology really needs to be changed, a dynamic edition of the ontology through the style hierarchy is technically possible but we have not experienced it.

5.2.5. Step 5: Production of formalized knowledge

We argue that the automatic extraction is not a difficult problem because the formats used for courses today (.doc, .sxw) possess a standard XML expression. The DTD used are public and the organization of the mark-up is quite easy to understand by a human agent. This markup can be treated by dedicated XSL transformations to extract the annotations and express them in the desired formalism (in our case RDF). A large range of input formats is acceptable. The method may apply to XHTML files, MS-Word documents or Open Office documents for example. The technical restriction for the application of the developed method is the use of accessible XML formats, with a DTD providing enough functionality to specify styles for paragraphs and words.

Once annotated, the document file is processed through different XSL transformations that produce on the one hand, a set of XHTML resources (the pedagogical resources of the model), and on the other hand RDF statements that contain the knowledge extracted from the annotations.

To give the reader a better understanding of the extraction process from a technical point of view we present below the steps performed by the "importing" mechanism, after the annotation phase took place:

1. First, the resources contained in MS-Word or OpenOffice Writer files are uploaded onto the server. For the later, the file is unzipped to access to the inner folder structure and XML files.
2. Then, a first stylesheet processes the XML file and extracts the annotations. It generates a unique RDF file containing all the knowledge extracted from the layout information. This transformation takes the domain ontology file as a parameter to obtain the URI of the concepts annotated in the resource. The generated RDF file is stored physically on the server
3. The following phase consists in preparing the XML content for the separation process into different resources. Basically, the content of each resource is copied into specific XML tags.
4. Based on these tags the content is sliced by a splitter service. This service generates as many files as there are resources identified and fills them with the corresponding content.

5. For each of the XML file generated by the splitter another XSL transformation translates the original format (MS-Word or Open Office) into XHTML.
6. Finally, the files are installed on the web server and the annotations are loaded into the knowledge base (see chapter 6).

5.2.5.1. Extraction mechanism

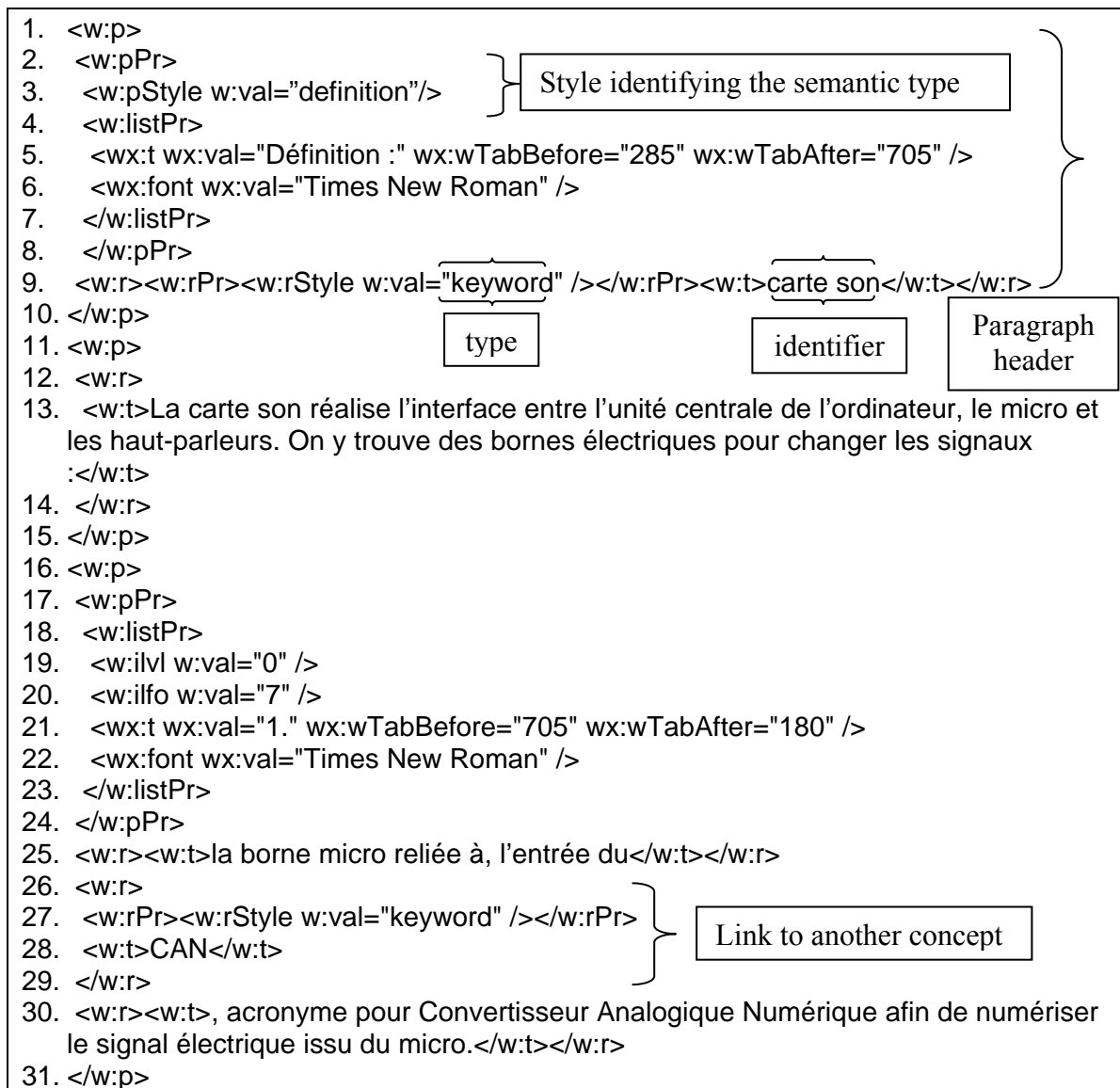


Figure 44 – Annotated content in MS-Word-XML format

The layout of the resource, visible in the middle of figure 40, matches the XML format showed on figure 44. The type of the resource is identified by the style of the first paragraph (w:pStyle tag, line 3). The domain concepts are identified by the style “keyword”, line 9.

An example of the RDF annotations generated is shown on figure 45. In this counter part, the domain concept in the title of the paragraph translates to a *subject* relationship between the resource and the concept (line 6). Links to external concepts, for example here “CAN” line 11, are specified with the relation “edu:keyword”. Whenever possible we used standards like Dublin Core (DC, 06) to encode knowledge such as *title* (line 4) or *subject* relationships (line 6). It allows us to compare this approach with others and to envision that such information might be later on shared between different systems. The model presented on figure 38 is

encoded in an RDFS ontology and appears here through the classes like “edu:Definition” line 3, “edu:Notion” line 7 and the relations “dc:subject” and “edu:keyword” (line 6 and 11).

In (Brusilovsky, 03) a similar technique was proposed to annotate courses using RTF, another markup language. This was restraint to one style and the visual correspondence between layout and semantics was hard coded, whereas it is totally dissociated in our proposal.

```

1. <?xml version="1.0" encoding="utf8"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
3.   <edu:Definition rdf:ID="defid2248279">
4.     <dc:titre>Définition :carte son</dc:titre>
5.     <edu:contenu rdf:resource="file:/data5.xhtml">
6.     <dc:subject>
7.       <edu:Notion rdf:ID="carteSon">
8.         <edu:label>carte son</edu:label>
9.       </edu:Notion>
10.    </dc:subject>
11.    <edu:keyword rdf:resource="#CAN"/>
12.  </edu:Definition>
13. </rdf:RDF>

```

Figure 45 - RDF annotation for QBLS

5.2.5.2. Domain representation using SKOS

In the QBLS-1 experiment, the domain concepts are solely represented by instances of the concept of “Notion”. In the second experiment the domain is described using the SKOS, Simple Knowledge Organization System (SKOS, 05), standard. SKOS is defined precisely in the scope of describing domain knowledge, such as thesauri, classification schemes, subject heading lists, taxonomies and other types of controlled vocabulary. Terminologies and glossaries are also envisioned within the framework of the Semantic Web.

SKOS is interesting to use in this context because it offers a basic meta-model that is sufficiently rich to express hierarchies, but does not impose strict semantics. With little training the teacher can be autonomous in the task of manipulating SKOS models for his/her course. This is not the case with more constrained languages like RDFS or OWL, which require a good expertise to generate models, that respects the semantics of the language.

A language for expressing taxonomies was already proposed in the LOM standard using XML formalisms and a hierarchy of “taxon” tags. SKOS provides this with an even wider standardization perspective.

Figure 46 and figure 47 feature an example of XML content of an Open Office Writer file and the corresponding RDF annotations generated by QBLS. The annotation uses SKOS. This example, from QBLS-2 represents a part of the course shown on figure 43. The paragraph on “methods and parameters” (line 1) is annotated as relevant for both the concept of “Method” (line 2) and the concept of “parameter” (line 3). This is expressed by the “Primary_ref” style in XML-OpenOffice (line 2 and 3). The concept of “Object” is an external reference for this resource, annotated with “External_ref” style (line 8). In RDF, these semantics are carried by the links skos:primarySubject (line 5,6) and skos:subject (line 7) which are part of the SKOS standard. Both relations are specialization of the dc:subject relation used in QBLS-1 (see figure 45).

```

1. <text:p text:style-name="Definition">
2.   <text:span text:style-name="Primary_ref">Methods</text:span>and
3.   <text:span text:style-name="Primary_ref">parameters</text:span>
4. </text:p>
5. <text:unordered-list text:style-name="L1">
6.   <text:list-item>
7.     <text:p text:style-name="P1">
8.       <text:span text:style-name="External_ref">objects</text:span>have operations
          which can be invoked(Java calls them <text:span text:style-
            name="Primary_ref">methods   </text:span>)
9.     </text:p>
10.   </text:list-item>
11. </text:unordered-list>
12. <text:unordered-list text:style-name="L1">
13.   <text:list-item>
14.     <text:p text:style-name="P1">
15.       <text:span text:style-name="Primary_ref">methods</text:span>may have <text:span
          text:style-name="Primary_ref">parameters</text:span>to pass additional information
          needed to execute
16.     </text:p>
17.   </text:list-item>
18. </text:unordered-list>

```

Figure 46 - XML format in OpenOffice Writer

```

1. <rdfs:Resource rdf:about="http://www.inria.fr/acacia/prog101#chap1fic3.xhtml">
2.   <rdf:type rdf:resource="http://www.inria.fr/acacia/edu-onto#Definition"/>
3.   <edu:number>3</edu:number>
4.   <dc:title>Methods and parameters</dc:title>
5.   <skos:primarySubject rdf:resource="http://www.inria.fr/acacia/java-skos#Method"/>
6.   <skos:primarySubject rdf:resource="http://www.inria.fr/acacia/java-
   skos#parameter"/>
7.   <skos:subject rdf:resource="http://www.inria.fr/acacia/java-skos#Object"/>
8.   <edu:belongsTo
   rdf:resource="http://www.inria.fr/acacia/prog101#ObjectsAndClasses"/>
9. </rdfs:Resource>

```

Figure 47 – RDF generated for Java courses

This example shows that annotation can be redundant as the concepts “Method” and “Parameter” have been annotated several times in the document. This is a side effect of the automatic annotation. Each time the word appears it is annotated, even if this information is redundant. The extraction process allows us to filter such redundant information.

5.2.5.3. Conclusion

The last step of the process is linear and fully automated. Technically, it is based on several XSL transformations with stylesheets that are mostly generic and can be automatically customized. An interesting solution was deployed to split the resources because actual XSL transformation can only generate one output. It is a problem when several resources need to be extracted from a single file (in this case the pedagogical resources). The solution we propose is to generate an intermediate proprietary XML format that can be split with a very simple program (in Java here).

All the models are expressed in RDF/XML. They can be accessed directly by XSL stylesheets. This largely contributes to concentrate the process in a single technology, a must for maintenance and evolution of the developed code.

From a performance point of view the import of the Java course, containing 359 resources only takes a few seconds. Scalability is quite good as we found that such a file was already big to manipulate in the editor. We then recommend working on smaller entities like one file per chapter.

5.3. Evaluation

As explained above, we conducted two campaigns of experimentation using this annotation method. We study in this section the outcome of these experiences with regard to the evaluation of the semantizing process. In particular, we try to evaluate the amount of effective reusability offered by the method, we report on the qualitative and quantitative inputs we collected and present the difficulties met.

5.3.1. Reusing existing material

The first aspect to evaluate is the achievement of the main goal of the method: the possibility to reuse effectively an existing document. In this domain, the objective of the method is two fold: saving time by reusing material, but at the same time preparing the material for use in a dedicated system. We examine how those two objectives are effectively combined in practice.

5.3.1.1. Preprocessing operations

The amount of necessary preprocessing is quite important when looking at the difference between figure 36 and figure 40 and between figure 37 and figure 41.

Thus preprocessing was successfully automated in OpenOffice. Thanks to XML format of the slides, this operation is largely facilitated and the developed stylesheets could be largely reused. We are confident in the fact that slide show tools will evolve towards offering more styling functionalities in the future. A certainty is that open XML formalisms will be used more and more in popular content edition tool, making preprocessing instructions more and more generic and reusable.

A different problem concerns formats like PDFs that are not supported by this approach. They are not “editable” in an editor separating content from presentation. Another problem is that “interactive sequences” of slide show presentations lose most of their meaning when imported in a static text editor.

5.3.1.2. Layout manipulation

At first, the layout is somewhat imposed by the existing document. This layout might be difficult to interpret and the document model may reveal cumbersome to express. Even if the method has been applied several times (see chapter 9 for more examples) there are cases where the layout is too complicated, not homogeneous or just semantically poor to be of any value for an extraction of the document model. In this case, the reuse of the document must be questioned.

When reusing existing material, the reuse of the existing styles already in place must be considered as mentioned above. However technically, the styles have to be defined specifically for the annotation process. A mapping between original styles and the ones used for annotation must be established. In the best case, just renaming the original styles is enough. In other cases, manual action needs to be performed to reassign the new styles over the original ones. This other preprocessing task must be mentioned and included in the evaluation of the cost of reuse.

These difficulties show the necessity to start from a material of *reasonable quality*, and to take special attention to the *existing layout of the material*.

5.3.1.3. Conclusion: Reuse vs. Authoring

In the first experiment, the teacher did modify the content in order to refer explicitly to interesting concepts not originally mentioned. The connexity of the graph formed by the links between concepts and resources (see chapter 4) was ensured by this manipulation. In the second experiment, we demonstrated that this was not necessary. The teacher did not modify the content but created links between the concepts themselves using the domain ontology. Thus, the resulting graph would still be connected without modifying the resource. This is a more explicit way of encoding the knowledge that must “guide” the learner in his/her task and reinforce the argument in favor of reusing material

We insist on the fact that this method targets reuse more than creation. If the course does not already exist, teachers might consider different approaches. For example defining the model first, and then using it as a guideline for authoring, as proposed in most adaptive systems and in the MISA method (Paquette et al., 97). The “authoring” part in the method consists in expressing the additional knowledge that “wraps” around the content to facilitate its reuse. It constitutes the originality of the approach. If the content does not exist already, there is less interest in this method.

The reuse philosophy is totally generic. Each application of the method should target a specific learning system, but the scenario itself could be applied to a large range of systems. This is emphasized by the fact that we rely on standards (RDF, XHTML) for expressing both content and knowledge.

5.3.2. Qualitative view of the produced annotation

We conclude from these experiments that, apart from the lack of styling features in editors, the slide show format is quite convenient for the reuse process. In particular, the concise writing style often employed is well adapted for this type of annotation. The following paragraphs present the positive points observed during the experiments. They are qualitative judgments that we justify by empirical evaluation.

5.3.2.1. Manipulating visual information

The visual representation of the annotations seems to facilitate the annotation task for the teacher. This certainly depends on the cognitive profile of each practitioner, but this approach certainly helps “visual thinkers”. We have also observed that the possibility, for the teacher, to define his/her own layout for the styles (e.g. changing the color or the font) helps him/her in recalling the associated semantics. As learners need personalization of their interface, this is personalization for the teacher.

5.3.2.2. Exploiting existing markers

We also clearly observed that existing markers were valuable information to bootstrap the process. In fact, part of the annotation work is already done, in an informal way, in the resources layout. With sufficient quality resources, styles have semantics that are enforced throughout the content. In the courses we reused, documents already possessed a certain amount of the necessary information carried by visual clues (bold, italics, underlining, etc.). Straight forward annotation of a noticeable part of the knowledge can be obtained.

5.3.2.3. *Taking advantage of existing tools*

One of the originalities of the proposed method is to rely only on the available features in text editors to perform this task. According to (Uren et al., 05), integrating the annotation process into classical editors is a must. We verified that editors like OpenOffice Writer or Microsoft MS-Word provide some interesting features. They revealed easy to handle because teachers are used to manipulate them. The learning phase necessary to turn them into annotation tools was actually very short. They also offer the possibility to fully modify the content, something that no small tool can perform.

We have observed the importance for the teacher to stay in control of the content so that he/she can modify it as needed. There is no limitation on edition as the full capabilities of the tool can be exploited. This integration in standard tools has started to be considered in other works. For example it is the orientation chosen by the research project Alocom (Verbert et al., 05) as well as the commercial tool Course Genie (CG, 06). Such contributions go further than what we did by integrating plug-ins inside the interface of editing tools. This is of course more powerful but it requires an important engineering effort and administrative rights to install the plug-ins that are not always granted to teachers. The learning time for using the tools also increases. We prefer a more light weight approach, that fully takes advantage of the existing functionalities.

5.3.2.4. *Relying on an existing meta-model*

The use of SKOS comforts us in the fact that this kind of annotation is going to be extremely important in the future. We started to develop the method before the publication of SKOS, but the perfect match between SKOS philosophy and our project is quite interesting. For example, the fact that we already defined the concept of “Notion” and external and primary relations is a major clue in favor of the proposed semantization approach.

5.3.3. Quantitative data on two experiments

Table 5 presents a few figures concerning the experiments and the result of the semantization process.

Table 5 – Figures about QBLS-1 and QBLS-2

	QBLS-1	QBLS-2
Number of resources	92	359
Number of pedagogical types used (directly)	8	12/27
Number of domain concepts	41	171
Editing Tool	Microsoft Word	OpenOffice Writer
Annotation time	Undefined	20H
Expected time of use for the resources	2H	3 months
Number of resources discarded	None	54
Modification of content	Yes	No

5.3.3.1. *Resources*

The first course was quite large already and leads to the creation of 92 resources. Resources usually contained one or two paragraphs and could include pictures, formulas, and drawings. Considering that the course would only be used for two hours, this may seem big compared to the second experiment (359 resources covering a full semester). This has two explanations:

- This figure includes 18 resources relative to the questions involved in the QBLS process. In the second experiment, questions are not included in the count.
- The teacher over-estimated the motivation of the students and did not anticipate that the number of resources was in fact too high for a two hours course (as we shall see later, some resource were not used at all by students).

The table also mentioned that some resources have been discarded in the Java course. Such resources only contained code examples that were judged irrelevant in this context of use. They were necessary for the linear reading but did not make sense in a conceptual view. That means they could not be attached to any domain concept. Transition slides with only paragraph titles appear in this count but they “disappear” after the annotation process.

5.3.3.2. *Time spent*

The first experiment went through several iteration cycles because the method was not yet clearly defined. No figure is available for the time spent due to those iterations. The second experiment was conducted in a more linear way. It was dealing with quite a large set of resources and the annotation task was then foreseen to be relatively long. Nonetheless, the time spent is higher than expected. This can be explained by the fact that the method was still being refined at that time. The definition of the domain structure was not facilitated and direct upload of the course was not working properly yet. Therefore, it is likely that in further applications the annotation time will be shorter. Around 10 hours for 400 slides and a hundred concepts seems a probable result.

5.3.3.3. *Conceptual models*

In QBLS-1, 8 types were used to identify the pedagogical role of the resources (including the three types related to questions). In QBLS-2, 12 types, out of the 27 proposed by the ontology shown on the left figure 19, have been used. In the first experiment, only the most specific concepts were available for annotation (leaves in the hierarchy of concepts). In the second case, the whole hierarchy was available and the teacher used two different levels. This was useful for example when the nature of the resource could not be determined precisely to use an upper concept to annotate. However, the top distinction fundamental/auxiliary (see figure 39) was not used directly.

For the conceptual domain, the number of concepts in the second case is naturally much higher. A specific type of domain concept had to be defined for the case studies that illustrate the course. They are in essence specific notions of the course, but only contextual ones, that are related to the organization and strategy of the course. It is clear that for the domain a completely generic model (like a reused ontology from another project) cannot fit such contextual specificities. This is further discussed in 5.4.2.

Existing ontologies were retrieved from the web. For the domain, we only extracted a list of candidate concepts from an existing experiment (Henze, 05). The pedagogical ontology, from (Ullrich, 04), was reused without modifications. Only 11 types out of 25 available have been used by the teacher to annotate the resources. The most intensively used indicate the

pedagogical orientation of the content: Definition, Law and Examples. This is a quite natural result for an introductory course and thus comforts the assessment of this process.

5.3.4. A graph perspective for domain model

The domain models encoded in RDF graphs can be represented visually as graph structures. In this section we investigate the potential of such visualization.

5.3.4.1. Enriched domain model

The graph of the domain gives a visual representation of the teacher's vision. But only few relations between concepts are defined directly by the teacher (none in QBLS-1 and 60 in QBLS-2). Ontology building techniques sometimes rely on the extraction of relations from text to structure concepts. We propose to use the existing annotations to extract similar associations.

Links from documents to concepts are not typed for the teacher as explained previously (5.2.5.2). In RDFS we encoded them as "rdfs:seeAlso". The semantic navigation would use those links to go from one concept to another one through a resource.

We can then interpret each triple (concept, resource, concept) as a relation. Translating this interpretation into a relation in RDF leads to the definition of a "conceptual structure" without any resources involved (directly).

5.3.4.2. Diagnosis tool

The generated relations only express the fact that a resource exists and that it links both concepts. It offers an interesting view of the organization of concepts. For example, a visual representation allows us to spot isolated concepts, loops, etc.

Such a representation may serve as a diagnosis tool for checking the course coherence. Figure 48, presents the result of this conceptualization for the QBLS-1 experiment. The colored nodes are high-level concepts, like questions and themes. The "course" node is the violet one at the top. This graph shows the structuring impact of the "high level concepts": They ensure the interconnection between "grapes" of low-level domain concepts, which are disjoint. These interconnection links are indicated in black, whereas, the relations between low-level concepts are in red. The semantic of those links is contextual. It depends on the way the material has been authored.

Figure 48 - Conceptual structure in QBLS-1

The opposite is observed when drawing, with the same tool, the graph of concepts in QBL5-2 (see figure 49). In this figure, the relations in red indicate the links that have been manually defined between the concepts. We observe that the majority of the concepts are connected together in the graph. Links manually defined and those inferred from the annotations complete each other without overlapping. This shows that the structuring of the domain was necessary to bring the additional links that connect concepts together.

This structure was brought to the domain vocabulary during the annotation process to ensure the connectivity of the navigation graph. Connectivity is an interesting property to ensure that all the resources can be accessed through a conceptual navigation. We ask whether a good indicator of annotation quality could be the connectivity of the graph according to this link generation. If few manual structuring of the domain is necessary, then the conceptualization and the relations between concepts is a good match for the resource content.



For the moment we envision this type of representation as a diagnosis tool rather than an authoring tool. Even if the automatic generation of such graphs can be easily implemented they would be difficult to manipulate for teachers.

In addition, it would be useless to ensure artificially a connectivity that would have no justification from the point of view of resource content.

A disconnected course is necessarily bad. What the graph indicates is that logical paths (in the sense of paths that can be performed through a conceptual navigation) exist throughout the domain representation. Such paths show an undisputable coherence. This is far from ensuring a good quality for the material in general. Many other parameters (quality of writing, presentation, clarity, interactivity, etc.) have to be taken into account as well.

5.3.4.3. *Modeling tool*

Apart from the visual representation which is quite informative on the design assumptions of both the annotator and the documents. Such knowledge can also be used to compare different conceptualizations.

Taking a closer look at the potential semantics of the links, most of them clearly do not express subsumption. For example the following sentence is taken from the Java course: “Each class has source code (Java code) associated with it that defines its details (fields and methods)”. In this sentence the words “fields” and “methods” are annotated as external subjects. The resource itself is about the concept of “source code”. However, there is no subsumption relation between those concepts.

If the links cannot be compared to hierarchical domain model, on the contrary we feel that this kind of information (graph of the conceptual structure) is a good way to compare the modeling approaches. Because the concepts are not forced into a hierarchical structure, they also better define the domain to learn. Hierarchy is a design pattern for classification purposes, not for domain knowledge modeling in a pedagogical perspective. Unless the chosen strategy is purely top-down, domain knowledge has no reason of being particularly hierarchically organized.

In conclusion, we propose to *complement the semantization method by this diagnosis and modeling tool*. Thanks to the level of abstraction offered by the navigation model, such extraction of the conceptual structure can be easily generalized to any application relying on this model.

5.3.5. Difficulties met

The experimentation of the process, even though successful, has highlighted several problems not fully answered by the proposed method. We present them below.

5.3.5.1. *Resource evolution*

One of the main issues, not yet addressed, is the loss of the link with the original material during the semantization process. When a resource is retrieved from the web and annotated as we recommend for inclusion in an on-line system, the further evolutions of the resource cannot be integrated easily in the system. For example, if the original author updates the resource and reorganizes the content, this resource will have to be annotated again from the start. This is by no mean a specific issue for this work, but a more general issue faced when reusing material in general. A potential solution is to share the annotated version rather than the original one, and start the evolution of the material from the annotated version, so that the knowledge contained in the styles will evolve at the same time. However in many situations

this is not possible. Another possibility is to create a push mechanism that will at least alert the teacher that the original document has been updated, so that he/she can follow the evolution quite closely.

In a near future, matching techniques based on XML should be able to identify the evolutions in a document and associate the old annotations to the new content, thus facilitating the integration of updates by just requiring an annotation of the new parts. This direction is illustrated by research work like (Su, 01). However for the moment no practical solution can be implemented.

Another problem, linked to the evolution of knowledge expressed about the course, concerns dynamic behaviors. If the teacher wants to update annotations while the system is being exploited by students, we found it easier to use a dedicated editing interface and to act directly on the knowledge stored in RDF, rather than opening the large annotated file and resubmitting it again on the server. The changes, performed using the RDF edition tool (see 6.2.2), are not reported onto the original document. A solution might be to reconstruct the annotated document from both the extracted resources and the annotations. If possible in theory, this would require a large engineering effort in practice due to the complexity of XML formats in text editing tools. If XML schemas are easy to understand, generating complete files automatically represents a real engineering challenge.

5.3.5.2. *Defining prerequisites*

According to (Brusilovsky and Vassileva, 03), the “pre-requisite” role is the most used one to index pages with concepts. Many research works in the field also acknowledge to use this kind of relation (like (Dolog et al., 04) or (Bouzeghoub et al., 05) for example). We tried to ask the teacher to specify the “pre-requisite” concepts for a paragraph but the experience showed that this kind of relation was very hard to define. It is mostly contextual, depending on the learner’s background, his/her previous path, etc. Just by looking at the course, the prerequisite idea seemed difficult to express.

For example, if we consider the following sentence from the course used in QBLS-1: “*The microphone transforms the sound into an electric signal*”. This fragment is obviously about the concept of “microphone”, but would you say that the knowledge of “sound” (another concept of the course) is a prerequisite to understand this sentence, or that this sentence invites you to learn more about the concept of sound? Depending on this choice the path through the course is totally different but there is no valid reason why one way should be preferred over the other, unless we want to restrict user’s path to a constraint linear progress with no navigational choices.

Pre-requisite links may also be defined at the document level to express predefined paths among the resources (de Bra et al., 03) (Dolog et al., 04) (Henze, 05). However, this approach is incredibly complex for the author as shown in the small example presented in 3.1.7.2. Thus, we are doubtful whether a complete course can be equipped with a prerequisite structure more refined than the classical hierarchy of chapters and subchapters.

Existing large works that use prerequisite links, either rely on an ontological structure (Crampes et al., 00) or on the document structure (Henze, 05) and interpret them as prerequisite links. However, they were not manually authored in that scope. “requires” information is in fact part of a specific reading and navigating strategy always defined at a higher level. We feel that the semantics of this link is largely misused. Interviews with teachers showed the difficulty they faced to express relations like “pre-requisite”, or

“required-by”, whereas they could perfectly describe a top down or a bottom up navigation strategy.

We do not deny the existence of pre-requisites in on-line course systems. When annotating, the teacher implicitly defines pre-requisites. For example we observed that not all the terms related to domain concepts were highlighted by the teacher in the content (see 5.2.2). Depending on the context in which the resource fits in the linear structure of the course, some concepts are supposed to be known already. By comparing the result of an automatic annotation with the manual annotation, pre-requisite links can be determined. In this case they are “true” pre-requisites, and not interpretations of a conceptual model.

From the experiences we made, it appears that prerequisites can be defined at a high granularity. For example at the chapter level it is far easier to determine prerequisites. In this case, prerequisites express the knowledge necessary to understand the chapter resources as a whole: the prerequisites concepts are those out of the domain covered by the chapter. They can be used to check if the chapter is suitable or not for a given learner’s state of knowledge, but they cannot imply an organization of the resources inside the chapter itself. The application of this “pre-requisite” strategy at the resource level seems too restrictive in terms of learning paths and too difficult in terms of annotation task to be useful in a conceptual navigation context.

5.3.5.3. *Limitations of layout based annotation*

As mentioned earlier, this method solely relies on existing functionalities of editing tools. On the one hand, it constitutes a strong argument in favor of the method in term of adoption as no software needs to be installed and the training phase is very short. On the other hand it obviously reduces the possibilities for expressing annotations and highlights the limitation of an annotation purely based on the layout.

A first limitation is linked to the use of textual expressions to identify domain concepts. If a term identifying the concept does not appear anywhere in the resource, annotation with this concept cannot be performed. For fully textual courses, this case is not frequently met. It may even indicate an unclear redaction of the content as it is advisable for content clarity that a term appears to identify the concept, even if the resource refers to it implicitly. For mostly graphical courses however, the problem is different. Images must be described textually in the legend or using the text replacement functionality. In any case it represents an overhead of work. For diagrams, term recognition can be applied using the text contained in the drawings. The problem is that with today’s editors, text in graphical text boxes cannot be annotated with styles. Annotation must rely on the analysis of layout features (mostly the color of the text) to match it onto an existing style. This is a generic problem of information extraction based on texts and can only be overcome by using a different paradigm for annotation.

Another limitation is that pedagogical annotation of a resource depends on its “paragraph” style. Thus a resource can be annotated only with one concept. If multiple annotations are required, corresponding concepts, combining the different aspects into one concept, will have to be introduced on a case by case basis:

- If only one ontology is used to annotate (as in our case) the problem is avoided: It does not really make sense to have multiple annotations from the same ontology. If a resource has multiple types in the scope of a single conceptualization, the corresponding concept certainly needs to be introduced.
- On the contrary, in the case of annotating from several points of view, specific tools will have to be introduced. For example annotation can be performed from a pedagogical point of

view, and at the same time with cognitive learning profiles in mind (e.g. visual/textual, experienter/thinker, etc. (Dagger, 02)). This is a limitation but if encountered in real practice, the objectives in term of knowledge annotation are certainly much higher than what we are looking for in the semantization process and it involves a different scenario.

Finally, a last limitation was encountered through unexpected behaviors of the tools. Editors like Microsoft Word or OpenOffice Writer are only concerned with the visual aspect of content formatting. The semantic of the formatting might be deduced from the visual aspect but originally the tools were not designed for that purpose. Thus some resources may look the same but the underlying markup is different without any way of telling it from the interface. For example depending on how the course was created some words are split by unnecessary markup. You may find expression like those:

```
<text:span text:style-name="Primary_ref">Meth</text:span>
```

```
<text:span text:style-name="Primary_ref">ods</text:span>
```

In the interface this is perfectly equivalent to:

```
<text:span text:style-name="Primary_ref">Methods</text:span>
```

In the first case it is difficult to identify the concept. Of course heuristics can be developed to “tidy” the mark-up and avoid such problems. Still there is no guarantee that all problems will be solved. The only way to correct this is to check the extraction mechanism manually afterwards. In cases like the one presented above, alarms can be raised as none of “Meth” of “ods” will be recognized, but some errors are harder to identify, especially because the visual rendering in the tool will not show any inconsistency.

Technical solutions have been deployed as workaround for those limitations. This is nonetheless a difficult problem because it highlights a major weakness of the principle of the relationship between content and presentation, on which all the process relies. In this case we have different semantics for the same look. This somehow contradicts the principle that “*if two elements look the same they should mean the same*” (5.1.3.1).

The frequency of these incoherencies was low in the resources we manipulated. Nevertheless, it highlights the fact that current editors are not semantic editors and the question is raised whether such tools will evolve towards a more “semantically aware” mark-up and towards more structured documents. We consider it as a crucial question for the future of semantics extraction from layouts (not just for e-learning).

5.3.5.4. *Human factors*

Our experience also showed that most problems were not technical or conceptual ones but occurred because of the unreliability of the human eye. We observed that the teacher who was annotating and reusing his own course (QBLS-1) had difficulty to spot visible errors (like misspellings in the titles or in concept names for example).

This is an important result to consider for real applications. We are confident that such difficulty could be overcome if the teacher fully takes control of the whole system and can benefit from the automated checks (in the first experiment, the teacher knew that a pedagogical engineer would review the material). Nonetheless, this is actually the major drawback of not using a more constrained tool, where dynamic feedback could be given using checks performed in the editing tool itself.

5.4. Impacts

Looking back at our experience of semantizing a course and its initial objectives, we draw the following conclusions:

- The method induces a new life cycle for pedagogical resources.
- It reduces the extent of ontology inter-operability by limiting models to single applications and documents.
- It affects teacher's role in the organization of learning.
- It defines a new approach of conceptual navigation, based on the explicit construction of a conceptual space.

5.4.1. Semantization cycle

The proposed method allows us to envision a slightly different approach to learning application deployment. Compared to classical models, like the MISA methodology (Paquette et al., 97) or the knowledge life-cycle proposed by (Millard et al., 06), the role of the pre-existing implicit knowledge stored in the content is emphasized. Figure 50 shows a comparison between (Millard et al., 06) and the cycle introduced by the semantization method we propose.

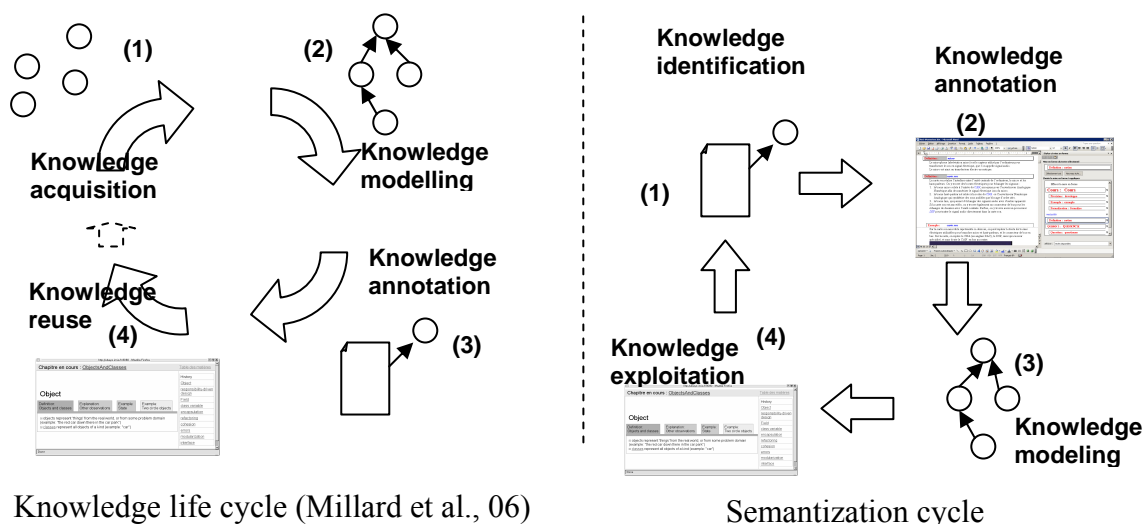


Figure 50 – Comparison between the knowledge life cycle (Millard et al., 06) and the semantization process

In the classical view (left), the first step is to acquire knowledge from a domain expert (1), and to develop a domain vocabulary. Then, this description is formalized in an ontology (2). Classes are created for the concepts and linked together. It is only in the third phase that actual resources are taken into account to be annotated with ontological metadata (3). The last step consists in evaluating and using the formalized knowledge, trying to exchange resources and integrate them in learning systems based on the defined knowledge (4).

The semantization cycle is somehow different, even if similar goals are pursued. The first step of the cycle is based on the final objective, which is reusing the resources. Resources are first selected as a coherent set and important domain concepts are spotted to form the vocabulary (1). The strategy and associated concepts are also defined at that time. In a second phase (2), the knowledge is systematically extracted from the resources using the annotation method described above. We can notice here that no practical method for annotation is proposed in the previous mentioned works. The third step (3) consists in organizing the concepts once they have been assigned to resources. That way the conceptual structure really matches the vision

proposed by the reused content. Formalized knowledge can be used in generic or dedicated tools (4). The cycle might be performed several times before the system is actually accessed by students. Exploitation in the final tool gives important feedback to the teacher on the possible mistakes or modeling problems that lead to inconsistencies in the proposed access to content (this remark applies to both cycles).

Concerning the feedback and the potential iterations of the cycle, we observed the importance of allowing the teacher to see the impact of his/her structuration effort on the final display. That means the cycle could be integrated at a finer level for each concept definition. As soon as a concept is identified as important for a resource, it is created and the access it allows to the content is validated in the final interface (i.e. the interface visualized by learners). Depending on this feedback, the teacher may go again through steps 1-2-3 and act on the structure of concepts to open conceptual navigation paths through the resources.

We can also analyze this cycle with the work of (Prie, 00) who describes such activity as a “conceptual annotation”.

- First, it is a process of contextualization. It can eventually be guided by known schemas, but can also be free. This joins the discussion on ontology reuse or creation mentioned in 5.3.3.3.
- It identifies resources (or documents) as belonging to specific genre. This identification remains partly implicit, but the discarding of some resources is a clue for this interpretation based on genres: Their genre was too different from the coherent set that the teacher wanted to build.
- It breaks free from the linearity of the original description contained in the complete resource.
- The conceptual annotation places the resources in a context that may change their original type, which is quite important to notice compared to approaches based on Learning Objects where resources have static description of universal value.

In the end, the cycle proposed is much more grounded in practice and closer to practitioners’ interests than classical models trying to deal with more generic knowledge models. It can be compared with the expression of the Learning Object life cycle described by (Catteau et al., 06) applied to knowledge expressed about a specific document.

5.4.2. Ontology interoperability

The method both reuses and defines conceptual models. Ontologies for pedagogy have been successfully reused but the knowledge about the domain needs to be customized to fit the reused document specificities. In that scope, we propose to build dedicated domain vocabularies using SKOS and discuss in this section the potential of reuse for domain models.

5.4.2.1. Problems with reusing ontologies

In many of the experiments reviewed in the literature (Colluci, 05) (Winter et al., 05), ontologies are small excerpts of imaginary domain ontologies. Authors acknowledge that the cost of creating a complete representation was out of their reach. This problem is quite generic and may be faced in many different semantic web applications.

The natural answer of the semantic web paradigm to this problem is to exchange and share existing ontologies, like pedagogical resources in the learning object paradigm. Existing conceptualizations are resources themselves that should benefit the whole community. However, like with pedagogical documents, reuse is difficult:

- One of the issues with domain models is their pedagogical nature. It is very likely that the reused ontology has not been defined for the same objective as the new one. In that case, some

necessary concepts will be missing. The global conceptualization offered will be hard to read, and will badly influence teacher's acceptance.

➤ The structure of the ontology relies on a specific vision of the domain. To be useful for annotating material, this vision must match the content. Otherwise, the formulation of the content will not be coherent. We already noticed that inconsistencies in the learning path are absolutely not acceptable for learners. It would disturb their early comprehension of the domain and ruin the learning process.

Reuse of domain ontology must then be performed with great care. A deep understanding of their role in the system is necessary to decide on their reuse. To evaluate the potential reusability of existing domain ontologies, we performed a comparison between several domain ontologies and identified the differences that mattered for reuse.

5.4.2.2. *Comparison of different models*

In a first experiment (QBLS-1) we made the deliberate choice to interact with a teacher whose courses are not related to computing, artificial intelligence or semantic web as opposed to a number of other examples, like (Henze, 05). That way we guarantee that the teacher/annotator was a novice to ontologies and knowledge representation. On the opposite, the chosen domain of Java programming for the second experiment is a very classical domain for this type of application. It allows us to compare this experiment with others

We compared two existing ontologies with the vocabulary defined for our Java course. The domain of Java programming is incredibly popular among e-learning researchers, certainly because most of them actually teach this subject. It revealed fairly easy to find existing ontologies freely available and addressing the same domain. Moreover, these ontologies use semantic web formalisms, which greatly help when comparing ontologies together.

They were developed in the context of research experiments about semantic web for e-learning: (Goble et al., 01) and (Henze, 05) and we downloaded them from the web.

➤ *Onto I*: The first ontology (Goble et al., 01) is the largest one. It is expressed in Daml+Oil but a straightforward translation was possible towards OWL-Full in order to respect up-to-date standards of the semantic web and thus facilitate comparison using actual tools.

➤ *Onto II*: The second one (Henze, 05) was an RDFS hierarchy of classes.

➤ *QBLS*: The QBLS conceptual model we developed is a structured vocabulary expressed in SKOS, the Simple Knowledge Organization Scheme. The hierarchical relation uses broader/narrower links. The SKOS model itself is expressed in OWL.

All three models are expressed in RDF, but comparison must be carried out carefully as the semantics of the primitives used are different (OWL, RDFS, SKOS). A reasonable solution is to associate specific semantics to the primitives according to the envisioned use of this knowledge:

➤ We compare classes in OWL and RDFS with SKOS concepts

➤ For the structure, the subsumption can be compared to broader/narrower links in SKOS.

From a strict semantic point of view the different structural relations do not carry the same semantics, but if we consider the intention behind the creation of these models it makes perfect sense: the ontologies have been designed for learning systems whose philosophy is very close to QBLS. They aim at supporting and enriching navigation over a set of web resources about Java (the Sun Java tutorial to be precise). The design intention behind the use of subsumption properties is to link concepts to offer "conceptual" navigation paths. The QBLS vocabulary shares the same goal: its organization intends to offer coherent navigation paths. Table 6 summarizes the above information about the three models.


Table 6 - Comparison of different domain ontologies for Java programming

	Onto I	Onto II	QBLS
Meta-model	OWL	RDFS	SKOS
Number of concepts	283	194	164
Semantic expressivity	subClassOf, restriction, intersection, union	subClassOf hierarchy	Broader/narrower Hierarchy

5.4.2.3. Conceptual coverage

To evaluate the overlap in term of coverage of domain concepts, we manually determined the percentage of overlapping concepts. Each line on Table 7 indicates the percentage of concepts that can find a match in each ontology (by column). A match is considered possible each time a concept (in line) and its counterpart (in column) identify the same thing or a specialization of it (e.g. “statement” in the first ontology and “assignment statement” in the other).

Table 7 - Concept overlap between the different ontologies

compared to 	Onto I	Onto II	QBLS
Onto I	100%	56%	57%
Onto II	71%	100%	73%
QBLS	42%	42%	100%

The overlap between Onto II and QBLS is quite high (73%), but this high value must be interpreted carefully. We knew about this ontology at the time of defining the QBLS vocabulary and it inspired the definition of our original concepts. The real high result is obtained with the comparison between the two ontologies Onto I and Onto II: 71% of the concepts of Onto II have a counter part in Onto I. It is difficult to establish a threshold, but with this value, reusing ontology Onto I in the tool associated to Onto II should be possible from the point of view of the concept coverage.

Looking at QBLS it appears that we have defined rather specific concepts as only 42% of the QBLS concepts find a match in Onto I or Onto II (third line). This value is definitely too low: less than one concept out of two has a counter part. Moreover having a counter part does not always imply rich conceptual paths in the application as several concepts often project on a single one. This difference can be easily explained: Onto I and Onto II have been defined to annotate roughly the same set of resources, which is the famous Java tutorial from Sun (there is quite a few years between the two publications so the material must have evolved in the meantime). Whereas QBLS vocabulary targets another set of resources, it then appears to be equally distant from both other ontologies.

We take this result as a strong argument demonstrating the importance of the targeted material for the conceptualization of a domain to learn, and thus the *impossibility to rely only on existing models defined for different documents*.

5.4.2.4. Structural aspects

Ontologies are formed by concepts but also relations between them. The structure brought by those relations is used to guide learning paths, make inferences to retrieve interesting resources depending on a context, etc. (see 6.3). Therefore, the structure of the domain ontology models how the system organizes the concepts to make them easier to learn. It has a strong pedagogical value. By comparing the ontologies, we actually compare the pedagogical approaches. For example, the concepts of “public”, “private”, “protected” in Java are defined

as subclasses of “method modifiers” in Onto II, whereas in QBLS they are narrower concepts of “access rights”. Both models are correct, but this introduces a fundamental conceptual difference in the way the domain will be learned using the ontology.

We evaluated the overlap of the conceptual structures by loading the result of the concept matching above into a semantic search engine (Corese). Using this engine, we can query for graph patterns (see chapter 6). In particular we look for following pattern: let ?x1 and ?x2 be two concepts of a first ontology, and ?y1 and ?y2 two concepts (may be identical) of a second ontology. Let also suppose that ?x1 is a subclass of ?x2, and ?y1 is a subclass of ?y2. If ?x1 matches ?y1 and ?x2 matches ?y2 then we decide that the subsumption or broader relations can be matched together.

In the SPARQL language such a pattern can be expressed by the pattern shown figure 51.

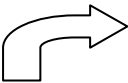
```
10. ?x1 ~ co: . ?x2 ~ co:
11. ?y1 ~ pr: . ?y2 ~ pr:
12. ?x1 edu:match ?y1
13. ?x2 edu:match ?y2
14. ?x1 rdfs:subClassOf ?x2
15. ?y1 rdfs:subClassOf ?y2
```

Figure 51 – SPARQL query for structure matching

The results of “matched” subsumption links are presented Table 8. Numbers count the arcs in the ontology in line that can be matched to the ontology in column. These numbers give us a good hint of the similarity between the navigation paths if using a different ontology.

The structure of the QBLS domain vocabulary is quite limited, so few matches can be found. We tried to exploit the additional links deduced from the annotations (see 5.3.4). With this structure, a larger number of relations can be considered. Of course the matching is only performed on the structural level.. The results of the match are indicated in the parenthesis on the QBLS line and column.

Table 8 - Match for hierarchical links

compared to 	Onto I	Onto II	QBLS
Onto I	XXXXXXXXXX	32	7(+14)
Onto II	1	XXXXXXXXXX	4(+3)
QBLS	1(+2)	12(+2)	XXXXXXXXXX

The very low numbers obtained show that the three models have very little in common from a structural point of view. Best result is obtained between Onto I and Onto II. However, Onto I defines 1185 subsumption relationships in total (actually only 347 are explicitly defined, the others come from the transitive nature of subsumption). From those figures we can definitely conclude that despite a similar domain of learning, the different models have little in common in an application perspective.

Automatic ontology matching mechanisms have not been considered here since the goal of this comparison is to show the differences between the ontologies, not the possibility to match them automatically. The tedious manual process guarantees that the obtained match is the best possible. A few ones might have been missed but certainly not in a significant amount for the global results.

To conclude this short discussion on reusing ontologies, existing ontologies present an interesting value for the identification of concepts, but the exploitation of the structure seems

really application specific. The pedagogical role played by the structure cannot be easily transferred from one learning context to another.

5.4.3. Teachers' role and learning technologies

5.4.3.1. Putting expertise forward

In the introduction, and as explained in (Villette, 99), we exposed the deep modification of teachers' role in e-learning compared to classical teaching. What teachers are now supposed to focus on is the conception of the learning activity and the "coaching" of the students. However, teachers are not the recipient for knowledge anymore. They must act as guides helping individual students along their learning path. This vision is the result of a long process aiming at using content as the primary mean of knowledge transfer compared to classical teacher to student relationship. Originally this was rooted in the idea of distance education, where the face to face is not possible. The benefits of the approach were soon extended to a large palette of teaching/learning situations.

This philosophy finds a perfect expression in the semantizing process described in this chapter:

- First, the teacher really puts the content forward compared to face to face teaching. Learning resources are not just one of the pedagogical tools, placed at the same level with the blackboard, or the projector. Students are directly facing course content and the teacher comes "from the side" as a facilitator. This facilitator role is prepared by annotating the course.
- The semantization should not try to impose a "perfect" conceptual model upon the student but just to help him/her understand the actual material (as opposed to approaches based on "pre-requisites"). This illustrates the crucial role still played by the practitioner. We showed that external models would be difficult to reuse and that they have to be created on purpose. This is a new role of "knowledge workers" for teachers. They have to explicit with digital tools what they previously tried to transmit directly to their students.
- In addition, the pedagogical expertise is offered more directly to the students. The role of pedagogical expert is emphasized in this method, confirming the prediction of (Villette, 99).

Globally the teacher's role switches from applicator and effector of knowledge transfer to expertise provider. The semantizing process helps putting this expertise in practical terms.

5.4.3.2. Link to Learning Design

In recent approaches, like the work on learning design standards (LD), the idea has emerged that the learner activity is more important than the content manipulated. This research stream was inspired by the feeling that activity aspects of learning have been largely neglected in e-learning. The semantization method answers this criticism by devoting its first task to the identification of the strategy in which the content will be involved.

In approaches based on the formalization of learning designs, we must not forget that the main communication channel in e-learning is done through screen displays, which means pedagogical content. Unless the planning of activities, or learning designs, does not involve any pre-existing source of knowledge, pedagogical content will continue to play an important role as a mediator between the learner and the system. The rationale of the proposed method is to place content, and knowledge, at the centre in the computer assisted learning scenario. The on-line course generated is only one aspect of the training. Activities described in LD could very well involve the use of the on-line semantized course. This is by no means an exclusive view of organizing learning, but it covers one very important aspect.

5.4.4. Redefining conceptual navigation

Due to the specificities of the proposed annotation method, the variety of annotations produced is restricted. Knowledge formalized in this context follows specific patterns, which impose some design constraints on the system where the resources and the annotations will subsequently be used.

The design choices stated in chapter 4 encompass several possibilities for conceptual navigation using ontological representations linked to annotated resources (see 4.4.2). With the specificities of the annotations obtained through the proposed method, we can precise our choice of conceptual space and navigation.

We use the term of “conceptual navigation” in the sense of (Crampes et al., 00) and (Brusilovsky, 03) to identify user exploitation of the semantized content. This navigation happens in a space that can be represented as a graph. In the following example, we detail step by step the effect of semantization on a linear document. For easing comparison with the models presented in chapter 4 the same convention for representing resources and concepts are kept.

5.4.4.1. Starting from the linear structure

The document is first considered as a linear suite of resources (see figure 52). This vision “slices” the original document (Buffa et al., 05). Typically, in slide shows, each slide constitutes a resource, and the only path available is the linear succession of resources.

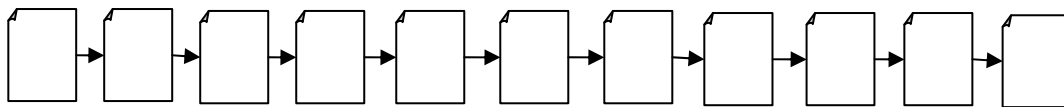


Figure 52 – The original document: a linear suite of resources

5.4.4.2. Concept annotation

The resources are contextualized by the introduction of additional knowledge. This knowledge relates to several points of views, but a single graph must be built from those different points of view (pedagogy, domain, etc.). Figure 53 illustrates this contextualization with the introduction of concepts. The concepts situated above the resources are domain concepts (taken from the example on signal analysis) and linked to the resources by the inbound and outbound relations “dc:subject” and “rdfs:seeAlso”. For a domain expressed in SKOS the relations “skos:primarySubject” and “skos:subject” could also be used. Pedagogical concepts at the bottom assign a pedagogical “role” (or type, with “rdf:type” relations) to each resource.

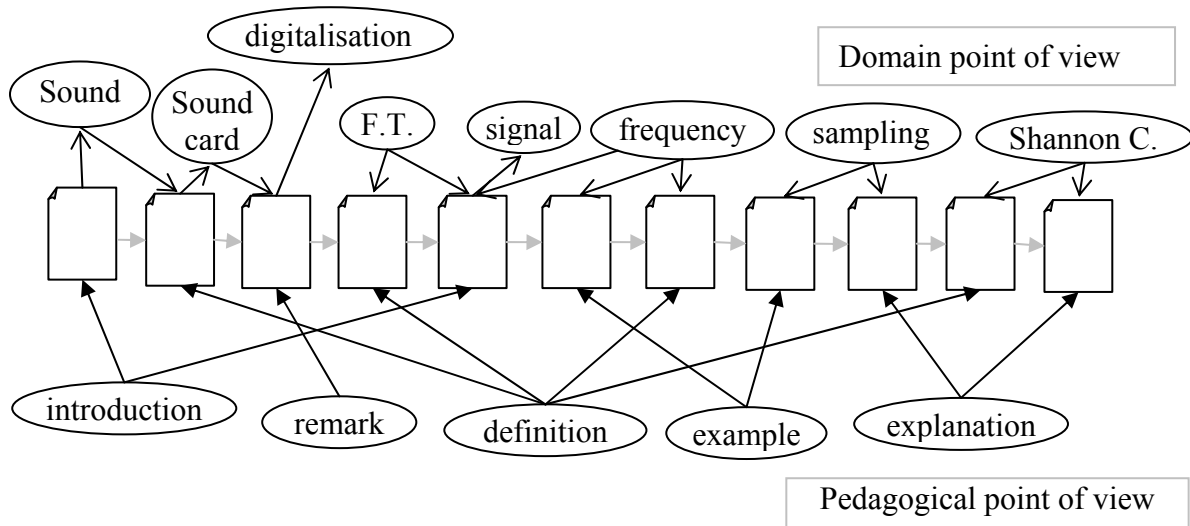


Figure 53 – Contextualization according to domain and pedagogical points of view

5.4.4.3. Domain concept structuring

The introduction of the concepts with inbound and outbound relations to resources opens new paths for navigation. In QBLS-1, we limited ourselves to this level, ensuring that the resources could be reached from the start using such navigation.

Structuring those concepts allows us to create more conceptual paths. The structure would “distort” the original linear disposition into a graph with more navigation paths. Figure 54 shows how the structural links between domain concepts (in thick black) “organizes” the concepts so that the linear way (still visible in light grey) is distorted.

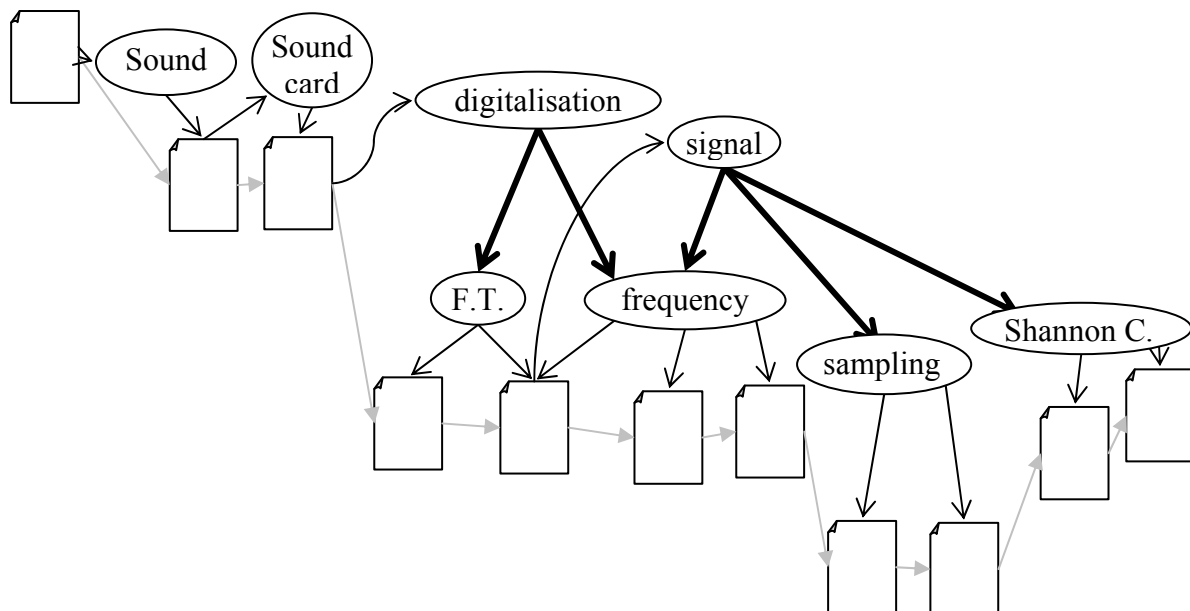


Figure 54 – Structuring knowledge to create conceptual paths in the content

5.4.4.4. Pedagogical concepts structuring

The above illustration shows the structuring of the domain. However, other points of view must be considered, like the pedagogical one. The pedagogical concepts are organized in a hierarchical structure that also contributes to the definition of navigational paths.

A simple example of such mechanism can also be represented on a graph like figure 55. Two resources are annotated by two different concepts from a pedagogical ontology. The exploitation of the semantic relations in this structure (subsumption in thin lines, or transversal relations in thick arrow) allows us to build a path that links the two resources (dotted arrow). Such paths are obviously combined with the previous domain guided behavior and lead to the realization of a conceptual space, where the original linear structure has disappeared.

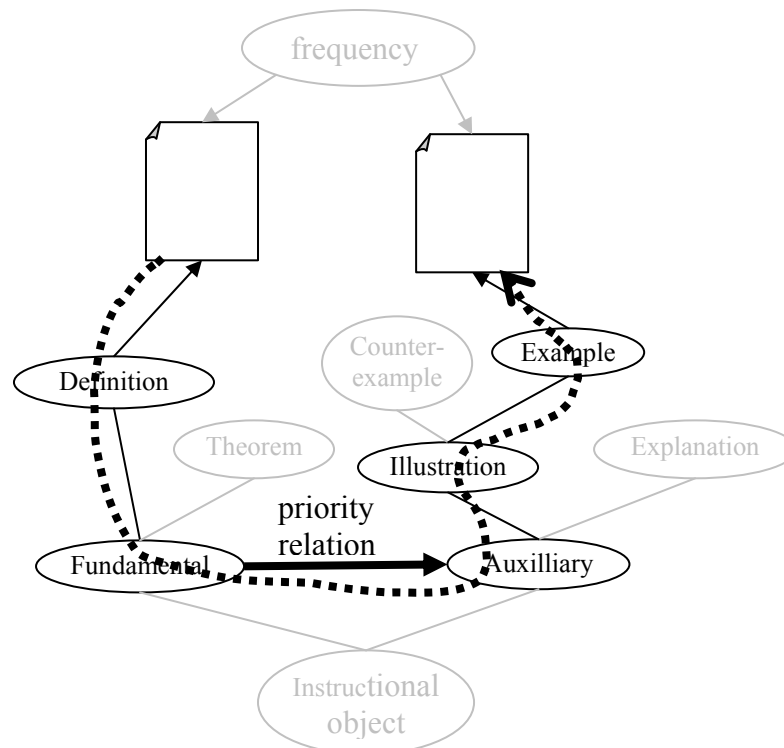


Figure 55 – Impact of pedagogical knowledge on path creation

5.5. Conclusions

In this chapter, we present a method for reusing digital course material. Our leading thread is to explicit the conceptual structure of the course from both the domain and the pedagogical point of view through annotations on the document itself. The annotation is specific, and oriented towards a unique vision and learning goal (decided upon at the beginning). It aims at giving teachers as much support as possible in the deployment of a course in a learning system offering conceptual navigation. The various types of conceptual navigation envisioned in the design phase can be supported by the knowledge expressed using this method. Navigation using rules would still need to express those rules independently.

The use of existing text-editing tools is emphasized, as well as the visual representation of annotations. Both characteristics make this quite original and strongly user oriented. The drawbacks appear with a more restricted freedom for annotation, and limitations in the tools themselves. The growing convergence towards more XML integration in large audience tools should solve some of those issues in the future.

The detailed method is completely generic for a large variety of courses that need to be exploited with conceptual navigation. The proposed method also perfectly fits within the context of the semantic web. The use of standards for knowledge representation (RDF, RDFS, OWL) makes it easier to integrate external sources of information (like the reused pedagogical ontology) and allows us to envision the application of this method in many different contexts

of course reuse. The necessary changes in the existing stylesheet should be minor and the output can remain in the standard language. This work at least shows the possible integration of external sources of knowledge using semantic web standards.

In the next chapter, we present how the generated annotations are used in practice in a navigation interface and how the different aspects of conceptual navigation are proposed using semantic web technologies.

6. EXPLOITING SEMANTIC WEB TECHNOLOGY 125

6.1. <i>QBLS: A Semantic Web infrastructure</i>	126
6.1.1. Architecture	126
6.1.2. Interaction cycle	127
6.2. <i>Semantic Interfaces</i>	128
6.2.1. Course interface.....	128
6.2.2. Annotation editor.....	132
6.3. <i>Semantic querying for learning interaction</i>	134
6.3.1. Semantic querying on domain knowledge	135
6.3.2. Including pedagogical knowledge.....	137
6.3.3. Retrieving literal information.....	142
6.3.4. Context awareness	143
6.3.5. Domain/range querying.....	143
6.4. <i>Semantic query processing</i>	144
6.4.1. Corese specificities	144
6.4.2. Dynamic aspects.....	145
6.4.3. Rule standardization	145
6.5. <i>Interface generation</i>	145
6.5.1. Integration of XML technologies	145
6.5.2. Dynamic pages	146
6.5.3. Detailed example of interface construction.....	146
6.5.4. Editor interface	148
6.5.5. Evaluation of the technical solution proposed	149
6.6. <i>Conclusions</i>	149

6. EXPLOITING SEMANTIC WEB TECHNOLOGY

In this chapter we expose how standards proposed for the “Semantic Web” can be effectively put in practice in the context of a learning system offering conceptual navigation in course content. We propose directions for this “operationalization” of knowledge from a theoretical, a technical and a practical point of view. This work is strongly related with the development of the QBLS system, which provides illustrations and examples for this chapter.

We first present the major interaction principle of QBLS. Simply put, user interacts with the system through an interface that interprets user actions (clicks on links) as semantic queries. Then we present the interfaces developed for the QBLS system. The third section details how the semantic queries are built incrementally, and what kind of formalized knowledge is involved. The forth section presents the specificities of the semantic search engine Corese that performs the queries. Then a last section is dedicated to the construction of the interfaces. In particular, we show how they are dynamically generated from a semantic search results. The solutions brought by the semantic web approach are highlighted.

Finally, we draw some conclusions about the use of semantic web technologies for e-learning in general and for building on-line course access systems in particular.

Technical details about semantic web technology are examined and this chapter requires some prerequisite knowledge about RDF and OWL for a comfortable and complete understanding.

6.1. QBLS: A Semantic Web infrastructure

6.1.1. Architecture

Our technical contribution in this thesis is to demonstrate the effectiveness of the development of an e-learning course access system (and an on-line access to resource content in general) based on the integration of a semantic search engine at the centre of the architecture.

The QBLS system relies on a client/server architecture using a standard browser on the client side. It is a usual architecture on the World Wide Web today. Most computing operations are carried out on the server side. The client just displays the interface from the XHTML code returned by the server.

On the server side, the backend of the system is supported by the Tomcat web server. It is a standard Java servlet and JSP container offered freely by the Apache foundation. The application itself is hosted on the server as a single “web-app” (for web application). It runs a manager program that fires up an instance of the Corese search engine (Corby et al., 04). A set of servlets is in charge of the various services offered by the platform (edition of annotations, resource upload, etc.). JSP pages provide the dynamic interfaces. The application repository also hosts the learning resources in XHTML format, the annotations and ontologies in RDF, as well as the various XSL stylesheets used to manipulate the content (extraction and display).

In addition to the centralization of all the data in a single place, which facilitates evolution and monitoring, this approach also ensures a maximal interoperability for “clients” as communication solely relies on the HTTP protocol and XHTML standard.

Figure 56 shows the different programmatic components of the system architecture (server/application/search engine) and the resources (formalized knowledge and content). Black arrows indicate the processing of a user request. The different formats involved in the communication between the components are indicated in parenthesis. It highlights the importance of W3C standards in the development of such architecture.

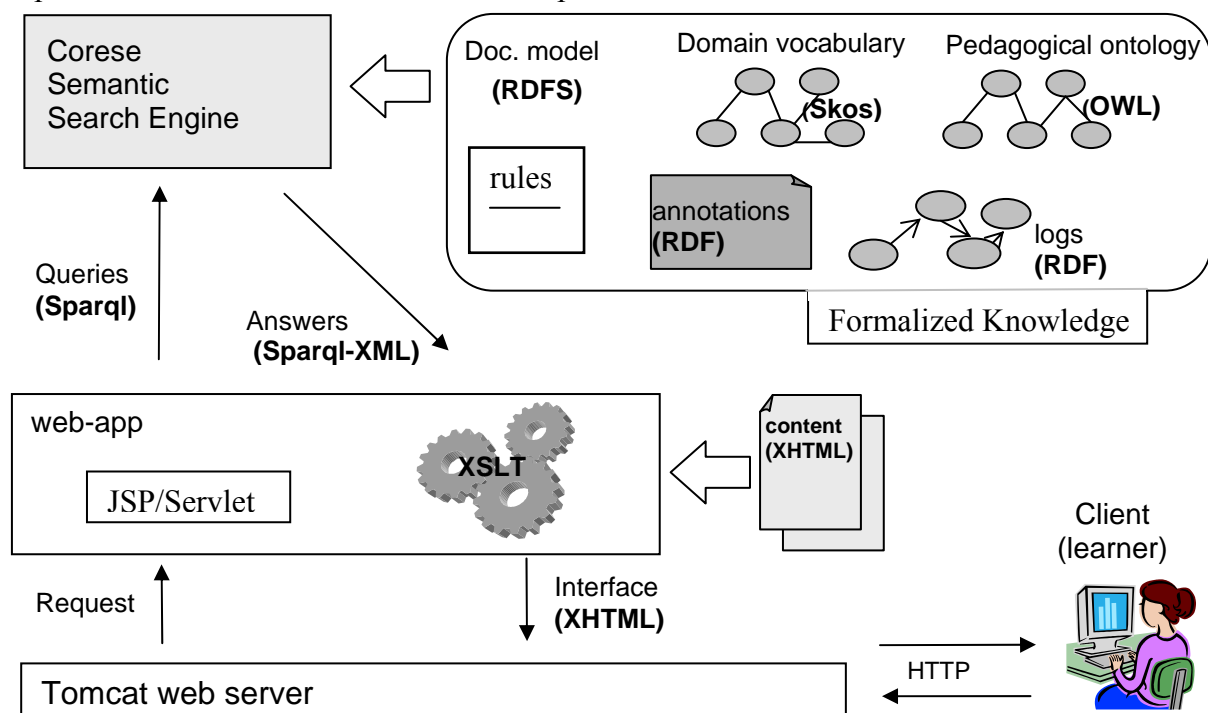


Figure 56 – Web-application architecture for learner access

We present this architecture as a *generic pattern for the integration of semantic search engines in knowledge intensive systems*. The access to knowledge, formalized using standards, is performed through a single entry point, and a standard querying interface (here based on the future W3C recommendation SPARQL (SPARQL, 06)). If replacing the client by another program, this architecture could also perfectly suit the deployment of a web service. The Corese engine in this environment offers a practical and efficient realization of a “*semantic middleware*”.

6.1.2. Interaction cycle

The main interaction principle we propose between the learner and the system is presented in figure 57. It is based on a request/answer mechanism:

- First, users read some resource content using the QBLS learner interface. When needed, they can send a request for additional information, for example about a specific concept. Learners express this request by acting on the interface (typically by clicking on a proposed link).
- The browser interface transmits the request to the server through the HTTP protocol. The server then analyses the request and constructs a semantic query to answer it.
- The query is then processed by a semantic search engine running on the knowledge base.
- Finally, the application constructs a result page from the query result and sends it back to the learner through an HTTP response.

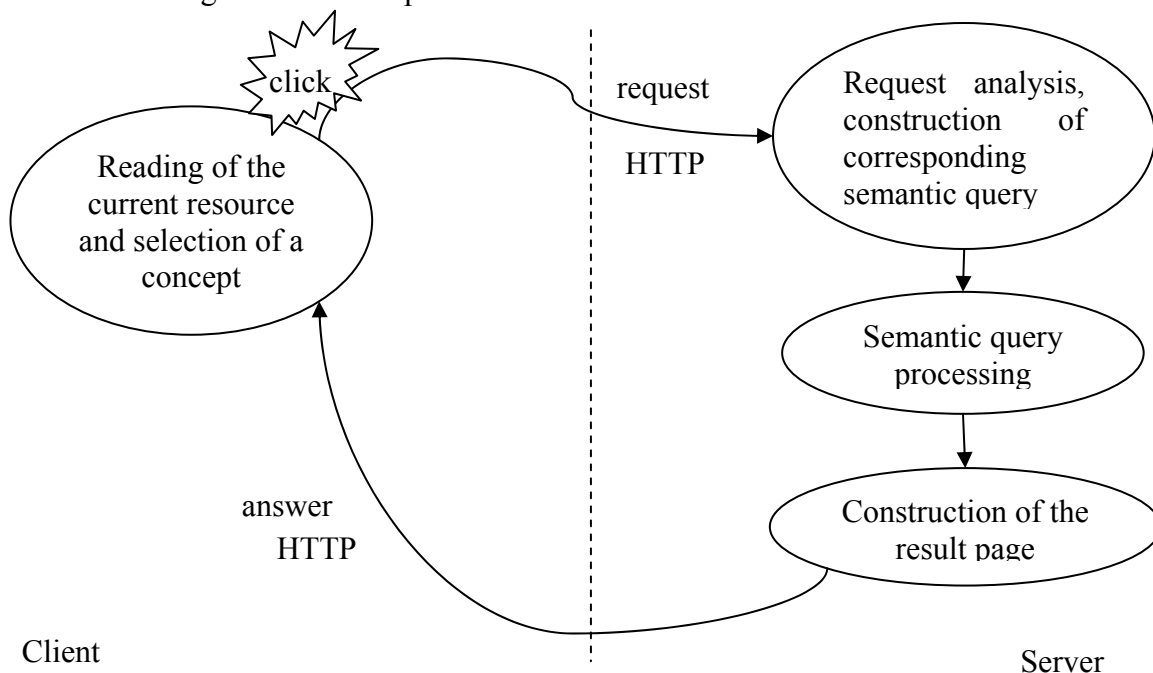


Figure 57 – Interaction cycle between the learner and the system.

Technical manipulations of semantic information are conducted on the server side. This interaction scheme follows the classical client/server pattern and the philosophy of the HTTP protocol. It is a “pull” mechanism (as opposed to “push” approaches): information is sent to the learner only on his/her request.

- From a pedagogical point of view, this allows students to follow their own rhythm. It offers freedom and a feeling of “control” over the navigation choices.
- From a technical point of view, it builds on the current solutions for the web and thus it is well suited for deployment in today’s environments.

The organization of the remaining of this chapter follows this interaction cycle:

- Display and navigation in the interface that triggers a request, 6.2.
- Construction a semantic query in answer to a user request, 6.3.
- Processing of this query, 6.4.
- Construction of an interface, 6.5.

6.2. Semantic Interfaces

6.2.1. Course interface

In this section, we present the QBLS interfaces for learners and explain how they allow him/her to trigger new queries by interacting with them.

6.2.1.1. Design choices

Our major issue when designing interfaces for the QBLS system was to reflect the underlying conceptual model in the interface itself, while hiding the complexity of the model to facilitate learner understanding. In the following, we explain why we made the choice of a hypertext based interface and we underline our scenario based design approach.

Hypertext navigation

On the web, documents are usually presented under the form of hypertext. Existing systems, like AHA (de Bra et al., 03), Metalinks (Murray, 03) illustrate different hypertext interfaces for accessing a course. In the case of AHA, the interface is adaptive and offers more than a static hypertext. Using hypertexts allows users to reach resources quickly in a non-linear mode of representation of a document. In static hypertexts, a network of links forms the navigation structure.

To use the hypertext correctly, users must construct their mental model of the document they navigate. This mental representation may reveal difficult to build and may become a source for disorientation: when not knowing the current location, how to get to a point etc. (Otter and Johnson, 00).

The user must perform specific cognitive operations: reading and understanding the information contained in a node of the hypertext, identifying the available links and select one of those links to progress towards another “node”. Such operations can induce a cognitive overload (Conklin, 87). Overload happens when several information pieces must be processed in parallel. For example, for information retrieval tasks the user must maintain its initial goal and understand the content of the pages as well as the relations between each page.

Given the popularity of hypertexts and despite the identified difficulties, we made the choice of this form of interface. The underlying “conceptual navigation” will help in addressing those difficulties.

A scenario centred interface

The interface must help users in following a specific scenario. In QBLS learners have to answer questions. In that scope, we decided that the current question should remain visible while navigating in the “conceptual space”.

Once this basic scenario is supported, the interface must also be intuitive, facilitate the localization, and allow users to access the desired information in a reasonable number of clicks. To ensure all those characteristics in the QBLS interface, two classical techniques of ergonomic design have been deployed. In (Tricot et al., 03) those modes of evaluation are described as inspection (for example with the heuristic evaluation) and empirical (user test

with a small sample of users). A list of existing heuristics grids for evaluation is given in annex 1. The application of both techniques guided our design choices.

6.2.1.2. Interface presentation

The interface plays an important role in the user navigation choices. To introduce the following sections we present here the interface developed for QBLS-1 and QBLS-2.

QBLS-1

Figure 58 below, shows the interface of the QBLS-1 prototype as presented to the learner on his/her first access to the system.

The user is first presented with the concept of “course”, the top-level concept representing the domain dealt with during the session. The “course” concept contains two resources: a list of questions (visible) and a summary of the main course topics (tab header on the right).

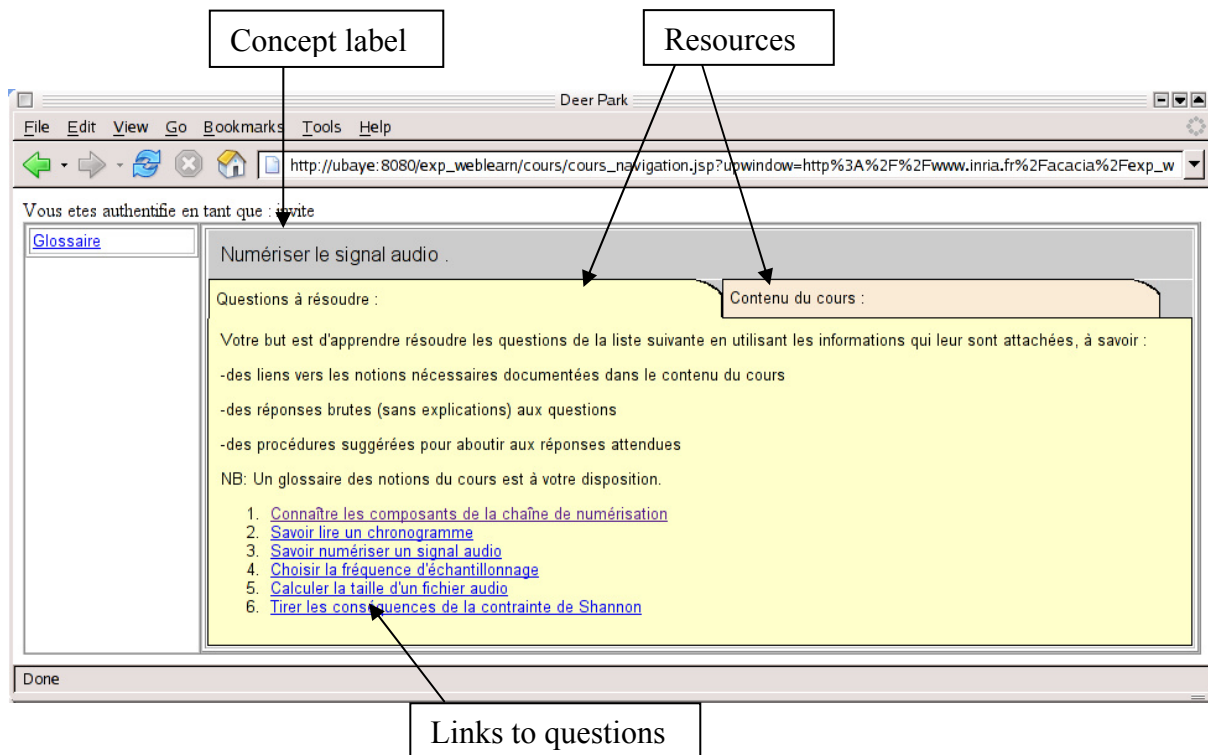


Figure 58 – Screenshot of the QBLS-1 interface, at the beginning of the learning task

The navigation model transpires through this interface: The visited concept is materialized by the grey frame and resources associated with this concept (in fact returned by a query on this concept) appear as tabs within this frame. Links in the content of a resource point to other concepts.

By selecting a specific question (i.e. clicking on the proposed links), the concept representing this question will replace the course concept in this window. An example of this situation is shown in figure 59. The first resource appearing for a question is the formulation of the question (“Enoncé”). This formulation contains links to specific domain concepts. Clicking on those links opens a second frame below the first one, where specific domain concepts are displayed. The layout of the bottom frame follows the same pattern as above. A history list for domain concepts is also available on the right.

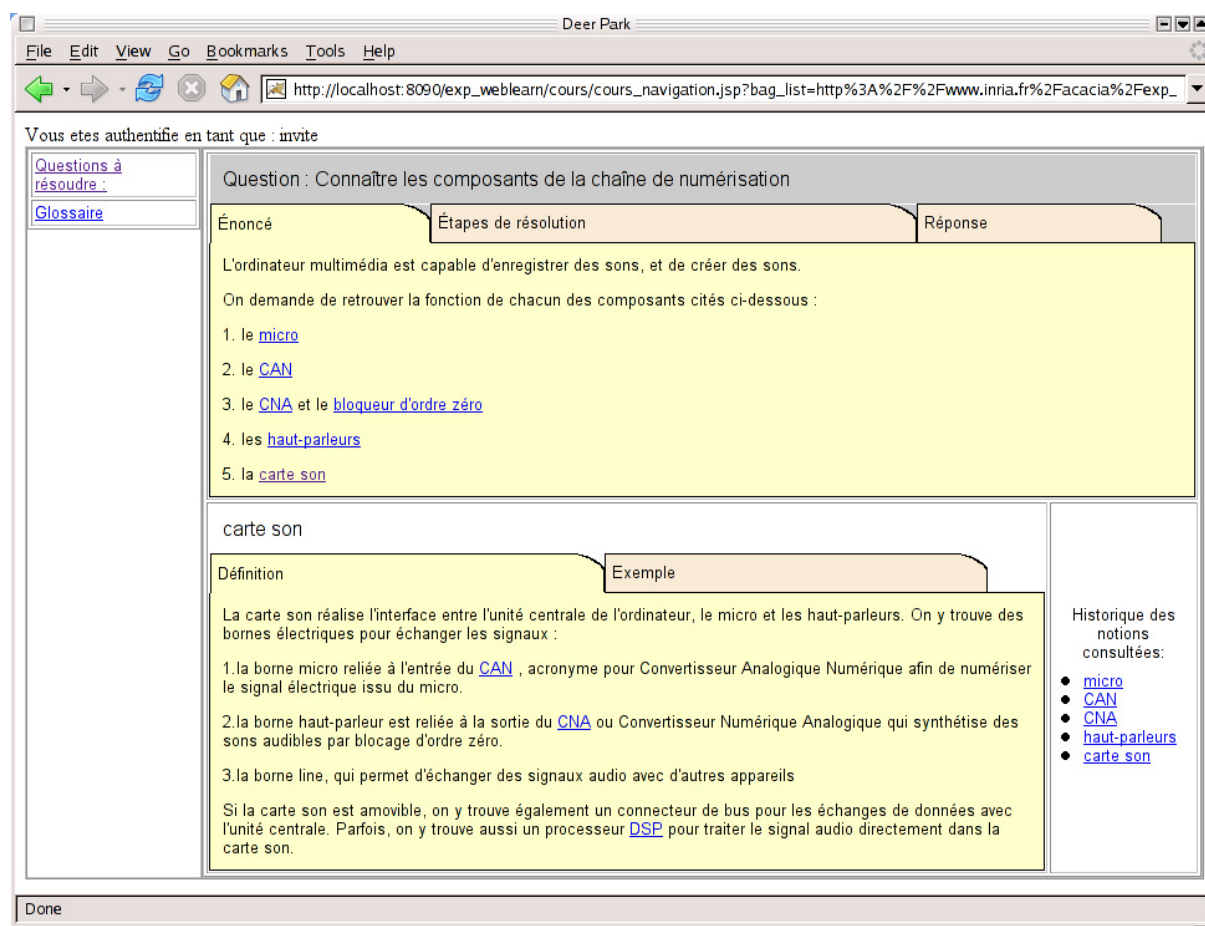


Figure 59 – Simultaneous display of a question and a domain concept

This interface is very specific to the implementation of the QBLS-1 scenario. It has been designed specifically for this scenario. The contribution of this work to the scientific project lies in the design phases and choices that lead to this interface and in the technical implementation based on semantic web technologies, developed in this chapter.

QBLS-2

The interface of the QBLS-2 experiment is presented in the screenshot in figure 60. The window can be divided in four zones:

- A banner at the top indicating the current chapter.
- A central area displaying the resources relative to a concept. On this example, four resources are presented, relative to the concept of “*Object*”. Resources are organized in tabs, with the header of the tabs always visible.
- The right part contains a list of previously visited concepts.
- At the bottom of the screen, a “back” button allows to go back in the navigation history.

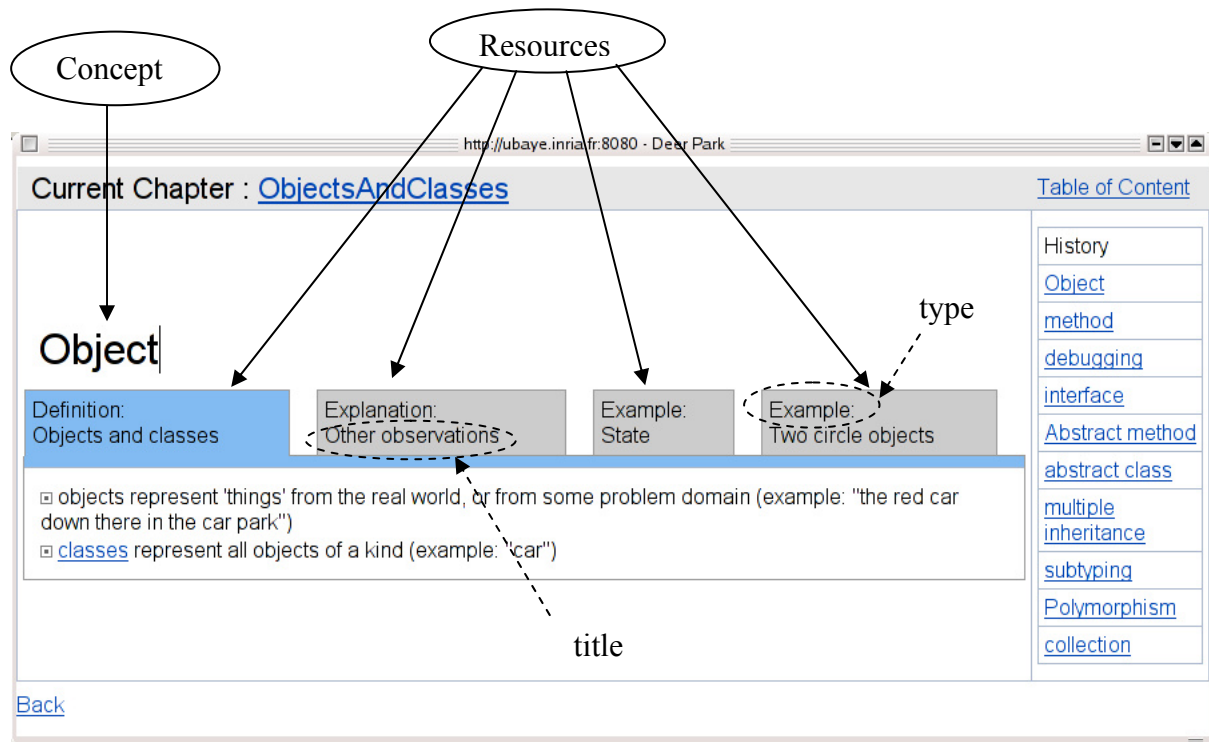


Figure 60 – Screenshot of the learner interface for navigating a Java course

This interface does not present the questions in the same window as the QBLS-1 experiment because questions are authored separately from this system. The look and feel is also different to suit the scenario. The comparison shows that both interfaces follow the same navigation model, but each one is closely linked to its intended use scenario.

6.2.1.3. Adaptive functionalities

A particular interest of semantic interfaces in this context is the potential integration of various sources of information into a single display. For example, figure 60 mixes information about the resources (titles, types) the user history (list on the right) and the context indicated by the current chapter (at the top).

In this interface, it is possible to reproduce what adaptive systems implement. For example, the QBLS interface proposes a feature of “link hiding”: It hides links that are not relevant in the current context. The decision to hide links is based on the contextual information of the chapter being studied.

Such adaptation is necessary because some concepts appear in a resource, but are only described by resources belonging to other chapters. If providing a link toward this concept, it would result in leading the learner out of the current chapter. This is not advisable for introductory teaching. Later on, when the system is used to find previously learned information, such adaptation can be switched off.

The use of chapter information in particular is just an example, many other information could be thought of to decide on link hiding (user history, preferences, time, etc.).

Such dynamic functionalities of hypertext are not specifically related to learning. They can apply to any kind of dynamic hypertext. What our experience shows is that true challenges now concern the design of dynamic mechanisms with regards to real learners. Many adaptations can be implemented in the QBLS system based on the expressed knowledge. However, we have no clue which dynamic aspect is really interesting in our teaching context.

6.2.2. Annotation editor

In this section, we present another interface of QBLS developed for manual edition of RDF files. It proposes guidance to help teachers modify the knowledge base directly.

6.2.2.1. Need for edition

The previous chapter presents a method for automatically extracting annotations from a learning resource. However, additional knowledge needs to be defined. This includes for example the list of students with their id, the SKOS vocabulary, etc. It also offers a way for teachers to modify the knowledge about the course while students are manipulating it, without having to go back to the annotated document. This might be used to make content more dynamic.

All the above tasks need the use of an annotation editor. It is obvious that RDF files cannot be edited directly by teachers. It would require too much knowledge of the RDF/XML syntax. To solve this problem we developed a generic on-line annotation editor based on interesting technical paradigms. They allow us to build a dynamic interface using semantic web technologies.

6.2.2.2. An on-line RDFS editor

The editor is built on the following principle: it takes an RDF/XML file as input, and processes it through an XSL stylesheet to generate an HTML page where every RDF element (node, relation, literal) in the file is a dynamic link. By clicking on an element, another page is generated. It contains a form allowing users to edit the selected element. The form is dynamically generated to take into account the ontology, and the context in which the element appears. This interface also follows the interaction cycle presented figure 57, except that it addresses the teacher instead of the learner.

The philosophy of the interface is to reflect exactly the RDF/XML structure and that each widget, label, etc. is active and can be modified. Action buttons (X,+,-) give the opportunity to suppress or add complete branches of the tree. When clicking on a widget displayed in the interface, all other options are greyed and the user can safely modify what he/she wants.

In this interface, the range and domain properties of the relations are interpreted as constraints. The user cannot modify an element to violate such constraints. In theory, this is not the right interpretation for domain and range relations: in RDFS every statement is true. Domain and range only complete resource type definitions. However, in practice it is convenient to limit the choices within domain and range constraints. It indicates correct “patterns” of annotation. For information, it is also the choice made by the OLR3 editor (Kunze et al, 02b), similar in its philosophy but not in its implementation.

The screenshot in figure 62 shows the edition of an annotation on the concept of sound card (“*carte son*”) with the resources attached to it. This annotation of sound card has a definition, a label and an illustration. The definition and the illustration are attached differently to the concept:

- The definition is defined in the same annotation file. This allows a direct edition of the content for advanced users.
- The illustration is declared in a separate file and linked to the concept instance using its URI. In the editor, instead of displaying the URI of the illustration we display the title of the instance for better readability.

This difference directly reflects the RDF syntax shown in figure 61.

```

1. <rdf:RDF>
2. <edu:Notion rdf:about="http://www.inria.fr/acacia/exp_weblearn#id2248243">
3. <edu:entete>carte son</edu:entete>
4. <edu:definition>
5. <edu:Definition rdf:about="http://www.inria.fr/acacia/exp_weblearn#defid2248243">
6. <edu:titre>Définition :carte son</edu:titre>
7. <edu:contenu>&lt;p ...&gt;</edu:contenu>
8. </edu:Definition>
9. </edu:definition>
10. <edu:exemple rdf:resource="http://www.inria.fr/acacia/exp_weblearn#id2248330" />
11. </edu:Notion>
12. </rdf:RDF>

```

Figure 61 – RDF annotation about “carte son”

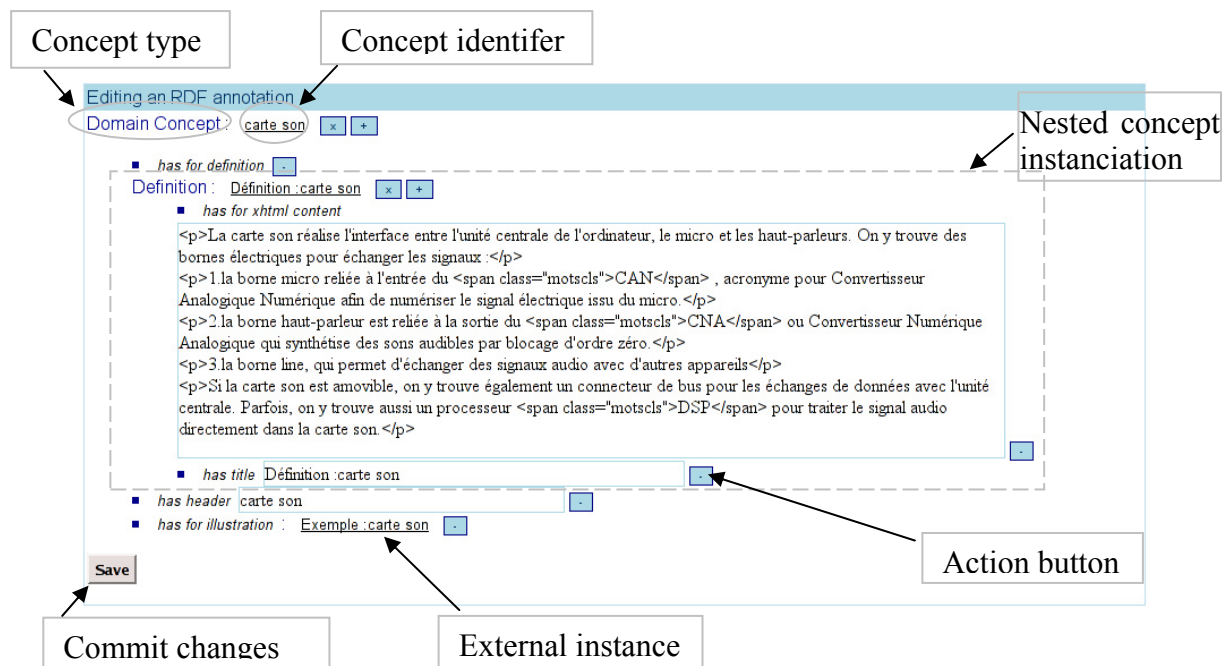


Figure 62 – Detail of an annotation in the editor

6.2.2.3. Example of edition

This tool is not primarily intended to be an authoring tool: we explained in chapter 5 how annotations on the course should be extracted from learning material. However to make content more dynamic, the teacher might want to remove or add links during the course without having to edit the annotated file.

For example, if the teacher wants to add an answer to a question at run-time, he/she can edit the annotation of the question. The interface figure 63 is presented to him/her when he/she wants to add a new relation to the question. The selector is dynamically built from a semantic query to the engine. The teacher selects the relation he/she wants and its expression in RDF/XML is added to the annotation file. The remaining of the annotation is greyed to prevent the edition of other elements while this process is not finished.

Choose a relation

Key Problem : Connaître les composants de la chaîne de numérisation

- has for item
- Test Item : Énoncé
 - has for xhtml content


```
<p>L'ordinateur multimédia est capable d'enregistrer des sons, et de créer des sons.</p>
<p>On demande de retrouver la fonction de chacun des composants cités ci-dessous :</p>
<p>1. le<span class="motscls">micro</span></p>
<p>2. le<span class="motscls">CAN</span></p>
<p>3. le <span class="motscls">CNA</span> et le <span class="motscls">bloqueur d'ordre
zéro</span></p>
```
 - has title Énoncé
 - has header Connaître les composants de la chaîne de numérisation
 - Descriptive relation ☒ add this relation
 - Descriptive relation
 - Auxiliary descriptive relation
 - has for illustration, has for example
 - has formalization, a pour formalisation
 - has for precision
 - has for procedure
 - has for solution
 - priority descriptive relation
 - has for item
 - has header

Greyed zone

List of choices

Figure 63 – Selection of a relation type in the editor

After adding the relation “has for solution”, the teacher is asked to select an existing question among the available ones, or to create a new one (“new...” option).

Select an ID

Key Problem http://www.inria.fr/acacia/exp_weblearn#sdqfsdf

- has for item
- Test Item http://www.inria.fr/acacia/exp_weblearn#qlqsodfi
 - has for xhtml content


```
<BR/><p>L'ordinateur multimédia est capable d'enregistrer des sons, et de créer des sons.</p>
<p>On demande de retrouver la fonction de chacun des composants cités ci-dessous :</p>
<p>1. le<span class="motscls">micro</span></p>
<p>2. le<span class="motscls">CAN</span></p>
<p>3. le <span class="motscls">CNA</span> et le <span class="motscls">bloqueur d'ordre
zéro</span></p>
```
 - has title Énoncé
 - has header Connaître les composants de la chaîne de numérisation
 - has for solution http://www.inria.fr/acacia/exp_weblearn#reponse6 OK
 - new...
 - http://www.inria.fr/acacia/exp_weblearn#reponse6
 - http://www.inria.fr/acacia/exp_weblearn#reponse5
 - http://www.inria.fr/acacia/exp_weblearn#reponse4
 - http://www.inria.fr/acacia/exp_weblearn#reponse3
 - http://www.inria.fr/acacia/exp_weblearn#reponse2
 - http://www.inria.fr/acacia/exp_weblearn#reponse1

Figure 64 – Selection of an instance reference in the editor

This generic editing tool is a good showcase for the potential of semantic “forms” in web interfaces. The logic behind the models is exploited to offer user guidance, and the complexity of the language syntax is partly hidden to the user. The novelty and interest of this example relies in the direct access given to the underlying formalism without manipulating the underlying syntax.

6.3. Semantic querying for learning interaction

This section details the construction of semantic queries to build a pedagogical interaction between the learner and the system. It presents four different aspects of querying we found important for building a navigation interaction on a semantized course. Navigation is not pre-computed. At the beginning but at each “step” a new query explores the possible directions

given the current context. A last part illustrates how queries are used to support annotations tasks in the editor presented above.

6.3.1. Semantic querying on domain knowledge

The annotations on the course provide information about several aspects relevant for a learning system (domain, pedagogy, etc.). The domain offers a conceptual view of the topics to learn and such view can be used to organize the learning paths. We present how such knowledge is used on each iteration of the cycle presented figure 57 to construct the proposed navigation.

6.3.1.1. Querying resource annotations in SKOS

The domain model is expressed using the SKOS meta-model (see 6.3.1.1). It is primarily composed of a list of domain concepts. A concept possesses several labels and concepts may be organized in a hierarchy of broader/narrower relationships. Figure 65 below shows an excerpt of the structured vocabulary used for QBLS-2. It shows the definition of three concepts (*Object*, *Field* and *Class*, line 2, 5 and 9). *Field* has broader concept *Variable* (line 7), and *Class* possesses two different labels (line 10, 11).

```

1. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns="http://www.inria.fr/acacia/java-
   skos#" xmlns:skos="http://www.w3.org/2004/02/skos/core#">
2. <skos:Concept rdf:ID="Object">
3.   <skos:prefLabel xml:lang="en">Object</skos:prefLabel>
4. </skos:Concept>
5. <skos:Concept rdf:ID="Field">
6.   <skos:prefLabel>Field</skos:prefLabel>
7.   <skos:broader rdf:resource="#Variable"/>
8. </skos:Concept>
9. <skos:Concept rdf:ID="Class">
10.  <skos:altLabel xml:lang="en">class</skos:altLabel>
11.  <skos:prefLabel xml:lang="en">Classes</skos:prefLabel>
12. </skos:Concept>
13. ...

```

Figure 65 – Excerpt of the SKOS vocabulary on Java programming

Domain expression and corresponding annotations for resources are encoded in the standard formalism RDF. QBLS can query this knowledge using the SPARQL language (see 3.3.2.2). SPARQL offers basic interrogation of triples. In the example presented below, triples are stored in the RDF annotation base generated from the semantization of a Java programming course (see Chapter 5).

A first simple domain query may ask for all the resources that have been annotated as being primarily about a concept. A query like in figure 66 will retrieve all the resources that have the concept of “*Variable*” as primary subject.

```

1. PREFIX skos:< http://www.w3.org/2004/02/skos/core#>
2. PREFIX <http://www.inria.fr/acacia/java-skos#>
3. SELECT ?doc
4. WHERE {?doc skos:primarySubject :Variable}

```

Figure 66 – A simple SPARQL query

The result of the above query is presented in figure 67 in the standard SPARQL/XML binding. It shows that two resources are linked to the concept of variable. Such result allows a

system to build a response page in answer to a request, through an XSL transformation for example (see 6.5).

```

1. <sparql>
2. <head>
3. <var name="doc"> </var>
4. </head>
5. <results distinct="false" sorted="false">
6. <result>
7. <binding name="doc">
8. <uri>http://www.inria.fr/acacia/prog101#chap9fic18.xhtml </uri>
9. </binding>
10. </result>
11. <result>
12. <binding name="doc">
13. <uri>http://www.inria.fr/acacia/prog101#chap8fic32.xhtml </uri>
14. </binding>
15. </result>
16. </results>
17. </sparql>

```

Figure 67 – Result of a SPARQL query in the standard XML binding

In the following examples we will refine this query to show:

- how the domain model can be further exploited (6.3.1.2),
- how pedagogical consideration can be introduced (6.3.2),
- how literal information about the concepts can also be retrieved (6.3.3),
- how a context of navigation can be introduced to adapt the results depending on this context (6.3.4).

In the following, the “PREFIX” header of the SPARQL queries will be omitted for better readability.

6.3.1.2. Exploiting the narrower/broader hierarchy

In some cases, using direct links between concepts and resources is enough to build a complete conceptual navigation space. For example in the experiment on signal analysis (see 8.2.1) the resource content offers enough direct navigation links towards the necessary concepts. It does not require structuring the vocabulary to offer more navigation paths. Queries like the one shown above will implement static domain guided navigation without ontological structure (see 4.4.2.3)

When the domain has to be structured by semantic relations between concepts, such relations must be exploited in the query. Typically, queries like in figure 68 exploit the “skos:broader” relation to extend the result of the query presented in figure 66. The returned resources may have for subject a more specific concept. In this example “*local variable*” is narrower than “*Variable*”, so resources annotated by “*local variable*” will also be returned in the result.

```

1. SELECT ?doc WHERE {
2. {?doc skos:primarySubject :Variable}
3. UNION
4. {?doc skos:primarySubject ?s ?s skos:broader :Variable}}

```

Figure 68 – Querying to include narrower concepts

The meta-model of SKOS is defined in OWL. The relation skos:broader is typed with owl:transitiveProperty. That defines the skos:broader relationship as transitive. In the above example, if a concept is linked to “*Variable*” through a succession of skos:broader relationships, the documents annotated with this concept will also be returned by the query.

The learner interface may propose not only the resources directly annotated with the requested concept but also resources annotated with narrower concepts. This enriches the possible navigation paths.

In other research works like (Henze, 05) or (Goble et al., 01), the domain is modeled through ontologies in RDFS or OWL. Inferences relying on the hierarchical structure of the domain concepts apply the same mechanism. Instead of the `skos:broader` relation they use the `rdfs:subClassOf`. With SPARQL queries and such RDF patterns we define *a generic mechanism for navigating resources annotated with hierarchies of concepts* that do not need to be hierarchies based on subsumption links specifically.

6.3.2. Including pedagogical knowledge

6.3.2.1. Querying pedagogical annotations

In our knowledge model, pedagogical knowledge is expressed in a pedagogical ontology where concepts express the pedagogical role of a resource (see 5.4.4.4). In the annotations, each resource has one pedagogical type. Such explicit pedagogical annotations may help learners in two ways:

- By displaying directly this information in the interface.
- By using it for automating the tasks of guidance and selection of resources.

For direct display, the pedagogical type can be obtained by placing a variable as object of an `rdf:type` triple in the body of the query: `?doc rdf:type ?type`.

To select a specific type, the variable can be replaced by an explicit URI taken from the pedagogical ontology. For example, figure 69 shows a query referring to such pedagogical knowledge. It looks for documents about the concept of “*Variable*” that are instances of the “*Illustration*” concept, or instances of subclasses of “*Illustration*” (e.g. “*Example*”, “*Counter Example*”, etc.)

<pre> 1. SELECT ?doc WHERE { 2. ?doc skos:primarySubject :Variable 3. ?doc rdf:type edu:Illustration} </pre>
--

Figure 69 – Querying including pedagogical knowledge

The query using domain knowledge in figure 68 and the query using pedagogical knowledge in figure 69 highlight a difference in modeling: The query on the domain concepts must explicitly ask for narrower concepts whereas for pedagogical knowledge, inference on the class hierarchy is performed as part of the semantics of the language.

In this conceptual design, the two models (domain/pedagogy) do not operate at the same level:

- Resources are represented as instances of pedagogical concepts. For example, documents that are instances of “*Example*” are also instances of “*Illustration*”.
- Resources are linked with domain concepts through “subject” relationships. If a resource has for subject “*Field*”, it does not mean that it also has for subject “*Variable*” but that when requesting resources about “*Variable*”, it is interesting to return also resources about “*Field*”.

This representation of domain and pedagogy can be proposed as a design pattern that *is well adapted to the envisioned query-based navigation in the context of learning*.

6.3.2.2. Scenario involving pedagogical knowledge

More complex behaviors using pedagogical knowledge have been deployed in existing systems like ((Crampes et al., 00) or (Henze, 05). Such functionalities rely on a precise scenario defining the role played by pedagogical knowledge.

In this section, we only address the common scenario where the order of presentation of the learning resources is computed according to their pedagogical type.

To enable the system to decide on the ordering of the resources, the teacher must express additional pedagogical statements expressing his/her chosen learning strategy. For example, he/she might express that “*Fundamental resources have priority over auxiliary resources*”.

This is expressed by the following RDF triple: edu:Fundamental edu:priorTo edu:Auxiliary. Such statement is then exploited by the system to supply the learner with the “fundamental” resources before the “auxiliary” ones. We detail next the complete mechanism that allows the system to propose this ordering.

6.3.2.3. Creating a relation order for semantic types

In the QBLS-2 example, 28 pedagogical types are defined and organized in a subsumption hierarchy. We present below a mathematical formulation to justify the creation of a relation order from a concept hierarchy and a set of priority relations.

Definition 1: a hierarchy of concepts O can be understood as a partially ordered finite set C of concept types. The order on C is denoted \leq .

Definition 2: A priority relation is defined as a binary relation $p(x, y), (x, y) \in C$. The set of user defined priority relation is called P

Definition 3: The “order” relation \triangleleft is defined as follows:

1. $\forall (x, y) \in C, p(x, y), \forall z \in C \ z \leq x, y \triangleleft z$
2. $\forall (x, y) \in C, p(x, y), \forall z \in C \ y \leq z \rightarrow \neg(\exists w | p(z, w), y \leq w), z \triangleleft x$
3. $\forall (x, y) \in C, y \leq x, \neg p(y, x), \neg(\exists z | x \leq z, p(y, z)), y \triangleleft x$

Theorem : relation \triangleleft defines a total order on T_c if and only if

$$\forall (x, y) \in C, x \perp y, p(x, y) \text{ or } p(y, x) \text{ or } \exists z | p(x, z) \ y \leq z \text{ or } \exists z | p(y, z) \ x \leq z$$

Proof: First implication

\triangleleft is a total relation order then $\forall (x, y) \in C, x \triangleleft y \text{ or } y \triangleleft x$. If $x \triangleleft y$ then 1. and 2. tells us necessarily that either $p(y, x)$ or $\exists z | p(y, z), y \leq z$. x and y are independent, a simple permutation gives us the first implication

Second implication: if $x \perp y, p(x, y)$ gives us $\forall z \in C \ z \leq x, y \triangleleft z$ with (1.) and for $z = x$ we get $y \triangleleft x$. If $\exists z | p(x, z) \ y \leq z$ with (2.) we can deduce that $y \triangleleft x$. By permuting x and y we demonstrate that in any case when $x \perp y$ we have a relation order between x and y .

When $y \leq x$, (3.) automatically gives us an order between x and y . Finally $\forall (x, y) \in C, x \triangleleft y \text{ or } y \triangleleft x$ the relation \triangleleft is total on C .

The transitivity of \triangleleft directly comes from (1.) and (2.) if $x \perp y$ and from (3.) if not.

The antisymetry is also obtained provided the additional hypotheses that p has this property.

If the relation order had to be defined manually for each pair of pedagogical types, the expert (teacher) would have to specify all the possible relations between types. This order can be illustrated by the following figure 70. In this simple example, only three relations are necessary to specify an order (left) that respect the definition given above. Here we introduce a forth one, in accordance with the definition to put node G in front. Using the subsumption links we can deduce the following relation order $G > A > B > D > E > C > F$. In the right, 11 relations must be generated to explicit the $<$ relation order.

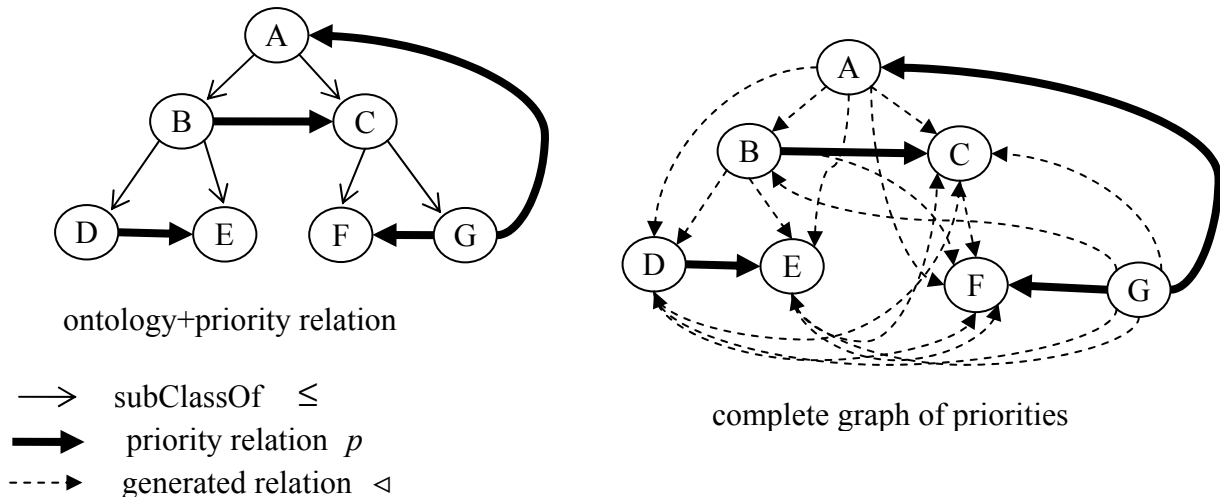


Figure 70 – Propagation of the priority relation on the ontology

In our example the ontology must define $C_{28}^2 = 378$ relations. This is equivalent to redefine a complete graph, connecting all the concepts together. It represents too much work and most of the relations can be determined automatically by exploiting the semantic structure of the ontology (in particular subclass relations).

From a small set of statements, and using forward chaining rules, an inference engine can complement this knowledge about the precedence of a type over another one. For example, all the subclasses of “Fundamental” should have priority over those of “Auxiliary”, provided that it does not conflict with other user-defined relations.

Generation rules presented in table 9 are sufficient to complete an ontology given a minimal set of user-defined priority relations. They use the syntax proposed by Corese semantic search engine and match conceptual graph rules (Sowa, 84). The premise of the rule contains a SPARQL query. For each set of answers, the triples indicated in the conclusion of the rule are added to the knowledge base (Corby et al., 04).

The following three rules formalized in table 9 are necessary to complete the graph:

1. The subsumption links produce priority relations from concepts that have not already priority relations pointing to them and that are not prior to their own ancestor (In the rule the first triple ensures that concept ?x is in the pedagogical ontology, otherwise this would apply to any class)
2. The priority of a concept over another one must be propagated to the descendants of the first concept provided they are not already subsumed by this first concept.
3. When a concept x is subsumed by a less prior concept w, concepts in the hierarchy that are between w and x have priority over the concepts x is prior to.

Table 9 - Rules used to infer priority relations between pedagogical concepts.

$?x \text{ rdfs:subClassOf } \text{edu:InstructionalObject}$ $?x \text{ rdfs:subClassOf } ?y$ $\text{option}(?z \text{ edu:priorTo } ?x)$ $\text{FILTER } ! \text{ isBound}(?z)$ $\text{OPTION}(?x \text{ edu:priorTo } ?w$ $\quad ?x \text{ rdfs:subClassOf } ?w)$ $\text{FILTER } ! \text{ isBound}(?w)$ \Rightarrow $?y \text{ edu:priorTo } ?x$	$?x \text{ rdfs:subClassOf } ?Sx$ $?Sx \text{ edu:priorTo } ?y$ $?x != ?y$ $?y \text{ not::rdfs:subClassOf } ?Sx$ \Rightarrow $?x \text{ edu:priorTo } ?y$	$?x \text{ rdfs:subClassOf } ?w$ $?x \text{ edu:priorTo } ?w$ $?x \text{ rdfs:subClassOf } ?z$ $?z \text{ rdfs:subClassOf } ?w$ $?x \text{ edu:priorTo } ?y$ \Rightarrow $?z \text{ edu:priorTo } ?y$
---	---	---

If the expert did not define explicit cycles, the priorities define a partial relation order. To ensure it in the mathematical sense, reflexivity must be added by another rule:

$?z \text{ rdfs:subClassOf } \text{edu:InstructionalObject} \Rightarrow ?z \text{ edu:priorTo } ?z$.

The above rule patterns are generic. They could be reused *to complete any relation order complying with the three rules defined above*. This mechanism of ordering constitutes an original result of our proposal.

6.3.2.4. Exploiting an ordering relation in SPARQL

There is a limitation with the current SPARQL query language for using a relation order directly. The language lacks expressivity for specifying that a result should be presented automatically ordered according to a given semantic relation. Only a numerical index for ordering is included in the proposal for SPARQL.

To solve this problem, we propose to compute an index that expresses this partial order. This index can then be used to order the results.

The computation of the index might be time consuming. The algorithm, shown in figure 71, has been designed to compute such an index on a hierarchy of classes given a top concept and a semantic relation. The algorithm works on the directed acyclic graph created by the order relation. The nodes of the graph are indexed by a number so that if $A \text{ priorTo } B$ then $n(A) < n(B)$.

Let P be the set of arcs of type “edu:priorTo” expressing the priority between two types
 $P = \{p \mid p =: \text{edu.priorTo}\}$

Let T be the set of instructional object types $T = \{t \mid t \leq: \text{edu.InstructionalObject}\}$

Initialization:

$\forall t \in T \Rightarrow n(t) = 0$

While $P \neq \{\emptyset\}$

$D = \{d \in T \mid \forall t \in T; \neg(t \text{ edu.priorTo } d)\}$

For-each $x \in D$:

For-each $y \in T \mid x \text{ edu.priorTo } y$:

$n(y) = n(x) + 1$

$P = \{p \in P \mid \neg(x \text{ p } y)\}$

Figure 71 – Algorithm to compute a generic index on a partial relation order

Figure 72, below, shows how the index is used in SPARQL syntax to specify an ordering of the results. The sorting key is specified by the words “ORDER BY” placed at the end of the query.

```

1. SELECT ?doc WHERE{
2.   ?doc rdf:type ?y
3.   ?y edu:order ?order
4.   ?doc skos:primarySubject :Variable}
5. ORDER BY ?order

```

Figure 72 – A query ordered according to the generated index

6.3.2.5. Generalization

The above solution has a number of limitations that we experienced during the development of QBLS.

- The algorithm is not dynamic. If the model evolves, it must be completely re-run to update the index. It might not be a crucial problem as evolutions of the model should not be frequent in learning applications to ensure stability. In other domains, it might be a problem.
- Given the complexity of the algorithm, it may reveal quite time consuming to run. In the pedagogical domain, we are dealing with rather small pedagogical ontologies, but in other application fields, scalability issues of such techniques might be raised.
- Finally, it forces the application to run the algorithm independently from the search engine, whereas all the other operations related to semantics are carried out by the search engine. From a technical point of view, there is a dispersion of the code, which is not advisable in the perspective of maintenance and deployment of new applications.

A proposition we make then is to extend the SPARQL syntax shown in figure 72, to include the expression of the ordering of the results according to a relation and not just a literal, a number or a date. For instance the syntax could be “ORDER WITH edu:order”, “edu:order” being the relation used to compare two nodes of the graph.

In an even more general perspective this could be done according to some comparator function or “ORDER WITH compare()”. The compare() function would be a two-arguments, user-defined function returning Boolean values. “compare(a,b)” returns 1 if a “is less” than b otherwise it returns 0. This would offer a programmatic way to define relations orders.

Through this example of applying rules to infer pedagogical relations, we show that *generic mechanisms can be used to enforce pedagogical rules in the learning system*.

We also propose a *generic enhancement for the SPARQL language targeting the ordering of the results according to any semantic relation or user defined function*.

6.3.2.6. Personalization with a stereotype model

Based on this priority mechanism, we offer the possibility to specify priority relations for a given profile. Instead of defining one set of priority relations, the expert teacher can define as many profiles as he/she wants and assign different priority relations within each of those profiles. Table 10, below, gives an example of two different profiles used in QBLS. They illustrate two classically distinguished cognitive profiles of “thinker” and “feeler”. They are also called “stereotype” models (see 3.1.4.2):

- The first profile gives priority to fundamental resources over auxiliary ones. In deeper levels of the ontology it gives priority to abstract resources like “*Law*” over more practical ones, like “*Process*”. This expresses a “thinker” profile.
- The second profile prioritizes auxiliary resources over fundamental ones, and “*Fact*” over “*Definition*”. It should match a “feeler” cognitive profile more interested in practical learning experiences than in theoretical exposure.

Table 10 – Comparison between to different profiles in QBLS

Profile 1 (thinker)			Profile 2 (feeler)		
Fundamental	edu:priorTo1	Auxiliary	Auxiliary	edu:priorTo2	Fundamental
Introduction	edu:priorTo1	Remark	Introduction	edu:priorTo2	Remark
Introduction	edu:priorTo1	InstructionalObject	Introduction	edu:priorTo2	InstructionalObject
Remark	edu:priorTo1	Conclusion	Conclusion	edu:priorTo2	Remark
Definition	edu:priorTo1	Fact	Fact	edu:priorTo2	Definition
Fact	edu:priorTo1	Law	Fact	edu:priorTo2	Law
Law	edu:priorTo1	Process	Process	edu:priorTo2	Law
Evidence	edu:priorTo1	Illustration	Interactivity	edu:priorTo2	Evidence
Evidence	edu:priorTo1	Interactivity	Interactivity	edu:priorTo2	Illustration

For querying those cognitive profiles, SPARQL is exploited again. It is possible in SPARQL to have a variable at any position of an RDF triple. Instead of asking for the `edu:order` relationship between the type `?t` of the resource and the ordering index (figure 72), the predicate of this relation is identified by the `?p` variable which is assigned to the type contained in the user profile. Thus, it is possible to personalize the query depending on user profiles. This technique is generic and it could be applied in many different contexts. Figure 73 shows the excerpt of the SPARQL query performing this personalization.

```
1. OPTIONAL{FILTER(?user = <%=request.getParameter("user")%>)
2. ?user edu:profile ?profile
3. ?profile edu:orderingType ?p
4. ?t ?p ?order}
```

Figure 73 – Personalizing the query according to a user profile

6.3.3. Retrieving literal information

Semantic queries enable to retrieve and order learning resources relying on both domain and pedagogical knowledge. Additional information might be required to build an interface (e.g. the title of the resources, the label of the concepts, etc. are necessary information to construct a rich interface). Through the unification brought by RDF, this “low-level” knowledge can be accessed in the same way. For example, figure 74 completes the previous query by augmenting it with triples about the label of the concepts (line 6) and the title of the resources (line 3).

```
1. SELECT ?doc ?title ?label WHERE{
2. ?doc rdf:type ?y
3. ?doc dc:title ?title
4. ?y edu:order ?order
5. FILTER(?concept = :Variable)
6. {?doc skos:primarySubject ?concept . ?concept skos:prefLabel ?label}
7. UNION
8. {?doc skos:primarySubject ?s . ?s skos:broader ?concept . ?s skos:prefLabel ?label}}
9. ORDER BY ?order
```

Figure 74 – SPARQL query completed with information about labels and titles

To sum up, the query described in this section allows us to build an interface like shown in figure 60. This interface is generated in response to a request on the concept of “*Object*” in Java. Four resources are returned, they are ordered from left to right according to their pedagogical type indicated in the tab header (“*Definition*”, “*Explanation*” and “*Example*”). The label “*Object*” and the title of each resource (“*Object and Classes*”, “*Other Observations*”, etc.) are retrieved from the query.

6.3.4. Context awareness

The queries presented above are parameterized by a given concept, a given user, etc. We can also introduce in the query the idea of context. In QBLS, context information is limited to the chapter currently studied. The proposed pattern is generic and may involve other contextual information.

We complete the query shown in figure 74, to retrieve the list of pertinent concepts for each resource belonging to the given chapter. The necessary additions to the query are shown figure 75:

- The documents retrieved (?doc) must belong to the chapter (line 3).
- The potential links from this document to concepts are retrieved through the dc:subject relation (this considers any primary and non-primary relation) (line 4).
- For each of those concepts, the associated external documents are determined (line 5, 6). This includes resources retrieved using domain vocabulary structure.
- External documents must belong to the chapter (line 7).

```

1. SELECT ?doc [...] ?external_concept WHERE{
2. [...]
3. ?doc edu:belongsTo java:Chapter1
4. ?doc dc:subject ?external_concept
5. {external_doc skos:primarySubject ?external_concept} UNION
6. {?external_doc skos:primarySubject ?c3 ?c3 skos:broader ?external_concept}
7. ?external_doc edu:belongsTo java:Chapter1}

```

Figure 75 – Additional triples queried to take into account contextual information

6.3.5. Domain/range querying

The construction of a query for building an interaction with the learner has been presented above. SPARQL queries might be used in many other contexts. We present below how it can be used to select appropriate relations when editing RDF annotations.

For example when someone wants to add a solution to a question in the knowledge base, it is interesting to propose him/her all the possible relations that can be grafted on a question instance. The domain and range defined in the model are used to restrict the possible choices, given the following assumptions:

- We assume that properties inherit the domain and range definition from their ancestor relations. This is implicit in RDFS: if P has for domain C, subjects of triples whose predicate is P are instances of the class C. If P' is a sub-property of property P, then all pairs of resources which are related by P' are also related by P. So subjects of triples whose predicate is P' are also related by P, and thus instances of the class C.
- If a relation P has a range defined on C' and C' is subclass of C, then relation P should be proposed when C is in the subject of the relation. Likewise, relations with domain defined on C, with C ancestor class of C' should be proposed if C' is the predicate of the relation.
- Finally, properties without range or domain are considered to fit any concept according to the RDFS model. That means they will always be proposed to the user. This hypothesis might lead to the proposal of too many relations. In such situations, heuristics might be deployed for example to reduce the query to a single namespace for example,.

The above constraints are illustrated by the following SPARQL query (figure 76) which retrieves all the “coherent” relations given a specific predicate and subject.

Using a similar type of mechanism, the potential objects can be determined for a given relation. This mechanism is used to construct the selectors shown figure 63 and figure 64.

```

1. SELECT ?p WHERE{
5. FILTER(domain_type = xxxxxxxx)
6. FILTER(?range_type = xxxxxxxx)
7.
8. {?p rdfs:domain ?domain_type}
9. UNION
10. {?p rdfs:domain ?d ?domain_type rdfs:subClassOf ?d}
11. UNION
12. {?p rdfs:subPropertyOf ?mp ?mp rdfs:domain ?domain_type
13.   OPTIONAL{?p rdfs:domain ?d} FILTER(!bound(?d)) }
14. UNION
15. {?p rdfs:subPropertyOf ?mp ?mp rdfs:domain ?parent
16.   ?domain_type rdfs:subClassOf ?parent
17.   OPTIONAL{?p rdfs:domain ?d} FILTER(!bound(?d)) }
18. UNION
19. { OPTIONAL{?p rdfs:domain ?f } FILTER(!bound(?f))
20.   OPTIONAL{?p not::rdfs:subPropertyOf ?mp} } FILTER(!bound(?mp)) }
21. UNION
22. { OPTIONAL{?p rdfs:domain ?f } FILTER(!bound(?f)) ?p rdfs:subPropertyOf ?mp
23.   OPTIONAL{?mp rdfs:domain ?class} FILTER(!bound(?class)) }
24.
25. {?p rdfs:range ?range_type}
26. UNION
27. {?p rdfs:range ?r ?range_type rdfs:subClassOf ?r}
28. UNION
29. {?p rdfs:subPropertyOf ?mp ?mp rdfs:range ?range_type
30.   OPTIONAL{?p rdfs:range ?r} FILTER(!bound(?r)) }
31. UNION
32. {?p rdfs:subPropertyOf ?mp ?mp rdfs:range ?parent ?range_type rdfs:subClassOf
   ?parent
33.   OPTIONAL{?p rdfs:range ?r} FILTER(!bound(?r))}
34. UNION
35. { OPTIONAL{?p rdfs:range ?r} FILTER(!bound(?r))
36.   OPTIONAL{?p rdfs:subPropertyOf ?mp} FILTER(!bound(?mp))}
37. UNION
38. { OPTIONAL{?p rdfs:range ?r} FILTER(!bound(?r)) ?p rdfs:subPropertyOf ?mp
39.   OPTIONAL{?mp not::rdfs:range ?mr} FILTER(!bound(?mr)) }
40. }

```

Figure 76 - Query determining the potential relations given a domain and a range concept

6.4. Semantic query processing

We detail in this section how the queries presented above are effectively carried out in QBLS using the Corese semantic search engine. This part focuses on the use of this engine, however most considerations are totally generic and would be valid for any search engine based on semantic web standards.

6.4.1. Corese specificities

Internally Corese search engine uses the conceptual graph formalism (see 3.2.2.2) to answer SPARQL queries. Queries are interpreted as graph patterns and the projection algorithm finds the matching results in the graph formed by the annotations. Graphs in CG format are translated into RDF according to the transformation proposed by (Corby et al., 00). An output to the SPARQL result format in XML is produced.

The projection algorithm in Corese has been enhanced with heuristics that speed up the search process. Developments conducted in the context of QBLS allowed us to introduce several enhancements. For example, it first matches the triples that possess the fewer candidates in the base. Sorting the triples of the query according to this frequency potentially improves the search process.

This search mechanism relies on the “closed-world” hypothesis. It means that all the knowledge potentially related to the query is known at the time the engine is started. In the context of e-learning systems like QBLS, we think that this assumption is perfectly justified because the knowledge base is the result of a semantization task and that additional resources cannot be “added” externally.

Still knowledge may evolve during the use of QBLS:

- When the knowledge base is changed using the editor, the engine is restarted, which implies delays.
- User navigation traces are dynamically added to the knowledge base, but they are not taken into account by the rule mechanism.

6.4.2. Dynamic aspects

The most difficult issue is currently the introduction of dynamic knowledge. In Corese, new knowledge can be added at run time but in that case, rules for example, do not apply dynamically to the new knowledge. Conceptual graph rules as implemented in Corese are production rules. They create new graphs that are added to the knowledge base. However, this production mechanism only happens once: if additional knowledge is introduced at run time, the rule engine must be restarted and all the inferences must be performed again, which might imply delays in the engine response if a complex set of rules is used.

Compared with the fully proprietary AHA system that has a dedicated dynamic rule mechanism, this issue is currently the main limitation of the implementation based on Corese generic engine.

6.4.3. Rule standardization

The Semantic Web standardization effort has not covered the domain of rule languages yet. Proposals for a “Semantic Web Rule Language” (SWRL, 04) have been made and a working group is currently developing a rule interchange format (RIF, 06) to answer this need. This common language for representing rules would complete the panel of standards, and the above architecture could be completely qualified as generic and standard based.

In addition, connections with other systems, like AHA, would be facilitated.

6.5. Interface generation

Thus, interaction with learners occurs through human readable interfaces presented in a web browser. In this section we present how the QBLS interface can be constructed from the result of SPARQL queries. RDF and other semantic web formalisms can be understood by both human experts and machines, but an interface is necessary in order to present this knowledge to learners.

6.5.1. Integration of XML technologies

Through the underlying XML layer on which the semantic web builds (see the “layer cake” figure 20), the generation of interfaces is greatly facilitated. XML technologies allow us to manipulate information and transform it with XSL templates in a declarative way. This includes the results given by a semantic search engine producing answers in the SPARQL XML binding (see figure 67). XSL stylesheets can be created to display the query results

depending on the targeted application. The connection with XML is part of the SPARQL proposal. The schema for expressing its results in XML is part of the future standard.

6.5.2. Dynamic pages

A complex interface, cannot be based on the transformation of a single XML result. To offer richer interfaces several components need to be “assembled”. The JSP language is used in QBLS to “glue” the major dynamic components of the interface. For example, each component of figure 60 is the result of a semantic query. The different queries are specified in a JSP file. The final aggregated result is an XHTML page visible in most browsers.

The choice of this technology is motivated by its ease of deployment, its flexibility and its wide diffusion.

Without detailing thoroughly this mechanism, which would be out of the scope here, we use the fact that JSP allows programmers to define their own tags. Such tags can be included in JSP pages along with classic XHTML and other JSP-specific tags. Tags are associated with a specific class programmed in Java. At run time when the page is requested through HTTP, the JSP container (here Tomcat) constructs a response page and replaces the user-defined tags by the result of the execution of a specific method of the associated class.

By introducing a tag, that queries the Corese engine and by processing the results through a given XSL stylesheet, at run time query results can be included directly into the page. The inner body of the tag contains the query in SPARQL syntax and a parameter specifies the stylesheet to use. Doing this, several tags can be used with different queries on the same page.

6.5.3. Detailed example of interface construction

We detail here the generation of “dynamic” parts of the interface based on semantic queries. For example, the query detailed in figure 74 is used to construct a navigation interface for resources. Here we illustrate such interface construction with a simpler query retrieving the history of visited concepts (frame on the right figure 60).

Each visit of a concept by a given student is recorded and stored in the annotation base. This information is then queried to construct a dynamic interface. Figure 77 below, shows an excerpt of a JSP file containing the custom tag “cos:query” used for the generation of the history list. It contains a parameter “xslt” indicating which stylesheet located on the server must be used to process the result.

The SPARQL query is placed inside the tag. The query itself is dynamic: the expression between the symbol “<%=” and “%>” (line 3) is evaluated by the server at run time when it processes the JSP page (this is a feature of JSP). The expression is replaced by the user ID currently logged on the system (line 3). The user ID is accessible as a request parameter and thus will replace the call to the accessor method placed inside the query. The other triples ask for the log records of the specified user (line 4), the concept visited (line 5) with its label (line 6) and the date of the corresponding record (line 7). Results are sorted according to this date and the display is limited to ten so that only the last ten results are returned.

```

1. <html>
2. ...
3. <cos:query xslt="/cours/navigation_history.xsl">
4.   SELECT display xml ?concept ?label WHERE {
5.     FILTER(?user = <%=request.getParameter("user")%> ).
6.     ?user edu:log ?visit .
7.     ?visit edu:visitedConcept ?concept .
8.     ?concept skos:prefLabel ?label .
9.     ?visit dc:date ?date}
10.   ORDER BY desc(?date)
11.   LIMIT 10
12. </cos:query>
13. ...
14. </html>

```

Figure 77 – JSP tag performing a query to the Corese search engine

This query is processed by the semantic search engine Corese that returns an XML response. The beginning of the response is shown below in figure 78, in the specification for SPARQL-XML response binding. The variable names defined in the query (line 3, 4) are used to identify the different values grouped into result sets.

```

1. <sparql>
2. <head>
3.   <var name="concept"> </var>
4.   <var name="label"> </var>
5. </head>
6. <results distinct="false" sorted="true">
7.   <result>
8.     <binding name="concept">
9.       <uri>http://www.inria.fr/acacia/java-skos#Object</uri>
10.    </binding>
11.    <binding name="label">
12.      <literal xml:lang="en">Object</literal>
13.    </binding>
14.  </result>
15. <result>
16. ...
17. </result>
18. </results>
19. </sparql>

```

Figure 78 – XML-binding for SPARQL results

This response is processed by the following stylesheet (figure 78). The XPATH expressions, in *italics*, match the standard SPARQL-XML above. They look for the ‘concept’ (line 9) and ‘label’ (line 10) attributes to build an HTML table.


```

1. <?xml version="1.0" encoding="utf-8"?>
2. <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
3.   xmlns:s="http://www.w3.org/2005/sparql-results#" exclude-result-prefixes="#all">
4.   <xsl:output method="html" omit-xml-declaration="yes" indent="yes"/>
5.   <xsl:template match="/s:sparql">
6.     <xsl:for-each select="s:results/s:result">
7.       <tr>
8.         <td>
9.           <a href="/prog101/cours/cours.jsp?concept={escape-uri(
10.            s:binding[@name='concept']/s:uri, true())}">
11.             <xsl:value-of select="s:binding[@name='label']/s:literal"/>
12.           </a>
13.         </td>
14.       </tr>
15.     </xsl:for-each>
16.   </xsl:template>
17. </xsl:stylesheet>

```

Figure 79 – XSL stylesheet processing SPARQL result

For each result of the query, an entry in the HTML table will be generated like the one below (figure 80), and will appear on the screen as a hyperlink towards the previously visited concept.

```

1. <tr>
2.   <td>
3.     <a href="/prog101/cours/cours.jsp?concept=
4.       http%3A%2F%2Fwww.inria.fr%2Ffacacia%2Fjava-skos%23Object">Object</a>
5.   </td>
6. </tr>

```

Figure 80 – Generated HTML output

The query is quite simple and does not involve complex reasoning. The originality of this approach is that all the logical operations are conducted on the server side. XSL stylesheets are purely static templates that directly transform results in XML into readable HTML.

This demonstrates, is the *integration of semantic web technologies within the actual web*. The proposed technique is *absolutely generic* and may be applied for a variety of context where dynamic interfaces need to be built.

6.5.4. Editor interface

The editor interface is also build with an XSL stylesheet but on a different principle. The edited file is processed through the stylesheet to generate an editable representation of the file. The stylesheet translate the content of the file to HTML and add dynamic widgets when appropriate. For generating the widgets (e.g. the selector), the page accesses the Corese search engine. It internally formulates SPARQL queries and retrieves XML results that can be stored in variables and exploited via XPATH like any XML input:

- Each RDF node is queried to get the label of its type in the desired language.
- URIs in plain text are meaningless for teachers who work on a closed set of resources. They are also hard to read. An innovative solution developed here is to hide them from the user and rather use some kind of “template” (defined in XSLT) to customize the display for a specific type. For example, resources are identified by their title. It improves the readability of the RDF annotation, and offers the possibility to handle multilingual annotations.

6.5.5. Evaluation of the technical solution proposed

6.5.5.1. Observations

Our implementation of a learning system based on semantic web technologies possesses a number of arguments in favor of its adoption and generalization as a *software design pattern*.

➤ It offers a uniform way to access knowledge. In the case of learning applications, pedagogical, domain and document information are required. RDF perfectly handles the expression of such heterogeneous knowledge, and SPARQL offers a unique access language for all sorts of information.

➤ Using such a pattern supposes to rely on an existing search engine. Following the standards, many engines should soon fit in that scope. The choice of engine would depend on the type of inferences required or on the performance indicators of each implementation. However, no extra code should be developed to perform inferences or manipulate knowledge. The amount of code to develop is then greatly reduced while allowing classical web techniques (HTML, JavaScript, etc.) to be used as well.

6.5.5.2. Code analysis

To consolidate the above observations, the code running on the server to manage the course consultation service has been analyzed. It represents less than a thousand lines (in the experiment on Java programming). These lines are distributed as shown table 11. All lines in the files are taken into account except blank lines and comments. The formatting respects usual standards (for instance XSL is fully indented). This encompasses two different interfaces (the 2 JSP): the one shown figure 60, and a dynamically constructed table of content. The lines of Java code concern the manager application and the definition of the custom tag for including queries in the pages. They contain a lot of code because of the priority algorithm described previously and because logging operations are performed at this level too (users' navigation traces are added to the knowledge base dynamically).

Table 11 – Lines of code for the QBLS-2 learner interface

File type	Number of files	Number of lines
JSP	2	233
XSLT	5	243
CSS	1	30
JAVA	2	325
TOTAL	8	831

Other components are necessary to produce the “teacher” interfaces (for example the editor). However, for the “core” code we can conclude that its reduced size greatly facilitates maintenance and evolution.

Compared to other learning systems such as AHA (de Bra et al., 03), composed of many Java classes, or Whurle (Brailsford et al., 02) for which the central stylesheet already contains more than 1600 lines of code, the advantage of our approach is obvious.

This philosophy of relying on a *generic infrastructure for semantic web applications* is further pursued by the development of the generic platform SeWeSe (Semantic Web Server) by the Acacia team.

6.6. Conclusions

This chapter demonstrates the integrative power of semantic web technologies by covering a large panel of functionalities interesting in an advanced on-line course consultation learning

system. Examples cover classical features of course consultation systems: resource selection, path planning based on pedagogical knowledge as well as more advanced features like dynamic content and adaptation. All these mechanisms are inspired from various existing tools, and not bound in any way to the specificities of QBLS.

The implementation based on semantic queries leading to the construction of end-user interfaces should be viewed as a generic guide and pattern to implement such functionalities.

The underlying architecture is also proposed as a generic design pattern. The “semantic middleware” approach answers a large variety of problems and there is little doubt left about its power and interest for e-learning applications. It may also be applied in other domains involving semantic querying like semantic intranets, semantic collaborative spaces, etc.

The generic aspect of the proposal in term of technological design is emphasized by the use of standards compared to proprietary formats (databases, files, etc.). The “semantic” layer offers a supplementary abstraction level that also increases the generality of this pattern.

The QBLS system can be reused as it is. Only modifications of the knowledge base would be necessary to apply it on a different course (domain model, resources). Its implementation is generic in the sense that ontologies and annotations can be changed without having to modify it.

From a performance point of view, the implementation revealed sufficient performance for applying this tool to a real learning situation. This question becomes crucial when considering not dynamic interfaces but dynamic knowledge. In the next section, we present a specific aspect of dynamicity including dynamic knowledge linked to user characteristics and actions. Performance issues will be mentioned then.

With such a framework, design limitations appear from the poor knowledge we have of the impact of such systems. Many features can be thought of and implemented, but we possess little knowledge about their impact in actual use. This is a much more generic problem of “intelligent” learning system. We do not intend to address it here, but we present some directions we explored in the following chapters 7 and 8.

It must be kept in mind that with such frameworks, *reliable options are now available for dynamic learning systems centered on the exploitation of resources through semantic web technologies.*

7. TOWARDS A SEMANTIC ANALYSIS OF LEARNING PATHS..... 151

<i>7.1. Collecting activity traces</i>	152
7.1.1. Knowledge collected on learner activity	152
7.1.2. Representation of learner activity	153
<i>7.2. System adaptation based on activity traces for learners</i>	156
7.2.1. Exploitation of learner models	156
7.2.2. Conceptual back navigation	157
7.2.3. Evaluation of adaptation mechanisms	160
<i>7.3. Teacher's interpretation of activity traces</i>	161
7.3.1. Teacher's interrogations	161
7.3.2. Visualization of the course in the log analyzer	161
7.3.3. Structural views	164
7.3.4. Representation and interpretation of single user navigation	166
7.3.5. Aggregated views for analyzing behaviors in groups	169
7.3.6. Evaluation of the log analyzer tool.....	172
<i>7.4. More automated analysis</i>	172
7.4.1. Statistical measures	172
7.4.2. Graph patterns	175
<i>7.5. Conclusion</i>	176
7.5.1. Outcomes of our proposed graph navigation model	176
7.5.2. Uses of interpreted paths	177
7.5.3. A broader perspective through the exchange of log data	177

7. TOWARDS A SEMANTIC ANALYSIS OF LEARNING PATHS

The importance of tracking or monitoring learner behavior is a key issue for education in general and for e-learning in particular. This is especially true for distance-learning situations, but even in the classrooms when interacting with computers, monitoring learners reveals difficult because the interaction with the teacher is modified compared to “classical” teaching.

Computer interaction should allow us to track learner activity more precisely in the spirit of personal tutors (Brusilovsky and Vassileva, 03). User activity on a computer leaves traces, or logs that must be gathered and interpreted. In this chapter we address this problem by looking how knowledge about the course, obtained previously through a semantization of this course, can help us provide new results for learner monitoring.

We keep here a practical approach. Complex tracking systems using video, or spying programs, are out of scope. Their use in a day to day situation is quite unrealistic for still a few years given the actual practice of computer use in schools and universities. We present, in the context of QBLS, how log information can be collected, formalized, represented and potentially interpreted using semantic web technology.

First, we justify the type of logs we are dealing with and propose a representation paradigm based on the navigation model presented in chapter 4. In a second section we present the interests of this representation of logs for adapting the system to each user's behavior. Then, we present a tool based on a visual paradigm to help teachers interpreting learner traces on a “semantized” course. A fourth section is dedicated to the perspectives of such user activity analysis.

7.1. Collecting activity traces

The data collected on user activity in “logging” system often express very low level information. Automatic interpretation of these data, based on logical models, rarely succeeds in bringing interesting personalized results. Before interpreting such information, intermediate representations must be used. For example, activity traces for an on-line course can be represented as navigation paths through the material. In this section, we present the traces we collect and the necessary steps that must be performed to start interpreting such traces as navigation paths.

7.1.1. Knowledge collected on learner activity

7.1.1.1. Recording data through HTTP

In order to support multiple operating systems, browsers, etc. and not rely on “spy” applications installed on the client machine, the tracking data must be collected on the server side. It ensures a good homogeneity of the data but it also means that only the HTTP requests sent by the client browser can be monitored (see the interaction cycle in 6.1.2). This is a very classical type of user monitoring in web-based learning system (ex. AHA). We do not consider potential network issues such as intermediate cache and network filtering. We make the hypothesis that all users’ requests are received on the server.

For each request received, the data collected on the server identify:

- Which domain concept was requested. The domain ontology defines those concepts.
- Which resource will be displayed. For each user’s request a resource is presented in the interface (see figure 60).
- Who performed the request. Learners are identified through a unique Id.
- When the request was received. The time reference is place on the server.

Log files on the server are used to store this information. Log files usually take the form of text-based suites of lines; in our semantic web context, we express logs by RDF statements.

This information is very incomplete with regard to the complexity of a learning activity. It is already a simplification from what could be recorded. For example, it does not differentiate between requests made by clicking a link in the history list, and a hyperlink in the content of a resource (see interfaces figure 59 and figure 60). Learners may also have access to other sources of knowledge (web, paper documents, peers, etc.) to fetch information from, while answering the questions.

As a generic rule, observable knowledge will always be incomplete. A first step to qualify the representation of activity traces is then to define the extent of the simplification that will be made compared to the real activity. This is important to define the types of interpretations that can be conducted on the data. For representing and interpreting navigation paths, such data seem appropriate, as we will see later on.

7.1.1.2. Categorizing user activity data

In complex tracking systems (ex: mouse tracking), the amount of data to process increases in large proportions compared to our server logs. The other way round, more high level observations might be made, like analyzing student’s productions (writings, answers, etc.) or ultimately asking feedback from students, in which case data sets possess a very high level of expressivity and rather small sizes, but require an important “human” input.

We can summarize the tracking/feedback problem in e-learning on the following continuum, in figure 81, depending on the level of information contained in the collected data.

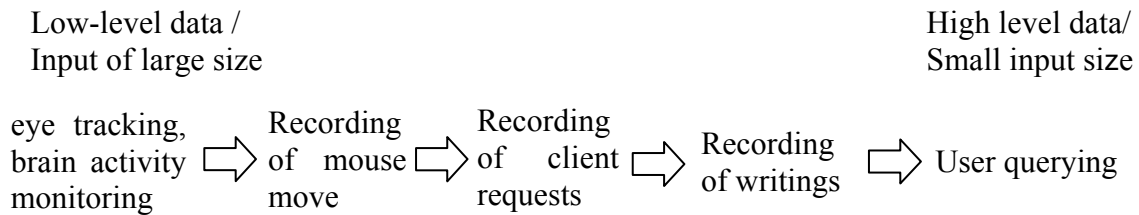


Figure 81 – Different information levels for learner tracking.

7.1.1.3. QBLS choices

The client requests, monitored in QBLS, constitute a “middle-range” type of information. Other criteria than the pertinence of the records for interpretation must be considered in this choice:

- Non-intrusive monitoring is an important argument: recording HTTP request can be done in non intrusive way and can be easily obtained given the system interaction paradigm based on client/server requests triggered by the user’s clicks.
- Interface specificities cannot be ignored. A different interface would allow to better track learner’s focus, but it would certainly imply too many clicks on the interface, and it would reduce its usability.

The choice of a particular log data also relies on a number of hypotheses that will be made when interpreting. In our case, for meaningful learning paths, we rely on the assumption that learners did not click at random on the interface, but that each action recorded is the result of a conscious process. In addition, we can only assume that learners actually read the content when visiting a resource.

In the end, there is a design *trade-off* between the type of log data recorded and the nature of the conclusions learners or teachers are looking for: Simplified data can be analyzed with generic techniques and provide generic conclusions. Whereas specific information, for example linked to interface specificities, chosen scenario, etc. offers a finer understanding of what learner did, but less generic conclusions.

In QBLS, we record very generic information like the time of request, the resource currently visible, etc. Such information could be recorded for any on-line course, but we also record specific information like the concept currently studied that is linked to the specificity of offering a conceptual navigation. These specificities are linked to the objective of representing user activity as navigation paths on a conceptual structure.

7.1.2. Representation of learner activity

7.1.2.1. A graph navigation model

The navigation model proposed in chapter 4 can represent several different conceptual navigations (see 4.4.2). It offers an *abstract* view of navigation that may suit different interfaces used for navigation, but it integrates the idea of conceptual navigation proposed in QBLS. We decided to use this model to represent user’s path graphically. Figure 82 shows an example of user navigation represented on the graph structure of a conceptual course. The nodes and arrows in thick line indicate the user’s path. The chronology of the steps is

indicated by the numbers from 1 to 11 (this example of a “semantized” course was first introduced in 5.4.4.3).

To create such representation of a user’s path, an interpretation of his/her recorded activity using the QBLS interface must be performed. It is important to keep in mind that the proposed visualization is generic while the interpretations are specific to a given interface and learning situation.

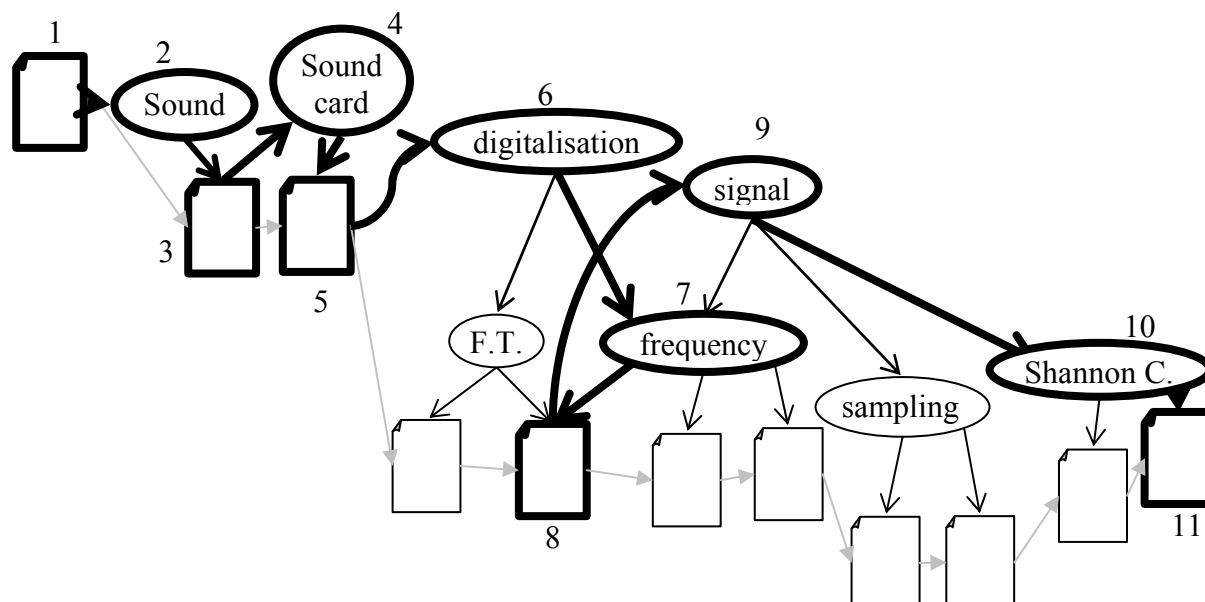


Figure 82 – Example of a navigation path visualized on a graph of the course

7.1.2.2. Transformation from sequential to connected path

Log statements in RDF are directly added to the knowledge base loaded in the Corese search engine. The RDFS model for expressing a log entry or a “visit” is presented in figure 83. Each visit is associated with a user, a time stamp, a concept of the domain and a resource from the course.

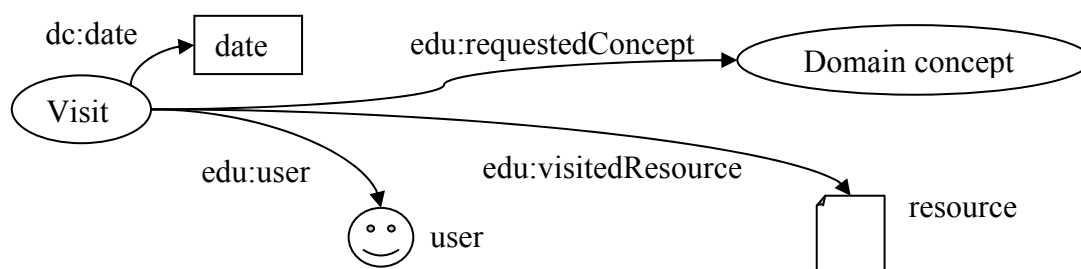


Figure 83 – The RDFS log model

Log statements are instantly available for querying. Before actually interpreting this data, it must be transformed to instantiate the navigation model illustrated in figure 82. We propose an automatic transformation from sequential visits to connected “steps”. The idea is to move from a disconnected list of visits to a connected graph representing the steps between the visited concepts and resources. Instead of expressing the “stops” of the path, it takes the “dual” approach by expressing the “jumps” from one stop to the other. On the graph, stops are represented by the nodes, whereas jumps match the edges.

The schema, on figure 84, illustrates the transformation from a graph pattern perspective. A first interpretation is performed. In the interface, concepts and resources are displayed simultaneously to reduce the number of clicks. We interpret a click on a new concept as two

steps of navigation. The first step represents the jump from the first resource to the second concept and another step from the second concept to the second resource. The transformation then introduces the distinction between the visit of a concept and the subsequent visit of the first resource attached to it. This is an interpretation compared to what happens in the interface where both actions are performed simultaneously (it generates only one record) and learners do not have a choice for this first resource.

If the path has not started yet, the first step (shown in grey in figure 84) will have to be added to complete the path.

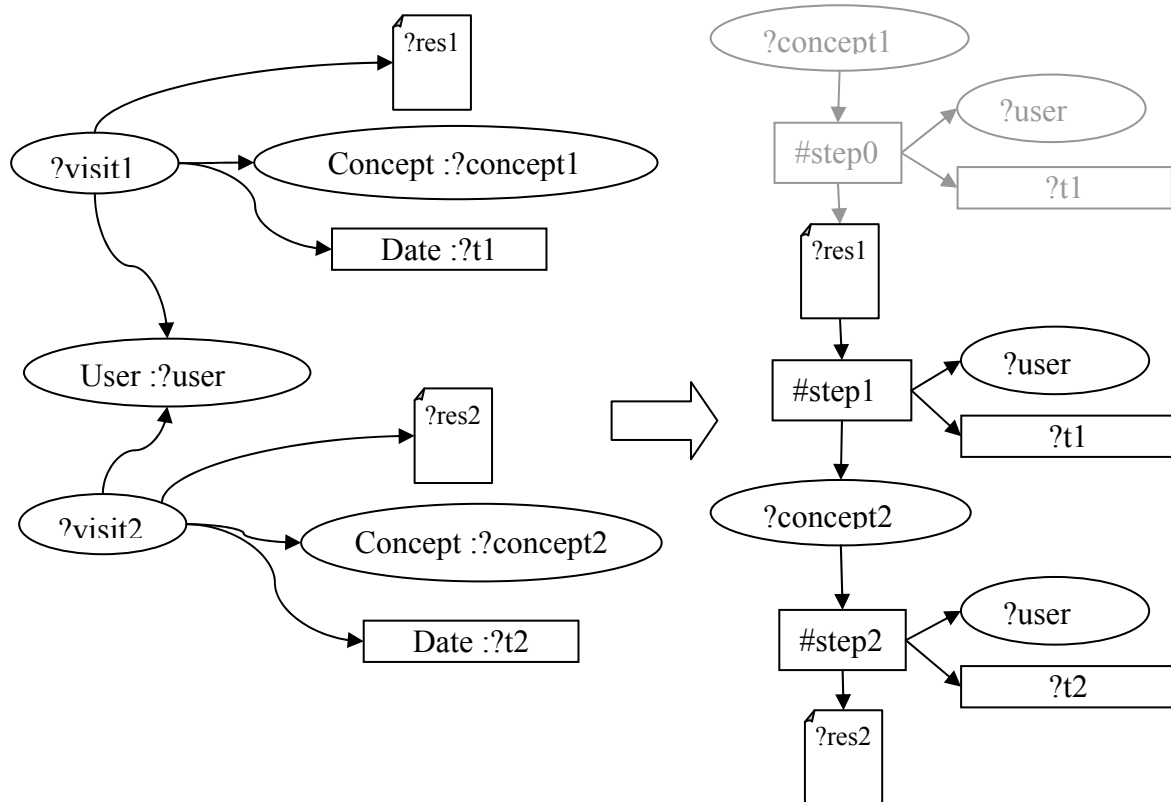


Figure 84 – Transformation pattern from two successive visits to a connected path.

In the underlying RDF, the transformation could be interpreted as a reification of the relations in the course graph (e.g. figure 82). Each step reifies an existing relation by associating it a user and a time stamp indicating when this relation has been “followed” in the navigation.

The transformation can be performed by a rule engine taking the above pattern as a production rule (like the rule engine of Corese). If working with RDF files, an XSL template can take the RDF/XML expression of the visits and generate the corresponding path in an RDF file.

7.1.2.3. Relevance of the proposed model for learner activity reporting

Given the above interpretation, the question might be raised whether this approach is realistic or not for tracking student’s activity. Path analysis on hypertexts often revealed quite poor results (McEneaney, 99), but our approach cannot be compared to classical hypertexts because navigation happens in a semantic space. We think it makes a difference, especially because the goal of a click can be much more precisely interpreted as a conscious action to access a specific concept (click on a concept link) or a specific type of resource (click on a resource “tab” see figure 85). This is a quite original perspective that certainly needs to be further refined. The presented navigation model is proposed as a potential reference for the design of future tracking systems.

7.1.2.4. *Qualifying the interpretation*

The three points of view, proposed in (Mazza and Dimitrova, 04), can be used to qualify the domains of interpretations that might be supported by visualizing student activity traces using such tool: social, cognitive and behavioral.

- The visualization of semantic course structures, and their interpretation by teachers, first focuses on the *behavioral* aspects. It is possible to recognize profiles that tend to visit answers first, from those who only use them afterwards. Students that perform a clean navigation can be separated from other profiles that need many iterations on the same material.
- The *cognitive* aspects rely on a much deeper and difficult interpretation of the navigation. The status of a non-visited concept for example, is difficult to evaluate. Several contradictory interpretations are possible: learners may have skipped it because they already knew about it. Alternatively, they were lost and did not think that the concept was even relevant for the given problem. So cognitive interpretations are possible, but they often need to rely on very strong assumptions.
- Social interaction is very difficult to monitor when natural channels of communications are used (typically in face-to-face interaction in a classroom). For distant learning situations, the system may record such communications mediated by tools (e.g. forum, instant messaging, etc...). Social interaction has a crucial impact on learner activity and knowledge construction and is a generic mechanism, present in any learning situation.

In the computer assisted learning situation of QBLS we only aim at collecting behavioral indicators automatically. The link to cognitive aspects is performed by a human interpretation. During the use of QBLS, we experienced social interaction under the form of discussion between groups of students on how to use the interface or how to answer the question. It may have influenced the user navigation that transpires in the logs. However, the cost of considering it is not reasonable.

In the following sections we present how the behavioral indicators are used. First, we present the low level system interpretations that enable automatic adaptations. Then we present teacher's interpretations that may lead to higher level cognitive information and can potentially trigger different actions.

7.2. System adaptation based on activity traces for learners

A major goal of interpreting activity traces in the domain of e-learning is to perform adaptation automatically. "*Adaptation*" or "*personalization*" describes the capacity of a system to react differently depending on the user or depending on his/her previous actions.

Several types of adaptation mechanisms have been identified on hypertexts. They range from shallow interface customization (colors, background, etc.) to deeper path planning algorithms. We presented in chapter 6 an adaptation based on a stereotype model to select the resources and order them in the interface according to a profile. In the following sections, we present the adaptation we implemented in QBLS based on the exploitation of the activity traces stored in the learner's "history" which play the role of a learner "overlay" model.

7.2.1. Exploitation of learner models

Log information in the learner model is expressed in RDF. It can be queried in SPARQL just like pedagogical and domain knowledge. For example, the query used to build the central part of the interface presented in chapter 6 can be extended to return additional information for customizing this interface.

Figure 85 presents the QBLS-2 interface where the colors of the tabs for each resource about the concept of “method” are adapted:

- On the left, a resource already visited appears in a grey tab.
- In the middle, the current visible resource has a blue tab header.
- On the right, a non-visited resource is represented by a yellow header.

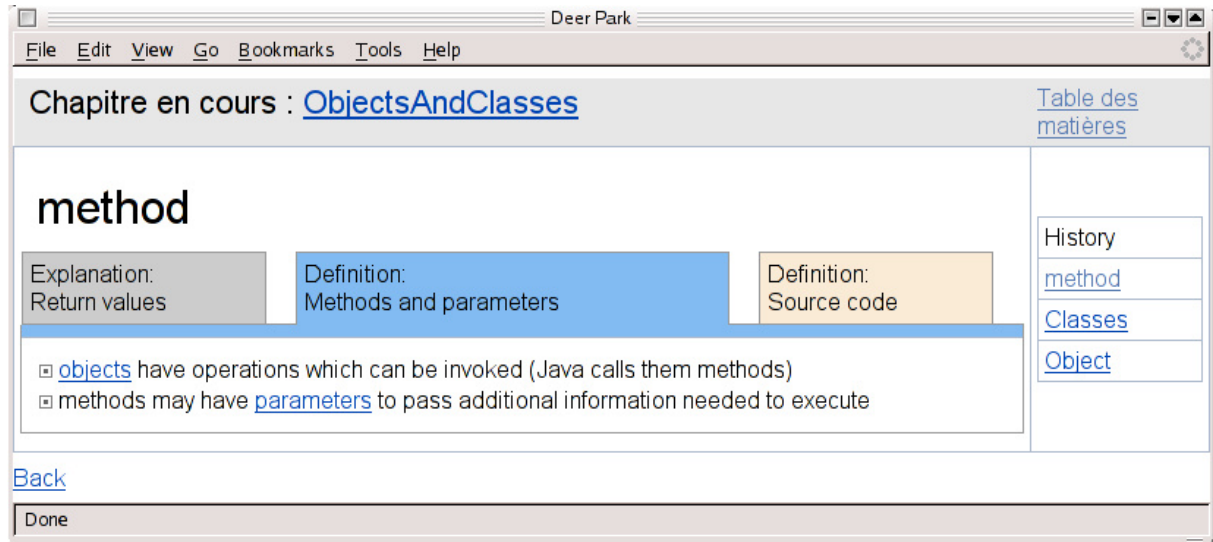


Figure 85 – A screenshot of the QBLS-2 interface with indication of non-visited resources.

Adding the expression presented below to the query presented in the previous chapter (figure 74) informs the stylesheet on the status of the corresponding document. In particular, it determines if the retrieved document has already been visited or not. For a yes/no answer, the operation only consists in checking whether the variable ?visit is bound:

```
OPTION{?user edu:log ?visit
      ?visit edu:visitedDoc ?doc}
```

The query can be refined by setting a time limit. The “date” value is computed inside a JSP script (using specific tags “<%= ... %>”).

```
OPTION{?user edu:log ?visit
      ?visit edu:visitedDoc ?doc
      ?visit dc:date ?d
      ?d > <%=date%>}
```

In the answer, the presence of the “visit” result indicates if the tab must be colored.

7.2.2. Conceptual back navigation

7.2.2.1. Definition

Figure 85 shows at the bottom a “Back” navigation button. Given the fact that learners navigate in a “conceptual space” and not just a hyperspace, classical browser features such as back navigation button may reveal irrelevant, if not misleading for students. In that scope, we propose the definition of a “conceptual back”. This “back” link leads the learner not to the previous document browsed, but to the previous concept requested.

The navigation model proposed in figure 82, can be used to visualize a conceptual back navigation:

- First, we consider the following conceptual space in figure 86, composed of three concepts (“Object”, “Class” and “Method”) and eight resources. To simplify, no link is defined between the concepts in this example. Using the interface, learners can navigate from

a concept to a resource and vice-versa (grey arrows). They can also switch between resources attached to the same concept (dotted arrows).

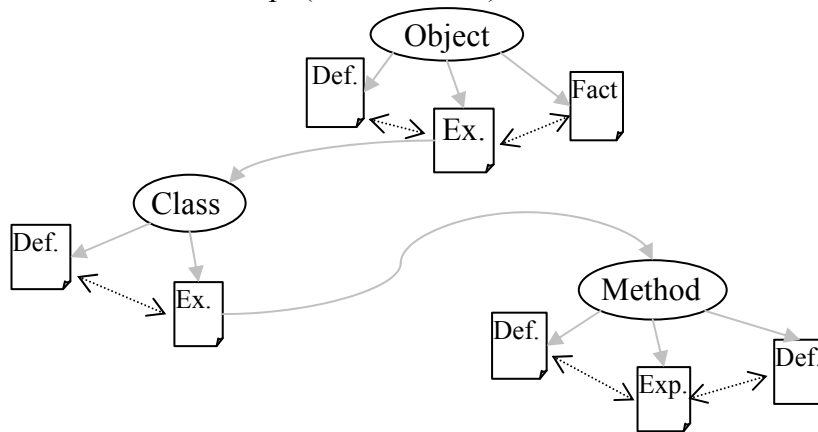


Figure 86 – Example of a conceptual space

➤ A learner navigates this semantic space, and his/her path can be represented on the conceptual structure like shown in figure 87. Like on figure 82, the path is represented by thick black arrows. The edges of the graph are numbered to identify each step of the path. Here, eight steps have been performed to navigate from the concept “*Object*” to the explanation about the concept “*Method*”.

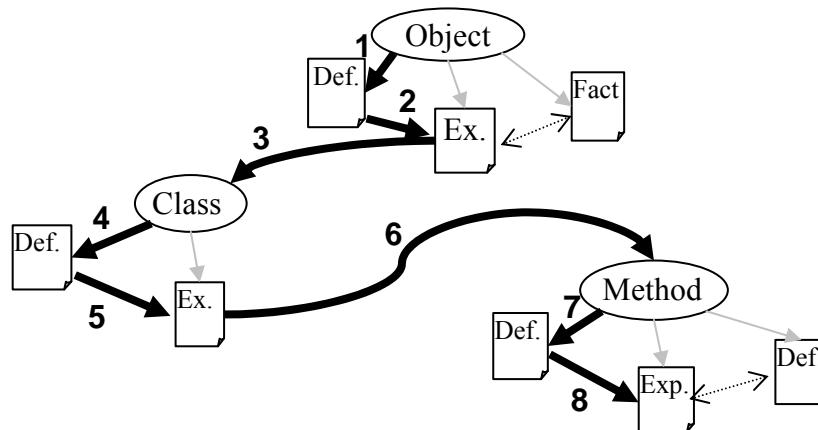


Figure 87 – Visualization of the conceptual navigation

➤ Using this representation of the path, a back navigation link can be computed. A classic “back” would lead, after step 8 to the previous resource browsed (i.e. the definition of method). The conceptual back leads to the concept of “*Class*” and the last resource visited about this concept: the example about “*Class*”. The “back navigation” then computes the thick arrow “back 1” presented in figure 88. The corresponding link is proposed in the interface. If pressing back a second time, the learner will be directed further up in his/her path following the “back 2” link.

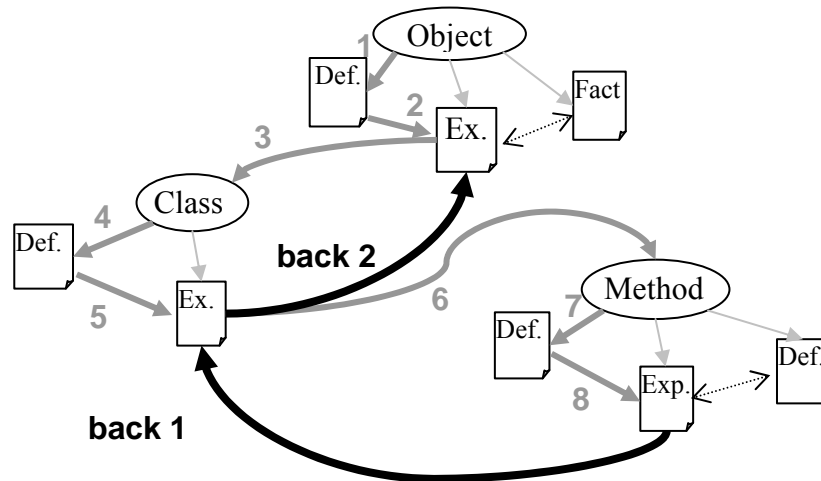


Figure 88 – Illustration of the back navigation on a conceptual space

7.2.2.2. Implementation

The computation of the destination of the back link is supported by the design pattern proposed in the previous chapter: a JSP dynamic tag performs a SPARQL query to retrieve the destination and shows it in the interface as a hypertext link.

The notion of “previous” statement or “last” statement is not provided by SPARQL. However, by combining sorting and comparison of dates, such expressivity can be obtained. We propose the following graph pattern to express that a visit v_2 happened “just before” a visit v_1 :

- Ensure that the date of v_2 is anterior to the date of v_1 ,
- Differentiate v_1 and v_2 according to the desired constraints (e.g. different concepts visited),
- Sort the results according to the dates of v_1 and v_2 , and select only the first result.

The following query, in figure 89, exploits these principles to return the “previously visited” concept and its associated document according to the above definition. In this query the pattern of “previous” visit is used twice, because when the back button has been used on the previous action (like after following the link “back 1” on figure 88), the choice of v_2 must be placed at the beginning of the “back sequence”. In this scope the log model has been enriched to differentiate between requests coming from the back button and requests coming from the remaining of the interface by introducing a sub-relation of “edu:visitedConcept”: “edu:backvisitedConcept”.

This query is quite complex and thus constitutes a good indicator of the robustness of the semantic search mechanism. It has been successfully implemented with the Corese semantic search engine and deployed in the QBLS-2 interface.

```

1. SELECT ?concept ?doc WHERE{
2.   FILTER(?user = <%=request.getParameter("user")%>)
3.   ?user edu:log ?visit
4.   ?visit dc:date ?date^^xsd:date
5.   ?user edu:log ?v2
6.   FILTER(?c = <%=request.getParameter("concept")%>)
7.
8.   { ?visit edu:visitedConcept ?c
9.     ?concept != ?c
10.    ?v2 edu:visitC ?concept
11.    ?v2 dc:date ?d2^^xsd:date
12.    ?v2 edu:visitD ?doc
13.  } UNION {
14.    ?user edu:log ?v1
15.    ?v1 dc:date ?d1^^xsd:date
16.    ?v1 edu:visitedConcept ?c
17.    ?visit edu:backVisitedConcept ?c
18.    ?concept != ?c
19.    ?v2 dc:date ?d2^^xsd:date
20.    ?v2 edu:visitC ?concept
21.    ?v2 edu:visitD ?doc
22.    OPTIONAL { ?v3 edu:visitedConcept ?concept
23.                ?v3 edu:visitD ?doc
24.                ?v3 dc:date ?d3^^xsd:date
25.                FILTER (d3 > ?d2 )
26.            }
27.    FILTER ! bound (?v3)
28.    FILTER (?d2 < ?d1)
29.
30.  }
31. }
32. LIMIT 1
33. ORDER BY desc(?date)
34. ORDER BY desc(?d1)
35. ORDER BY desc(?d2)

```

Figure 89 – Query finding the previous document and concept indicated by the “back” button

This computation does not take into account several back and forth. Thus, the back feature is limited in terms of the number of back steps it can reach. Users can only come back up the first cycle encountered in history. This feature is linked to the choice of the underlying navigation model, showing its interest and originality.

7.2.3. Evaluation of adaptation mechanisms

7.2.3.1. Scalability issues

Adaptation requires the manipulation of a large knowledge base containing the navigation history of the student. On an implementation like QBLS-2, used weekly for three months, the number of recordings per student reached 500, which is comparatively rather small compared to an “intensive” use. Nonetheless the first scalability issues appeared when we tried to compute complex queries such as “conceptual back” navigation on the complete base of records. There is a need for either a more expressive query language optimized for such requests or a richer model that would reduce computations made at run-time.

From a technical point of view, such adaptation mechanisms appear to be a *challenging ground for the generic tools of the semantic web*, given the dynamics and short response time required. Our experiments contributed to the improvement of the Corese search engine, and further research will pursue this effort.

7.2.3.2. *Benefits for learners*

The benefits of adaptation mechanisms like conceptual back navigation, resource selection or link hiding, are not well known. A simple ergonomic principle is stability. Adaptive interfaces that evolve over time may disturb learners by creating an “unsettling” learning environment. In addition, social interactions may be disturbed if interfaces are too different between two learners. In the end, it requires training to get used to such mechanisms and their benefits for learning are still largely unknown.

What we can argue from our experience is that *semantic web offers powerful means to propose adaptive interfaces through its standard formalisms and tools*. However, the interpretations and hypotheses justifying adaptive mechanisms must be conducted carefully. For purely automatic interpretation, we limit ourselves to the mechanisms presented above and will focus in the next part on semi-automatic interpretations involving teachers’ participation.

7.3. *Teacher’s interpretation of activity traces*

Another goal of learner activity recording is to give feedback to the teacher on this activity. We propose to draw the graphs of each user’s path automatically, based on the navigation model proposed in figure 82 and see what emerges from such representations. This proposal is backed by the implementation of a specific module of QBLS dedicated to the analyzing of user traces, called the “QBLS *log analyzer*”.

7.3.1. *Teacher’s interrogations*

Before representing and interpreting the collected data through graph representations, we precise the goals of this process. By interviewing our teacher the following lists of potential feedback that would be helpful was collected:

- Knowing all the concepts visited by a student, or which concepts have never been visited.
- Knowing the order in which concepts have been visited by a student.
- Knowing if students look at the answers.
- Knowing who gave a result without consulting the answers.
- Getting group statistics on the visited nodes.
- Knowing the number of visits for each concept.
- Knowing which resources have never been visited.
- Knowing the percentage of the course that has been visited.
- Knowing, in average, if students used the glossary

In the following, we explain how we answered at least some of these interrogations through an original visualization of learner activity traces.

7.3.2. *Visualization of the course in the log analyzer*

7.3.2.1. *Generation of SVG representations*

For (Mazza and Dimitrova, 04), the goal of using information visualization techniques for student tracking data is to reduce data processing to a minimum. In our case also, most of the

reasoning is performed in the instructor's mind, who draws his/her own conclusions about the students rather than have them inferred by algorithms.

The graph visualization is based on the underlying expression of the course structure in RDF. RDF possesses different syntaxes, in particular a graph representation that is part of the W3C recommendation. By drawing the course as an RDF graph, we obtain the type of visualization presented in figure 86.

However, RDF does not specify how the graph elements should be placed and organized on a two dimension area. As highlighted by the “concept maps” paradigm (Novak and Canas, 06) the disposition of the concepts carries some semantics. To avoid misleading representations of the course structure, we used an automated tool to generate the graphs. This tool, called Graphviz (Graphviz, 06), uses a placement algorithm to minimize the crossings and the total lengths of the edges. On the graph, colors and shapes are used to identify the semantic types of the different nodes.

The graphs are directly generated in the Scalable Vector Graphics format (SVG, 06). This representation can be directly integrated into an XHTML page and displayed in a web browser.

The schema below, in figure 90, shows the process we developed to generate the representation of the course. First, the RDF content is processed by an XSL stylesheet that generates a file in the “dot” format of Graphviz. Then the Graphviz tool generates an SVG representation. The resulting SVG is finally transformed to customize it further than what the Graphviz tool allows.

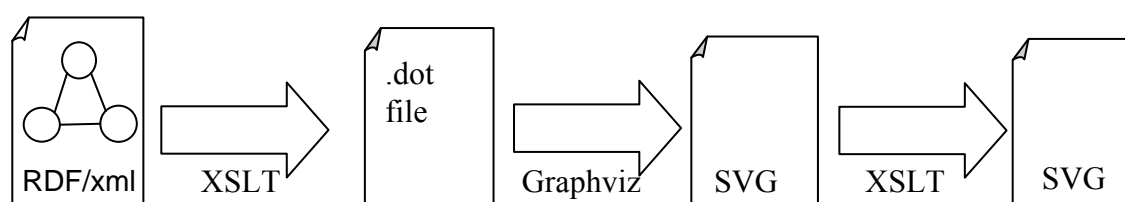


Figure 90 – Process to generate graph representations from RDF files

7.3.2.2. Presentation of the log analyzer interface

The screenshot below, in figure 91, shows the interface we developed to monitor student's activities using graph representations of the course. The central area shows the SVG graph generated with the above method. A dynamically constructed user list can be selected on the left, and a menu bar offers some visualizing options. Nodes in the graph have different shapes and colors. The shapes follow the convention adopted previously: concepts are ellipses and resources are figured by rectangles. The four specializations of concepts are identified by the color of the ellipses as shown in figure 91.

The placement of the nodes of the same type on “lines” has been determined by Graphviz after setting priorities for the different concept types. The node size is customized to fit the corresponding label and does not carry any other semantics. The technical principles behind the remaining of the interface (user lists and menus) are the same as those presented in chapter 6, based on custom JSP tags.

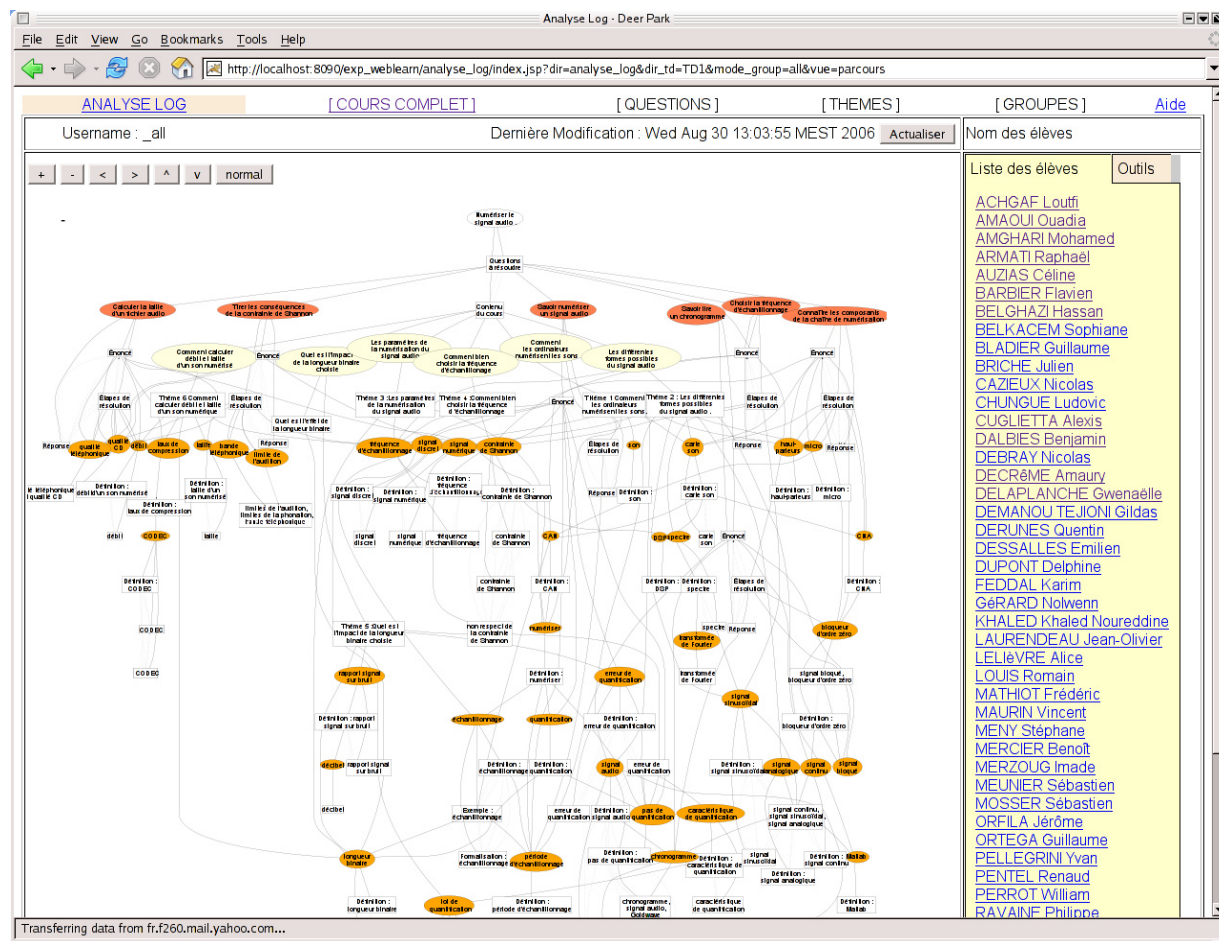


Figure 91 - The log visualization interface

Table 12 – Different types of nodes in the course graph

Course node	Numériser le signal audio
Assessment	Calculer la taille d'un fichier audio
Chapter or main theme	Les différentes formes possibles du signal audio
Domain concept	taux de compression
Resource	Définition : débit d'un son numérisé

7.3.3. Structural views

7.3.3.1. A rich graph structure

Without any information about user's activity, the visualization of the course constitutes in itself an interesting feedback for the teacher. This indeed gives a good overview of the complexity of the course, and of the structure brought by the annotation.

The graph structure is very rich and it is difficult to get a good overview of the potential navigation directions while visualizing the whole graph composed of more than a hundred nodes. The same problem has been faced for hypertexts (McEneaney, 99).

7.3.3.2. Using semantic queries to help visualization

The graph is in fact an RDF graph. It may be used to answer semantic queries, for example to find the nodes that can be reached from a given resource or concept. The interest of this graphical approach is to keep the correspondence between the graph and the semantic information, allowing us to exploit both.

The following query (see figure 92) uses SPARQL with specific graph operators offered by the Corese search engine. It retrieves all the nodes that can be reached from the resource “#Question1”. The operator “all::” specifies to search all possible paths (it does not stop when one is found) with a length inferior than or equal to 22. This number is the longest path that exists in the course. It must be specified to avoid infinite loops when cyclic paths are found. The types `edu:PedagogicalResource` and `edu:loopProperty` are the generic class and property of the ontology. They match all the possible relations that form the graph.

```
1. SELECT ?y WHERE {
2.  ?y rdf:type edu:PedagogicalResource
3.  {FILTER(?y = #Question1) }
4.  UNION
5.  {FILTER(?x = #Question1) ?x all::edu:loopProperty[22] ?y}}
```

Figure 92 – SPARQL query to select paths spanning from one node

The log analyzer collects the results of the above query to highlight the corresponding nodes in the SVG visualization. A stylesheet takes the SVG representation of the course and modifies it according to the result of the query. The SVG graph is then included in an XHTML page. Figure 93 illustrates this principle. All the graphs presented by the log analyzer are created based on this process.

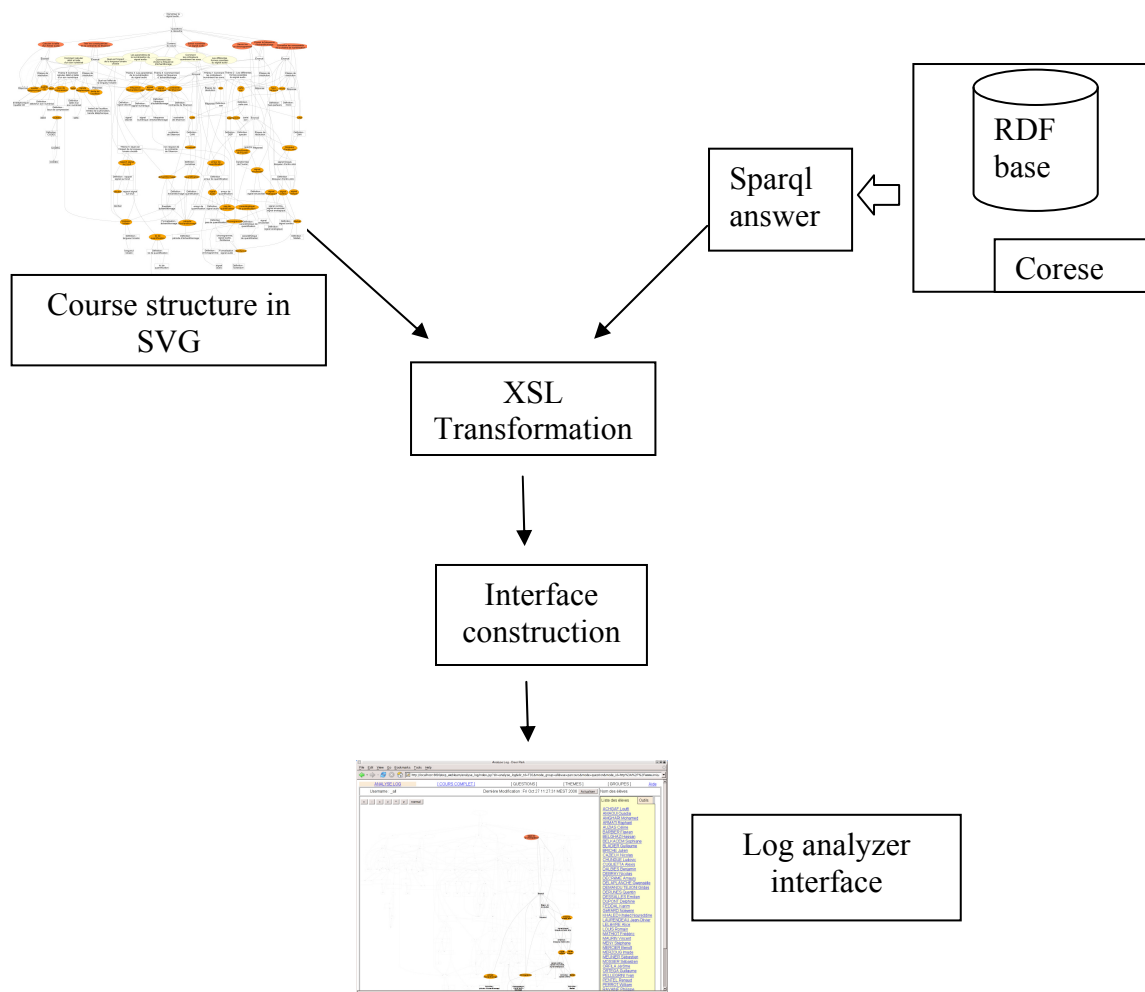


Figure 93 – Construction process for the visualization of the log analyzer.

Figure 94 below, shows the graph with all the possible paths spanning from the question “*Savoir lire un chronogramme*”. This presents a smaller graph on which information about the user will be much more readable for a teacher.

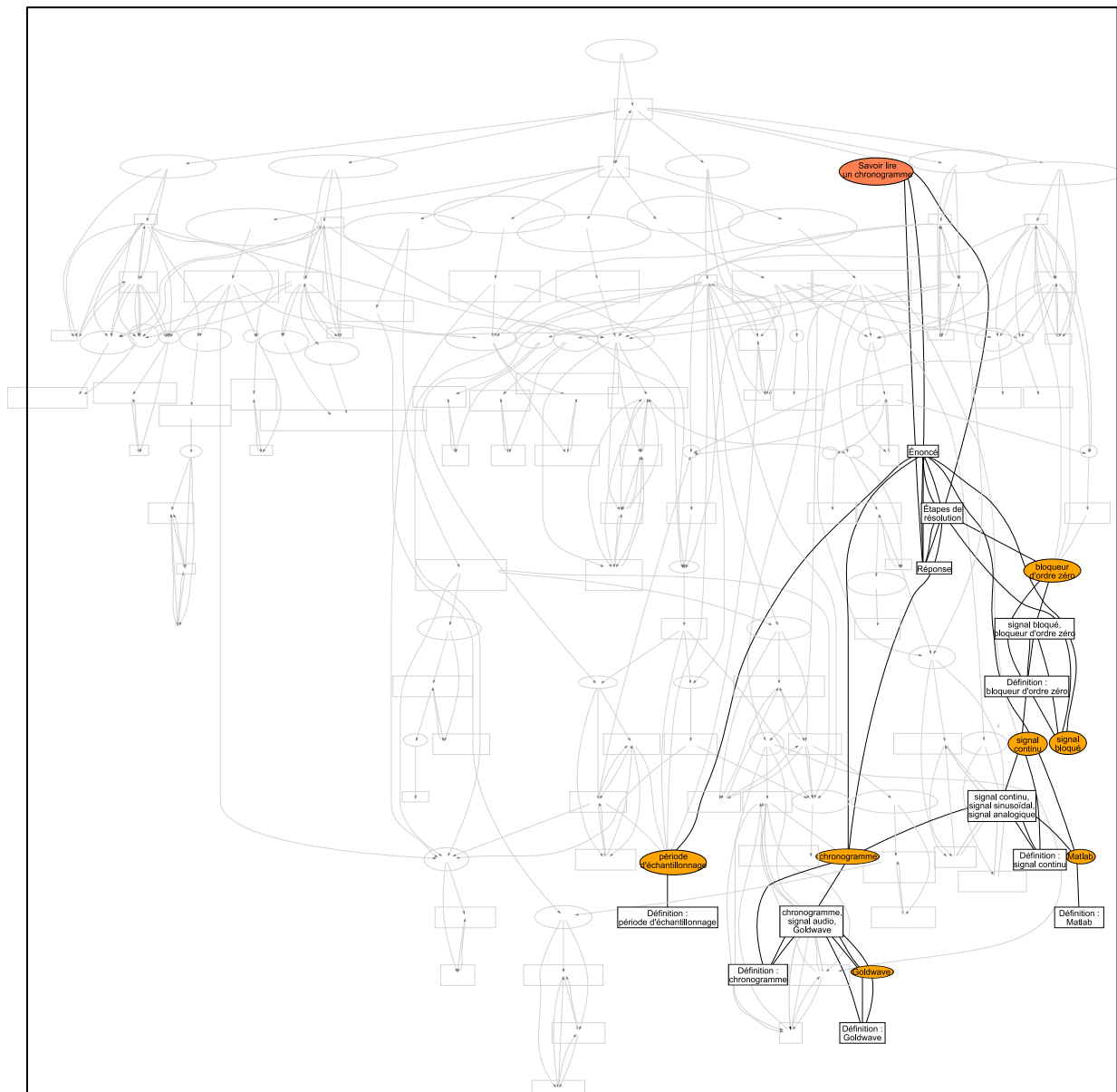


Figure 94 - Selection of a single question and the associated conceptual path

7.3.4. Representation and interpretation of single user navigation

7.3.4.1. Describing a user's path

The graph representation introduces a metaphor with geographical maps. Considering the complexity of the graph, the full path is quite difficult to represent and manipulate. However, with focused representations, such as the one presented above in figure 94, it is possible to track precisely user activity. For example, the user's navigation can be tracked for a specific question.

Figure 95 shows the navigation path of a learner within the “semantic space” as we propose to represent it. The path used by the student is highlighted in red. The width of the edges is proportional to the number of times the learner has performed this step.

On this figure, a teacher would see that the step linking the question (top node) to the first document (the statement of the question) has been performed four times, at four different moments of the course. Exact times are not reported in the graph but from the total number of

steps for this student (166) we can determine that two visits happened at the beginning (20, 57), and two others towards the end (132, 158).

After consulting the question statement twice, the learner came back (step 132) to access directly the answer to the question (step 133). Finally he came back again later to navigate the concepts, this shows that he/she did not try to answer the question before having the answer, but that he did visit some of the concepts associated with the question (“*signal bloqué*” and “*chronogramme*”) at the end (steps 159 to 163). This is valuable information for the teacher to help him/her adapt the teaching strategy. In particular, it gives information on when to make resources such as answers available to students.

Student’s paths may contain some “noise”. For example, when students discovered the interface, the first clicks might not be meaningful as the student navigated to understand the interface but not really to perform his/her learning activity. If this might be the case for the first clicks, we are confident in the quality of the logs. This activity is different from classical web browsing. The task assigned is precise and requires a conscious navigation. The “noise” on log data in such on-line course consultation system should be lower than when considering web browsing in general.

7.3.4.2. *An original overlay model*

Representing user navigation, as a reification of a conceptual structure of the course, constitutes a typical overlay model. A visualization graph based on overlay representation is also used in (Zapata and Greer, 04). However, it only focuses on domain concepts represented through Bayesian networks. One originality of our proposed visualization is that it takes into account both the conceptual aspects (domain concepts) and the actual resources navigated. Thus, it operates a mix between the purely conceptual views and the classical hypertext navigation reporting. The explicit semantic information held by the graph makes this model quite specific compared to other approaches.

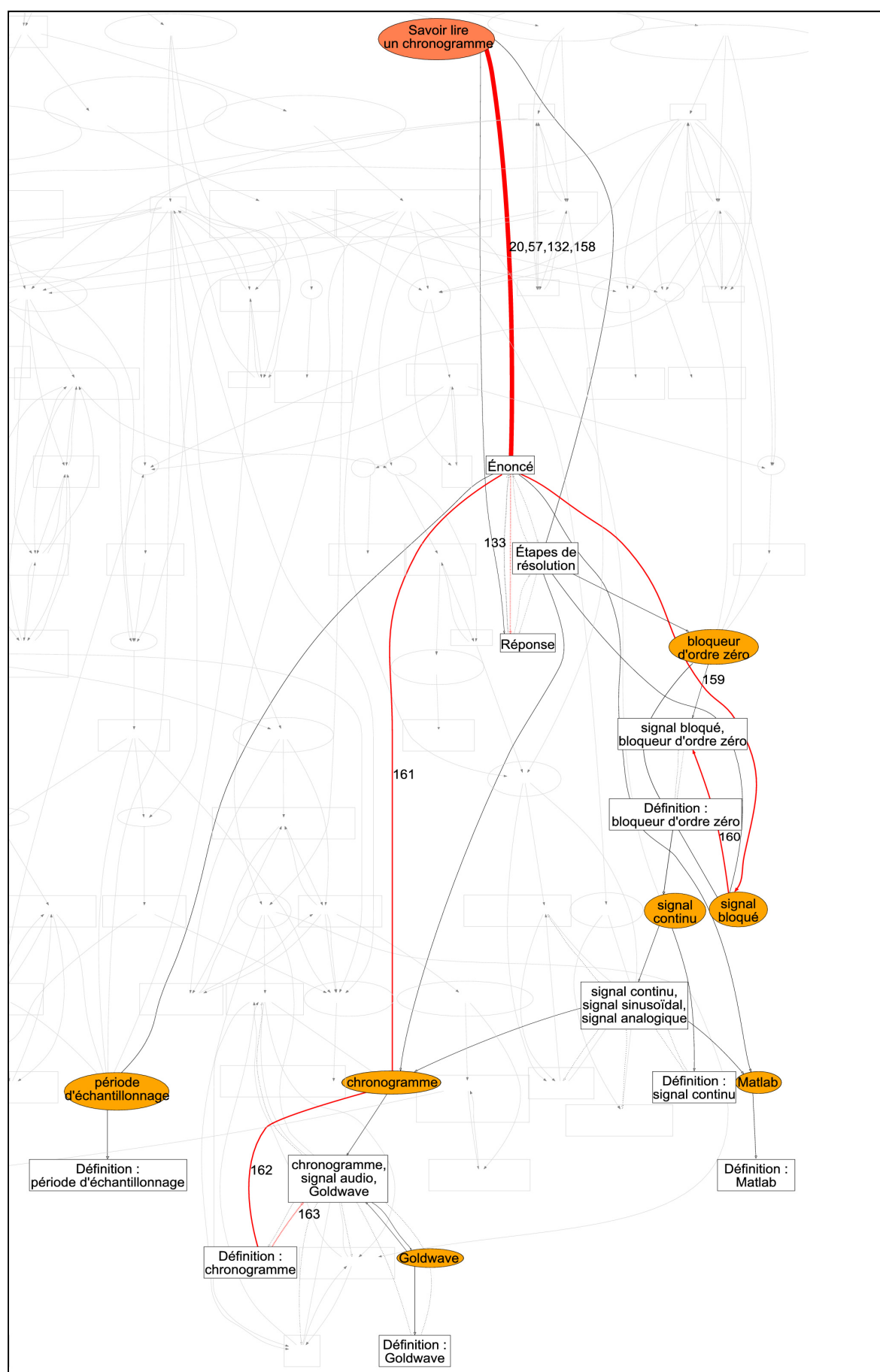


Figure 95 – Visualization of a user path for a specific question

7.3.4.3. *Generic interpretations*

During an interview of the teacher, we collected the following interpretations of learner activity using the type of graphs presented figure 95:

- If a student visited a resource several times, the associated concept is supposed to be known.
- A concept that has never been visited is considered as not known. This is based on the assumption that all users are beginners, for more advanced levels such hypothesis would not be true.
- A high frequency of visits for a node may point to a difficulty. This may contradict the first statement. But in fact it is complementary and illustrates the problem that a concept may be known but hardly understood in the sense of Bloom's taxonomy (see Table 3), as in the "parrot" behavior.

To improve such interpretations, we propose a categorization of user paths. In this scope, domain concepts are valued with a number ranging from 1 to 3 expressing the importance of each concept in the course. Given this valuation, we distinguish the following types of "visits":

- If the learner's path only contains the most important concepts then this consultation is a "*basic consultation*".
- If it contains at least the most important concepts then this consultation is a "*consultation of necessary items*".
- If all the concepts have been visited then it is a "*complete consultation*".
- Finally, if not all the important concepts have been visited then it is an "*insufficient consultation*".

The automatic categorization can be presented to the teacher to help him interpret the user's path. In the case of a "basic consultation" for example, the teacher may identify a student that needs to be assigned further exercises. An "insufficient consultation" profile may need personal tutoring to solve a potential incomprehension, etc.

Heuristics could be thought of to generate the valuation of the concepts automatically. For example, they could depend on the position of the concept in the graph, its links to other concepts, etc. This is a refinement compared to the knowledge expressed in the annotation method (see chapter 5) and thus it represents an overload of work for the teacher.

7.3.5. Aggregated views for analyzing behaviors in groups

7.3.5.1. *Group selection*

Visualizing information about a single user is important for individual feedback. However, for the teacher it is also interesting to interpret the course usage for specific groups of learners, compare their behaviors, etc. For learning assessment, (Delozanne et al, 07) also proposes to first collect and analyze information on each student for each given exercise, then to build a higher level view of one student activity on a set of exercises and finally to build an overview of the whole class activity. The previous section showed how feedback could be given on the first two steps, we now propose to give feedback at the group level.

Groups may be formed by assembling students that perform the labs together in the same room. In this case we can observe "group behavior", see 7.3.5.2. However, a group may also differentiate students according to their background, majors, etc.

Using the underlying RDF formalism, SPARQL queries to the semantic search engine Corese can result in visually aggregating information, for example to evaluate a specific group. All

users' paths from a group are assembled as if they constituted the visits of a single user. The following query, in figure 96, aggregates information about a group "g1". The SPARQL result is formatted by the grouping operator, allowing the XSL stylesheet to translate the result straight away into the SVG graphs shown in figure 97.

```

1. SELECT ?x ?y count ?step WHERE{
6.   ?x ml:vers ?step
7.   ?step ml:vers ?y
8.   ?step ml:user ?user
9.   ?user ml:group "g1"}
10. GROUP BY ?x
11. GROUP BY ?y

```

Figure 96 – Selecting steps from a group

7.3.5.2. Group behaviors interpretation

Comparing two groups may reveal interesting differences. To illustrate this, we compare the graphs of two groups showing behavioral differences. Figure 97 compares group 1 and 2 (administrative groups that did not perform the assignment at the same time). Group 1 is on the left and group 2 on the right. Each edge width is proportional to the total number of steps performed by the members of the group.

Observation

When looking at a question in particular, it appears that the first group accessed the concept “*qualité téléphonique*” from the resource at the top (the question statement). Whereas the second group massively visited the “hints for answering” (90 visits) and went on to read about the concept from the hints. These differences are indicated on the graph by the dotted arrows. Both groups visited the answer from the question header with a similar frequency (42/41), but in the second group many visited the answer also from the hints (2/23).

Interpretation

This comparison shows a different approach between the two groups. The first group did not use the hints, but navigated from the links offered in the statement of the question. The second group was more inclined to use the hints. This result was confirmed by the direct observation of the students performing the assignment as well as by the analysis of their answer sheets. Globally the second group felt more at ease with the system and performed the whole lab in less time than the first group. A potential explanation of this observation is that the teacher was more confident in the short introduction he gave on the system the second time. The second group was then more aware of the possibility to consult hints.

This example illustrates that it is possible to *spot different behaviors by looking at the graphs*. Automated processes should be developed to directly point out such variations to the observer, and just let him/her do the final interpretation.

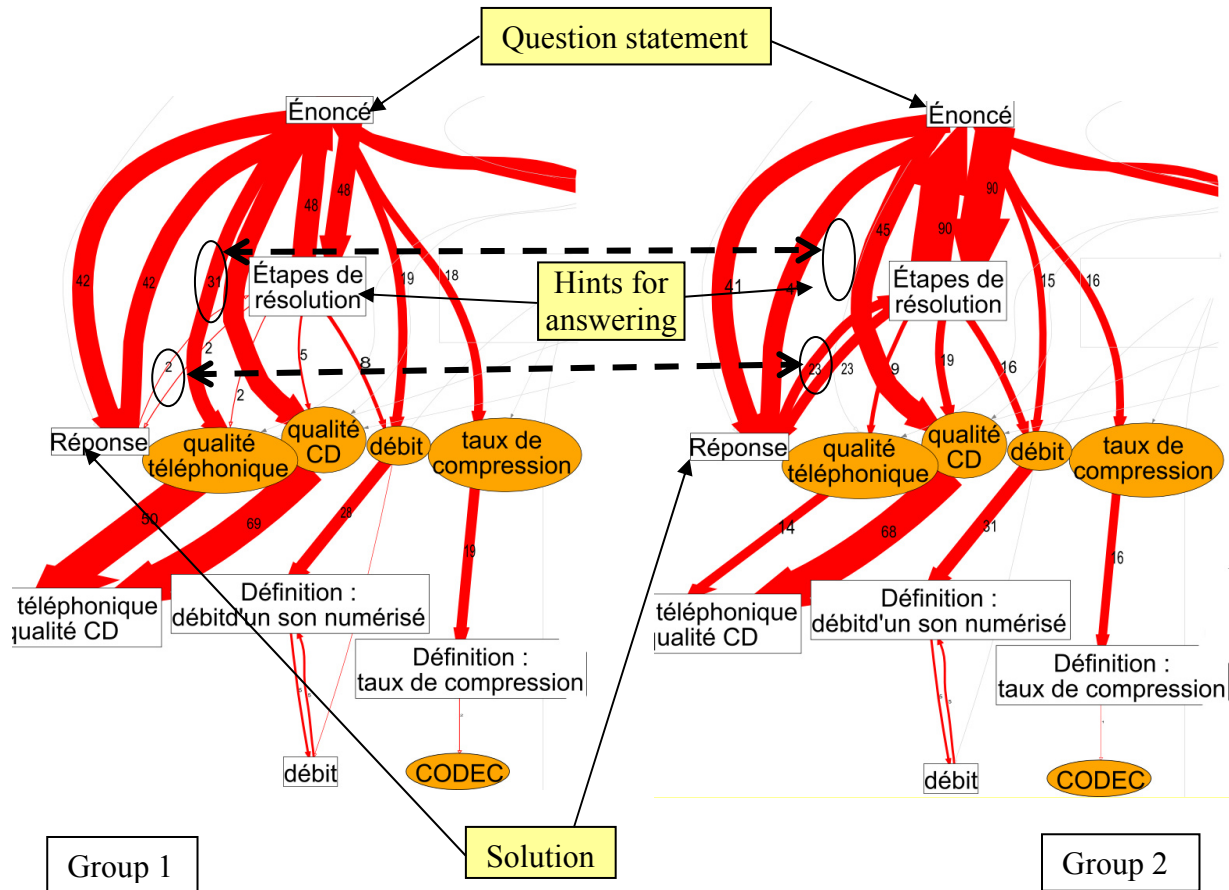


Figure 97 - Comparison of two group behaviors on a focused part of the navigation space

7.3.5.3. Semi-automatic interpretation using semantic queries

Semantic queries, based on the SPARQL query language, can be used to detect behavioral differences in the paths automatically.

For example, frequent visits of hints may indicate a difficulty to answer the question, or a specific learning profile for the group. The query figure 98 retrieves, for each question and each group, the number of users that took a link from a question statement (“*énoncé*”) to some hints (“*étapes de résolution*”). Using this query helps the teacher in his/her interpretations of group behaviors. These indicators already integrate parts of the teacher expertise, so we can talk of semi-automatic interpretation.

```

1. SELECT ?question ?group count ?user WHERE {
2.   ?question edu:enonce ?header
3.   ?question edu:procedure ?hints
4.   ?header ml:vers ?step
5.   ?step ml:vers ?hints
6.   ?step ml:user ?user
7.   ?user ml:groupe ?group }

```

Figure 98 – SPARQL query retrieving information about group behaviors with questions

Other graph inferences can be used to spot other differences using semantic queries, like finding the most visited resource type, the most navigated “concept structure” (only considering domain concepts), etc.

In the end, the application of the navigation model to represent aggregated information offers an opportunity to analyze and compare group behaviors as well as individual ones.

7.3.6. Evaluation of the log analyzer tool

7.3.6.1. Graph aspects

The log analyzer was used experimentally by the teacher during an interview. From this interview, we can formulate the following comments concerning the use of graphs for learner activity analysis:

➤ A path on a graph is an attractive way to represent student's activity. Many other contributions rely on close paradigms to extract and build learner profiles. What is original here is the *integration of log data with the semantic description of the course*. Annotations expressed at the beginning on the original material are exploited to enrich the visualization. In return, it allows rich interpretations to be performed, for example by taking into account complex navigation patterns (including concepts and resource types).

➤ The introduction of graph visualizations is often justified by the “natural” and intuitive nature it would carry. For artificial intelligence scientist it certainly does, but for a regular teacher this is yet to be demonstrated. As any kind of visualization, *it requires a learning phase to get familiar with the representation*. The problem of the size of the graph and the positions of the nodes may negatively affect the interest of such visualization for a teacher. Representation is limited to a two dimensions space whereas many more dimensions exist in learning paths.

➤ The graph supposes a *static structure*. If the structure changes during the navigation, for example if adaptation is performed on the structure itself, the graph may not reflect the choices proposed during the learner's navigation. For example, if several links are hidden at some time, interpreting the navigational choices at that time must take into account only the visible directions. User tracking in adaptive systems thus requires different tools than static graphs.

7.3.6.2. Feedback on the log analyzer tool

The teacher also formulated the following needs concerning the interface:

➤ The tracking system presents interesting features in the context of real-time tracking (during the course). It certainly allows the teacher to propose a better dynamic feedback to the students he/she monitors.

➤ The graph manipulation should be facilitated. Proposals for text-based search and better zoom on the graph were formulated.

➤ More information (see below section 7.4) for a single student should be presented at once, in a sort of personal synthesis.

We claim for a teacher-oriented interpretation of such data. When it comes to tracking and interpreting learners' activity, the scenario and expertise deployed are the major characteristics to take into account. *Without the teacher's experience of the content and of the students, such tracking information is difficult to interpret automatically*. Eventually this expertise can be encoded in complex graph patterns (see 7.4.2).

7.4. More automated analysis

7.4.1. Statistical measures

The visualization of users' paths through the above representation only covers one aspect of user activity tracking. It should be seen as a complementary tool to other approaches based on statistical analysis or classification algorithms for example.

Without going into details, mathematical tools can help classify learners according to characteristic vectors describing each of them. In particular, the mathematical space for the

information might be formed by the number of visits for each concept and document. This results in a large dimension vector. The most discriminating axes can be computed (using Principal Components Analysis for instance). However, the meaning of these axes is very difficult to interpret.

Without using such complex algorithms, simple statistical data about the number and frequency of resource consultation might be interesting. A visualization of such data on the graph, rather than in a list, can complete the analysis.

Figure 99, below, presents a comparison between two identical requests, concerning the most visited nodes of the course. The first visualization on the left uses a simple list display, whereas the second on the right reports this information on the graph using a gradient color. The list is easier to read and the numbers appear clearly. However, not all the nodes can be displayed because of the length of the list. On the graph the overall view is emphasized, non-visited “areas” can be easily spotted in white, and all the nodes are visible at once. It appears that the two representations complement each other.

ANALYSE LOG

[COURS COMPLET]

[QUESTIONS]

[THEMES]

Username : _all

Dernière Modification : Thu Aug 31 18:44:51 MEST 2006

Actualiser

Type	Nom de la ressource	Nombre de consultations	Couleur
Precision	Questions à résoudre :	1972	
Chapter	Numériser le signal audio .	1079	
Test item	Énoncé	921	
Test item	Énoncé	804	
Test item	Énoncé	770	
Test item	Énoncé	592	
Test item	Énoncé	429	
Test item	Énoncé	399	
Precision	Contenu du cours :	315	
Precision	Précision : qualité téléphonique et qualité CD	310	
Procedure	Étapes de résolution	240	
Key Problem	Savoir numériser un signal audio	212	
Definition	Définition : contrainte de Shannon	210	
Domain Concept	qualité CD	209	
Key Problem	Savoir lire un chronogramme	191	
Key Problem	Calculer la taille d'un fichier audio	189	
Key Problem	Choisir la fréquence d'échantillonnage	175	
Key Problem	Connaître les composants de la chaîne de numérisation	171	
Procedure	Étapes de résolution :	151	
Definition	Définition : débit d'un son numérisé	135	
Domain Concept	contrainte de Shannon	129	
Key Problem	Tirer les conséquences de la contrainte de Shannon	126	
Important to know	Thème 1 Comment les ordinateurs numérisent les sons.	123	
Definition	Définition : carte son	121	
Solution	Réponse	114	
Definition	Définition : bloqueur d'ordre zéro	114	
Definition	Définition : CNA	114	
Procedure	Étapes de résolution	113	

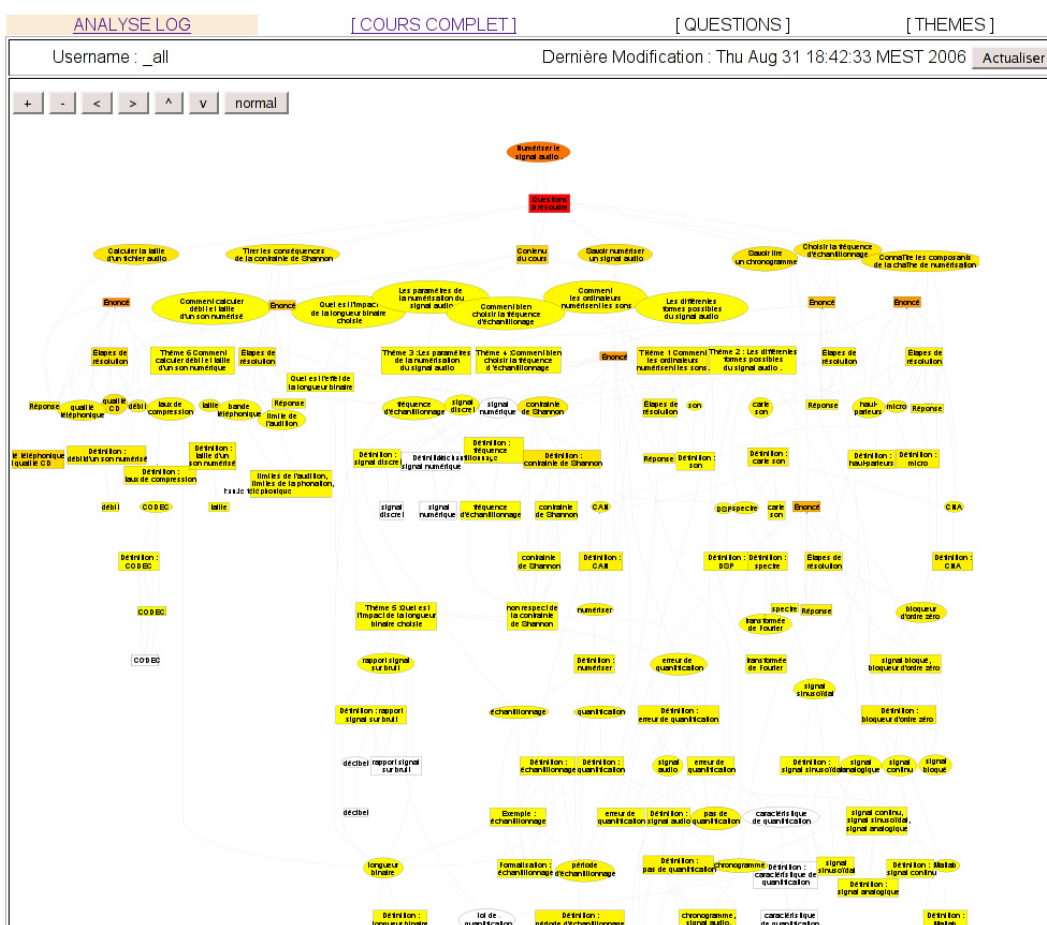


Figure 99 – Comparison between a list display and a graph display for visits frequency

This small example demonstrates the complementarities of different visualizations. From a technical point of view, both representations are built with XSL stylesheets processing the result of the same query.

7.4.2. Graph patterns

7.4.2.1. *Expressing patterns*

During the evaluation of the interface for visualizing log information, two teachers used the interface themselves and pointed out a number of functionalities that they would appreciate in such a tool. In addition to the needed improvements for the manipulation of the graph view, they expressed the need for services that can be regrouped under the idea of using graph patterns to diagnose learner behavior.

The patterns might be either extracted from selected users or explicitly defined:

- To connect this information with more classical feedback like exam results, different groups could be created based on student background, identified behaviors, etc. and matched with the results obtained. A visual check would quickly allow the teacher to find correlations. Once hypothesized, such correlations can be easily verified numerically by querying the RDF base.
- The notion of “typical paths” defined by the teacher also emerged. In the principle of conceptual navigation, a single path cannot cover all the profiles. However, one or several “recommended” paths could play the role of references. Individual differences with the reference might be spotted automatically, and the teacher may decide whether these deviations are critical.

We envision the role of graph visualizations as an exploratory tool for the definition of graph patterns. Visualizing the graph allows teachers to express what regularities they would take into account to help/classify users. As there are so many different patterns possible, the visual input helps choosing which patterns are to be looked for. Then, once those patterns are known, they can be expressed by a technical partner in a query language like SPARQL. Visualizing the paths on a graph would act as a design step to help specify, in practical terms how paths have to be interpreted by the system, and eventually what action has to be undertaken. The visualization allows the teacher to anticipate the results of the queries and thus select the most discriminative patterns. An example of such pattern is given next.

7.4.2.2. *Defining a graph pattern in SPARQL*

We mention SPARQL as a query language able to search for graph patterns, a first example for characterizing groups was presented in figure 98. In this section, we present another example of pattern and how it can be effectively used.

We look for students that do not visit examples when they visit a concept. The following query in SPARQL (figure 100) gives back the ids of the students and the number of times the pattern was found for each of them. The pattern is quite simple: there must be a concept (?c) linked to an example (?d), a visit to this concept by a user (?user), and no visit of the example by this user. Here the “negation by failure” is used to determine that an example has not been visited (if no projection is found then the assumption is false).

```

1. SELECT ?user count ?v1 WHERE {
8.   ?c edu:exemple ?d
9.   ?v1 ml:a_pour_util ?user
10.  ?v1 ml:noeud_cours ?c
11.  OPTION{
12.    ?v2 ml:a_pour_util ?user
13.    ?v2 ml:noeud_cours ?d}
14.  FILTER( ! Bound(?v2))}

```

Figure 100 – Looking for behavioral patterns in the RDF graph

For students who present a high frequency of this pattern, we can assume that those students are not inclined to learn with examples because they did not try to use the proposed ones. We can assign them a corresponding cognitive profile, for example here “*intuitive*” according to (Brown et al., 05). In the end, this *pattern matching mechanism allows us to assign learning “profiles” to students.*

7.4.2.3. Introducing semantic distances

The above pattern is somewhat arbitrary and may fail in detecting correct profile because of their strict definition. In this perspective, we propose to make approximation for the analysis of user paths with graph patterns.

For example, visiting an illustration and an example are closer actions than visiting a definition, and an example using the ontology shown in figure 22. Pattern matching algorithms may use ontological distances to relax the constraint of the patterns and thus return more results. Using the ontological distance calculation proposed in (Gandon et al., 05) the distance between the concept of “illustration” and its subconcept “example” is smaller than between “example” and “definition” which are spread apart in the ontology. By choosing the right threshold, the previous query can be approximated to return patterns where illustrations, counter-examples, etc. have been visited but not including definitions and its close concepts.

This would allow the system to find learners that do not exactly match an arbitrary defined profile but are close it.

7.5. Conclusion

The results of this exploratory work lead to the definition of research directions for activity tracking by exploiting graph properties in conceptual representations of courses. The work presented here possesses many ties with the domain of user modeling (UM) and the activity analysis in hypermedia.

Valued graphs are often used for user modeling. They usually represent the conceptual level (Zapata and Greer, 04) and do not pay attention to the actual resources visited. In hypertext analysis, for example for web sites, only the document structure with its hyperlinks is represented on graphs. The originality of our approach is to rely on a composite structure mixing the “resource” level with the “conceptual” level.

Patterns in hypertext navigation classically characterize user profiles, but they do not rely on the type of the documents visited and their relations to each other. Pattern only rely on structural aspects (Berendt and Berstein, 01).

The interest of visualization for involving teachers in the interpretation loop is underlined by (Mazza and Dimitrova, 04) and we follow this philosophy. The sections below summarize our contribution.

7.5.1. Outcomes of our proposed graph navigation model

We proposed to view an annotated course as a “navigable” structure on which user’s actions (clicks) are represented as “navigation steps”. First, we showed that adaptation mechanisms

can be obtained based on this overlay model. Then, by offering a visual graph representation of this navigation space, we visualize users' paths on a two dimensions structure. The graph representation helps teachers in interpreting student's behavior. Such interpretations can be supported and even encoded in graph patterns.

Interpretations of learner paths are contextual. We demonstrated the necessity to include teachers in the interpretation loop to take into account the context and specificities of the interface, of the scenario, of the social environment, etc. (Delozanne et al, 07) proposes a set of design patterns for learner assessment. In this scope our approach can be viewed as using the following ones, if we make the assumption that learner's path reflect the way they answer a question:

- Specific software for analyzing learner's solution (LA1.2)
- Human assessor to check the automatic analysis of the learner's solution (LA1.3)
- Learner's progression in an individual learning activity (LA2.4)
- Overview of the activity of a group of learners on a single exercise (LA3)
- Overview of the activity of a group of learners on a set of exercises (LA4)
- Automatic clustering (LA4.1)

The originality of the proposal is to combine pedagogical resources and concepts on the same graph. It exploits the RDF underlying formalism and the semantic web technology to develop automated and dynamic visualizations.

7.5.2. Uses of interpreted paths

Based on these results we mention the following directions as potential uses of the generated interpretations:

- It may lead to the modification of the course content or structure. For example, teacher's interviews confirmed the idea that if a concept of large importance had been consulted too often with regard to the rest of the course, there is certainly some difficulty about this concept. On the opposite, if a concept has never been visited, the structure of the course is certainly misleading.
- Automated feedback can be given to the student. For example, links to pertinent resources, given the user's path, can be proposed. This implements adaptive path navigation. It is also called "recommender" feature. It brings the QBLS system closer to highly adaptive systems while keeping it simple from both a technological and usage point of view. The course visualization could also be used directly by students to invite them to analyze their own path and decide themselves on the appropriate directions to follow.
- The pedagogical annotations on the course can be linked to the Bloom's taxonomy. We can envision augmenting the part of automation in the analysis by relying on such association. Depending on the learner's path and the types of resources visited, at least some hints on the "level" of mastering of each concept could be given to the teacher. However, a fully automated interpretation is not a realistic goal.

7.5.3. A broader perspective through the exchange of log data

The chosen granularity for recording user actions is rather low. Eventually higher-level representations that aggregate low-level information of logs into more semantically expressive knowledge might be introduced. Such models would allow the system to get rid of the large static history that reduces the framework performance.

We formulate two recommendations for further investigating this problem with regard to the proposal of (Brooks and McCalla, 06) who envisioned exchanging log information over different systems.

➤ Log information can only be exchanged at a high conceptual level. We have seen that interpretation is tightly coupled with the design intention behind the system interface. Trying to share basic low-level information of resource usage is both difficult given the size of the data manipulated and pointless as interpretation out of context cannot be performed with such data. To ensure the portability of log information, only high-level concepts, based on a shared ontology, can be used.

➤ Current observation means in web-based systems are limited. Unless intrusive mechanisms are introduced (questionnaires, exercises, etc.) the information at hand can hardly justify a detailed interpretation and precise conclusions about learner's ability, skills, knowledge, etc. Activity traces should be considered carefully when used to adapt/customize system reactions. It relies on many interpretations that lead to uncertainty regarding the real needs of the learner. In this scope, learner "profiles" based on semantic approximated patterns should be further investigated.

8. EVALUATION: METHOD AND RESULTS	179
8.1. <i>Generic evaluation of a learning system</i>	180
8.1.1. Specificities of evaluation for learning systems	180
8.1.2. Evaluation focus	180
8.2. <i>Description of the experiments</i>	181
8.2.1. QBLS-1	181
8.2.2. QBLS-2	181
8.3. <i>Evaluation methods</i>	183
8.3.1. Visual feedback through direct observation	183
8.3.2. End-user questionnaires	183
8.3.3. Activity reporting	184
8.3.4. Answer sheets	185
8.4. <i>Results and observations</i>	186
8.4.1. Evaluation framework	186
8.4.2. Usage	186
8.4.3. Usability	189
8.4.4. Acceptability	191
8.5. <i>Synthesis</i>	192
8.5.1. Pedagogical considerations	192
8.5.2. Course reuse	193
8.5.3. Navigation model	194
8.5.4. Global setting	194
8.6. <i>Evaluation framework and generalization of the results</i>	194
8.6.1. Specific methods for specific conclusions	194
8.6.2. Best practices for evaluating e-learning systems	195
8.6.3. Recommendations for evaluating semantic web systems	196
8.7. <i>Conclusion</i>	197

8. EVALUATION: METHOD AND RESULTS

QBLS, as a system and a learning strategy, is a demonstration platform for the technical and theoretical conclusions presented in this work. However, it is also a real system used for learning in the context of university courses. The deployment of the system for a real course brings interesting observations from the theoretical and practical point of view regarding the hypotheses made at the beginning for conceptual navigation in on-line courses and for the propositions exposed in the previous chapters (5, 6 and 7).

We conducted a rigorous evaluation of the use of the system during the real teaching sessions. Local evaluations have already been presented for the annotation method, the efficiency and scalability of the technical choices. An evaluation at an operational level, on the real use case, is necessary to assess the impact of the proposal.

In this chapter, we first clarify the exact goals of this evaluation: The range of possibilities is wide and we justify the specific point of view we adopt. We draw our conclusions from two real applications of QBLS, named QBLS-1 and QBLS-2. We present the evaluation framework we deployed for each of them, and discuss the interest of the different measures at work. This includes positive and negative results concerning the justification of the proposals. A section is devoted to the potential generalization and universal value of the results. Finally, we deliver some thoughts about the evaluation framework itself with regard to the experience gained during its application.

8.1. Generic evaluation of a learning system

The interface and usage scenario of the system are issued from an important design phase. Tools like heuristic evaluations of user tests have been applied (see the interface result in 6.2.1). The results of those evaluations lead to improvements in the design, and these steps are part of the design process. It had important consequences on the conceptual navigation choices as we adopted a specific navigation paradigm after this evaluation. However, this study does not constitute an evaluation of the system itself. In this chapter, the evaluation will not aim at enhancing the system but actually measuring its effectiveness, its impact, etc.

8.1.1. Specificities of evaluation for learning systems

Evaluating the utility of educational software does not only consist in checking if the user can perform an assigned task, but also in evaluating the completion of the learning objective.

The global impact of a system on learning is difficult to evaluate outside a learning theory. Such evaluations belong to the domain of didactics and are out of the scope here. For specific domains, the acquisition of procedural knowledge can be encoded in precise workflows, and then is easier to track down (Merrill, 99). Such evaluation focuses only on one aspect of learning. QBLS and most lecture based courses have a much wider range of learning goals (recall, application, analysis, etc.) the evaluation of which requires specific techniques over long periods of time.

Various methods exist to evaluate learning. In particular, a distinction must be made between quantitative and qualitative methods:

- The quantitative methods aim at measuring objectively the impact of a learning system. Results are usually difficult to generalize. Partial indicators can be collected through tests, exams, questionnaires, etc. For example, the “learning progress” is usually measured by marks attributed to students based on the fact that they performed a given exercise or not. However, it is neither a complete nor a reliable measure. Multiple choices questionnaires allow evaluators to assess a large domain and are easy to collect, but they only target a specific kind of knowledge.
- For the qualitative methods, direct observation remains a highly valuable feedback. Individual observations inform us on the interaction between the learner and the system (e.g. functionalities, difficulties, applied strategies, motivation, etc.). Observation of groups of learners gives information on their interactions with the system and the knowledge they elaborate through such interactions (Gilly et al., 99). Observing a class as a whole helps in identifying its characteristics and functioning. Such data may be complemented by interviews with learners. Interviews can be collective or individual. They can also be opened and structured or semi-structured depending on the level of preparation of the questions (Barfurth et al., 94).

8.1.2. Evaluation focus

What can be clearly evaluated is the realization of the prescribed task. In this domain, problems may concern the content (e.g. the knowledge necessary to answer the question is not present in the offered resources) or the scenario and tool (e.g. the procedure or system does not offer the functionalities required for the assigned task). The focus of the evaluation will not be placed on learning outcomes.

Not evaluating the learning progress of students may sound strange when studying a learning system. Learning as a whole cannot be evaluated directly and we accept the outcomes of the assigned tasks as hypotheses of the learning strategies. The evaluation presented here will

mostly focus on the completion of learner task, rather than on the expected outcome of the task that belongs to other scientific fields not studied in this thesis.

Evaluation will especially focus on the specificities of the system described previously (navigation model, pertinence of the annotations, etc.) from the user (or learner) point of view. The evaluation of the technical means related to the implementation of the semantic web paradigm has already been discussed thoroughly in the previous chapters (scalability, complexity, code size, etc.). From the learner point of view, the system forms a single entity, and only the evaluation of its visible features should matter in this context.

8.2. Description of the experiments

The QBLS tool was used for two separate courses, with different students. Both experiments give us the opportunity to evaluate our system.

8.2.1. QBLS-1

The first experiment was conducted with a group of 47 students, in first year at ESSI school (third year of university). The lecture was planned two days before the first lab. Students were separated in three administrative groups of nearly equal sizes. The labs were not conducted in parallel but happened at different times during the week. Each session took place in the presence of a group of two or three observers. One of the members of this group was also ensuring technical assistance. Each student was facing a computer, eventually its own or the ones already placed in the classroom. Each computer had internet access and thus each student could connect to the QBLS system using a browser (Firefox or Internet Explorer). The interface of QBLS is presented in figure 59. Most students were running Microsoft Windows operating system.

The course content was provided by the teacher who also gave the lecture and supervised the labs. The original document used was a slideshow on the same domain created the previous year, by the same teacher. This is thus an example of reuse of material from one year to the other. The questions have been authored by the teacher and added at the end of the document to allow their integration into QBLS through the importing mechanism described in chapter 5. In its final version, the course is composed of 92 resources.

For students the goal of the two hours lab was to learn how to answer the questions by using the available information. The information proposed by the QBLS interface consists of:

- resources of the original course,
- suggested procedures,
- answers.

The use of external sources of information, like notes taken during the lecture, was allowed. Answer sheets also containing the post-test questionnaire were distributed to the students. They had to reproduce on those sheets their solution to each question, including the necessary justifications and explanations. Room for self-analysis with regard to the given answers was provided in the answer sheets. At the end of the lab, all this material is collected. In parallel, student actions were recorded through the logging system described in chapter 7.

8.2.2. QBLS-2

The second experiment targets a larger group of students on the domain of Java programming. Like the previous experiment, it took place in the normal curriculum of students. Students' level was varying from total beginner to little experience in Java programming. The whole

course consists of bi to tri-weekly lab sessions (of 2 hours) during 14 weeks. The course involves 84 students in first year of engineering. All the students were different from the first experiment, except one. During the first 8 weeks, the course is solely constituted by the labs using the BlueJ programming environment. Then, formal lectures are given using the original slide show loaded into the system.

The labs are supervised by a team of four teachers, each one usually supervising the same group. Only one of those teachers authored the assignments and another one annotated the material chosen by the first one. The assignments to perform during those labs consist in programming small pieces of code or answering questions. They are placed on a wiki. Before the labs, the teacher who annotated the material added links in the wiki towards QBLS using a specific macro.

Figure 101 shows the QBLS-2 interface (in the small frame) displaying two resources (a definition and an example) found about the concept of “accessor method” in Java with their associated pedagogical type and title. The frame in the background is a snapshot of the wiki interface displaying a list of questions/exercises and a link towards the concept of “accessor method”.

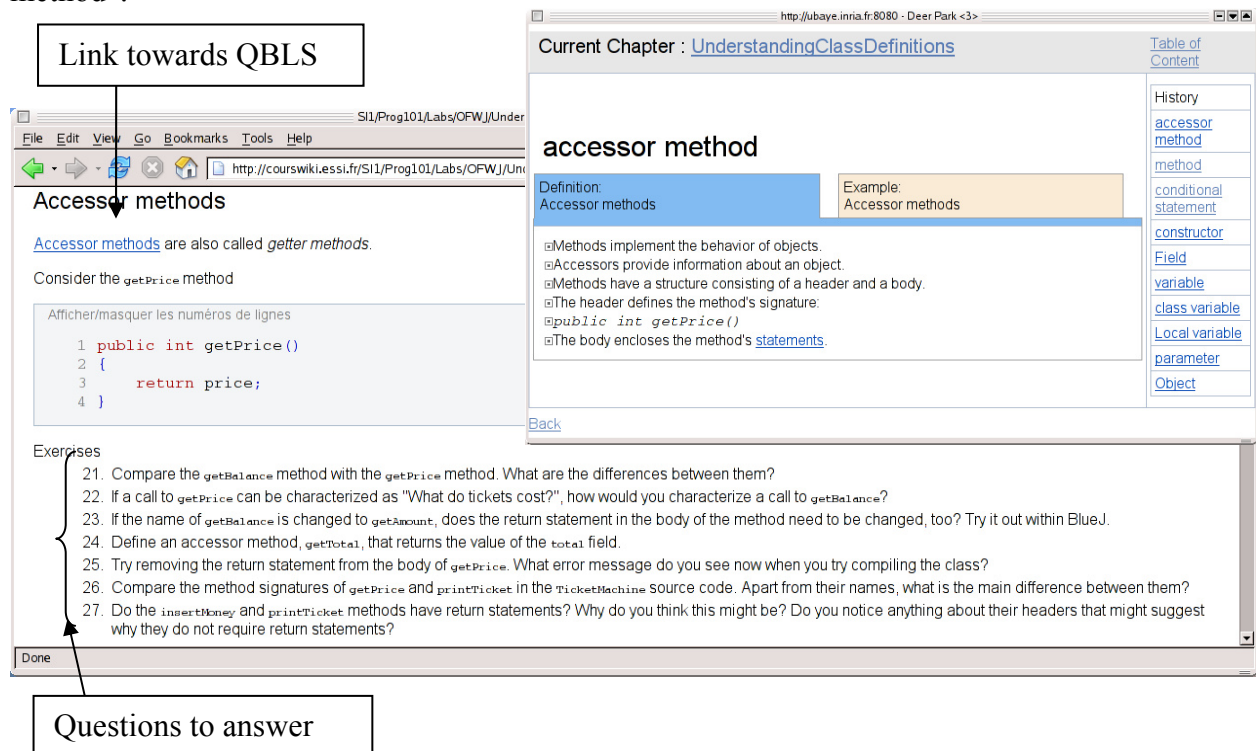


Figure 101 - Screenshot of the QBLS-2 interface accessed through a wiki page

The course content was entirely reused from slides publicly available on the BlueJ web site. It contains a total of 443 slides constituting the pool of resources. Our experience is thus an example of a real attempt at reusing external resources from the web.

Students had to write their answers to the questions on their own wiki page. The activity on the system has been recorded throughout the 4 months of the experiment. Students were identified by their wiki login. This enabled us to track precisely their activity for analysis. In addition, they answered a questionnaire for a user-centered evaluation.

8.3. Evaluation methods

According to (Nogry et al., 04), several methods must be combined to evaluate the impact of a learning environment. We proposed to deploy four evaluation procedures through the experiments (I and II). They are based on the following information sources:

- Direct observation
- End-user questionnaires
- Activity recording, in particular, the recording of navigation paths
- Answer sheets analysis

These evaluation indicators do not require too much involvement from the student (only a questionnaire is filled), but they highly rely on the teacher's expertise. They give a good framework for evaluating a system in a real learning situation without overloading the learners. Each of the four sources mentioned above contribute to the evaluation described in the following.

8.3.1. Visual feedback through direct observation

The first evaluation, and most simplistic one, is obtained through direct observation of the learners interacting with the system. This is one of the way teachers monitor student's activities during classical face-to face labs. We put this in practice in QBLS-1 by introducing three "passive" observers in the lab room. In QBLS-2, we only relied on the observations gathered by the teachers supervising the labs.

The drawbacks of this method are multiple:

- With one observer, only one learner at a time can be observed, so this observation misses the activity of all the others.
- The learner knows that he/she is being observed. This is both uncomfortable and intrusive for the learner's activity. Learner's reaction could be very much affected by the presence of the observer, and thus the observed behavior might not be the one adopted when alone. Moreover, the necessary short time spent with the observer (if all the students have to be monitored) does not allow learners to get accustomed with this observation.
- The observer has preconceptions about the student and on what he/she wants him/her to do. It results in a possibly biased conclusion with regard to the actual observations.

Despite its flaws, direct observation reveals information that would be difficult to capture otherwise. Teacher's expertise is particularly emphasized in the evaluation and other observation means (questionnaires, log analysis, etc.) are often used to confirm visual feedbacks. Observation largely decides on which hypotheses must be confirmed or invalidated using the other means of evaluation.

In on-line courses, teachers miss this valuable feedback. This is perceived as a major frustration. This is a strong argument showing the importance of this input.

8.3.2. End-user questionnaires

Questionnaires are usually used to get feedback from a large group of users (ex: mass surveys). They should be anonymous to ensure a total freedom of speech for the learners. To bring meaningful results, questions must target specific issues and must not be confusing for learners. We present the questionnaire we developed in annex (annex 2) as an example for such evaluation.

In QBLS-1 and 2, post-experimental questionnaires were filled in by learners. The questionnaire contains 30 affirmations targeting different issues that could not be evaluated

through observation. It is separated in three parts. A first part is based on a *System Usability Scale* (SUS) grid recognized in the domain of software evaluation (Brooke, 96). It also contains more specific affirmations about the interface and the way participants used it. A last part tries to get a grasp on the realization of the assigned task and the learning outcomes.

The answers to the questionnaire use the Likert scale (Likert, 32), which consists of a numeric scale (1 to 5 in this case) that the participant uses to position his/her agreement or disagreement with the proposed affirmation. This is an interval scale that allows us to compare the amplitude of the different scores given by the participants.

8.3.3. Activity reporting

Through the computer-mediated interaction, student activity is recorded. Obtained traces guide our evaluation using two different methods: a graph based analysis, presented chapter 7, and a statistical analysis of the raw data.

8.3.3.1. Graph based analysis of logs

The log analyzer system presented in chapter 7 was used to evaluate student's activity with QBLS. The graph visualization may apply to the representation of a large number of situations where navigation in digital resources is involved. To use this tool in particular, the navigation must comply with the proposed navigation model (see 7.1.2.1).

In QBLS-1, students only interact with one interface in a well-defined navigation space. This allows us to use the log analyzer tool for this experiment. In QBLS-2, interactions with external pages containing the questions made it difficult to monitor student's activity as coherent paths. Recording students' actions in various interfaces provided by different servers raises a number of technical problems that could not be solved at that time.

8.3.3.2. Statistical analysis

A common way to analyze activity traces is to adopt a statistical approach for exploiting the data. Simple measures like means, frequencies etc. give explicit figures about groups or individual activity with the system. Existing reporting tools for LMS such as (Mazza and Dimitrova, 04) do not currently integrate the idea of conceptual navigation. Then we processed the collected data from "scratch" using SPARQL queries and existing spreadsheets tools.

We emphasized the use of statistical measures to evaluate learner activity particularly in QBLS-2 experiment where the system was used over a long period. The large sample of data collected gives more significance to such analysis.

The information collected is based on the learner clicks. They allow us to determine the time and frequency of connections. For example, figure 102 shows the evolution of the number of clicks per day on the system during the four months of the experiment.

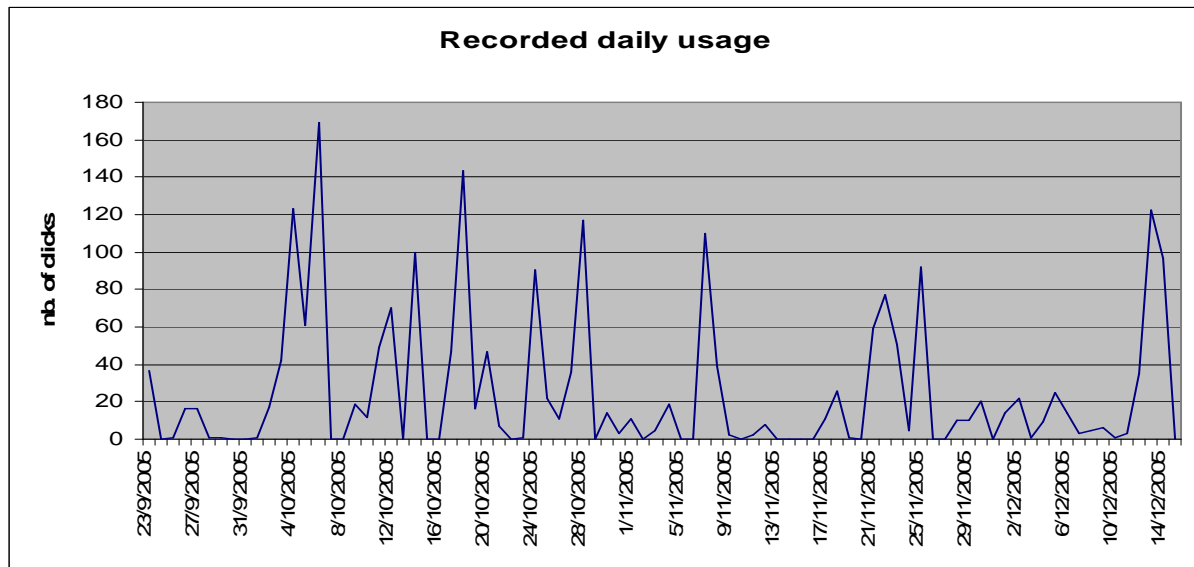


Figure 102 – Usage statistics over the four months of the experiment

Other interesting indicators are the analysis of the “sessions” of each user. By analyzing the dates and times, each “activity sequence” can be isolated and analyzed. Statistical analysis reveals a very rich information source, but like log analysis, it requires an interpretation phase that depends on the interface specificities, the assigned task, etc.

8.3.4. Answer sheets

In QBLS-1, students’ answers were collected on paper. In QBLS-2 students had to produce their own wiki page and write their answers on it. Pages could be accessed by teachers. Generally speaking, learners’ productions are a rich source of knowledge about their understanding and mastering of the subject to learn. Evaluation during exams is often based on such specific productions. In a lab situation, even if no formal assessment is performed by the teacher, students can be asked to produce a readable document for control purposes.

Exploiting this complex source of information requires a large amount of work from the teacher who has to read and analyze each production. Nevertheless, it is a powerful way to evaluate the effectiveness of learning and the potential difficulties met by learners.

In QBLS-1 the teacher gathered the answer sheets produced by the students and identified the types of errors/misconceptions to better evaluate the learning process of each of his/her students.

The following categorization of errors is proposed:

- Wrong or incomplete answers to questions.
- Learner detected a mismatch between his/her result and the given answer.
- The correction of the result is wrong after detection of a mismatch.
- Total incomprehension and ambiguities even after consulting the answer.

The identification of the errors is based on the analysis of the answer sheets produced by the students. The difference between errors made before or after reading the answer can be determined as students were asked to write the potential corrections in a separated section on their answer sheets. Such feedback possesses a high value in term of evaluation. However, it cannot be used on a regular basis but only for experimental evaluations.

Due to a lack of human resources, such detailed analysis could not be performed in the context of QBLS-2.

8.4. Results and observations

In this section, we analyze the results of the evaluation methods presented above through an evaluation framework based on three dimensions: utility, usability and acceptability. After presenting briefly those dimensions, we present, for each of them, the results that allow us to draw meaningful conclusions about the QBLS system in general.

8.4.1. Evaluation framework

According to (Tricot et al., 03) there are three dimensions for evaluating a computer learning system: utility, usability and acceptability.

➤ The *utility* evaluates the correspondence between the functionalities offered by the system and those needed by the user to reach his/her objective. At the highest level, utility compares the defined learning objective to the effective learning. Generally, this is evaluated by the observation of the capacity of the user to perform his/her task by using the system. Such evaluation requires the definition of clear objectives.

➤ *Usability* focuses on the possibility to use the system, either intuitively or after a reasonable learning time (in the latter the learning phase must be clearly described and take part in the evaluation). Usability may concern the interface, the navigation and its coherence with the final objective, the system in general, etc. Usability criteria as defined by (Nielsen, 94) can be applied with care in the case of educational software. However, educational software is quite specific compared to “classical” information systems. The high-level objective is learning and not only a direct access to information.

For example, analysis of navigation paths is mentioned as a potential way for evaluating the usability of a system. The shortest navigation path is usually preferred, but for learning, a long path that allows learners to visit several important concepts might be preferred to a shorter one that gives the answer to a question directly but presents fewer benefits for learning and understanding.

➤ *Acceptability* is the user’s mental attitude towards the system, especially the general feeling about it. It greatly influences the system integration into day-to-day activity. Various parameters influence acceptability, like cultural background, social environment or motivation. The perceived usability and utility of the system also influence this evaluation criterion.

In this chapter, any occurrence of utility, usability and acceptability directly refers to their definition in this evaluation framework. As explained by (Tricot et al., 03) the three dimensions can influence each other in all possible directions.

8.4.2. Usage

In both QBLS-1 and QBLS-2, student’s goal is to answer questions and produce answer sheets, either on paper (QBLS-1) or electronically (QBLS-2). We try to evaluate the realization of this goal by looking at the sub goals of this task: accessing the system, reading/navigating the course and answering.

8.4.2.1. Accessing the QBLS system

Evaluating the activity of students with QBLS first relies on the observation of the number of accesses, on their frequency, etc. For QBLS-1 an average of 300 clicks per students has been

recorded, coupling this result with the observations during the labs, it shows that the tool was effectively used for the assigned task.

In the case of QBLS-2, we recorded that in spite of being an optional tool, a majority of students have used QBLS-2. Only a third of them accessed more than 10 pedagogical resources out of the 359. We propose the following explanations and interpretations of this result:

- The use of the system was only suggested through links in the questions but there was no obligation.
- Paper handouts of the original slides had been distributed.
- A book was also available for students.
- Some of teachers in the team gave paper handouts of the questions themselves, so it was impossible to click on them and access QBLS as intended.

As a result, the impact of QBLS-2 “only” concerns a third of the students. However, we can claim that those users have really been attracted by the functionalities of the system. This shows the usefulness (at least a third of the students) of this access to knowledge, even if other sources of information (classical slides shows, paper handouts, books, etc.) are available.

We deduce from the above that *the system was effectively used for answering questions*.

8.4.2.2. Conceptual Navigation

The main feature proposed by QBLS is conceptual navigation. To evaluate the system usage, the impact of this feature must be evaluated.

Resource coverage

The goal was to make content more accessible and easy to navigate so that students may take a better advantage out of the proposed material.

In QBLS-1, in spite of the attention paid by the teacher to connect the resources with appropriate concepts, some resources were not visited at all (10%). In QBLS-2, the part of the material that remained unseen is slightly higher (20%).

Forcing the students to go through all the material would imply some kind of restriction in navigation freedom, which is opposed to the chosen pedagogical paradigm. Given the freedom students had in their navigation, this coverage indicates a much better result when compared to the previous poor usage of the available paper material reported by teachers.

The interpretation of the non-visited resources is delicate:

- It could be explained by a poor linking, and ill-structured conceptual paths. In this case, the semantic annotations are certainly not suitable for the assigned task.
- It may also show the “utilitarian” behavior we expected from students: they only browse what is relevant for their current focused problem, and indicated that the semantic annotations lead them directly to the necessary material.

Generally speaking, such figures should be taken carefully as the collected information does not indicate a real usage of the content. If we can guarantee that some resources have never been visited, opening a resource does not mean that it has been read thoroughly or that it benefited the user.

Another indicator could be the time spent on a resource. If this is technically possible, it does not give meaningful results in the context of QBLS because too many parallel activities may be conducted during the labs (reading the course, testing hypotheses, writing answers, etc.). The monitoring of time requires very specific tasks and thus does not apply here.

Navigation using pedagogical information

We also tracked specific behaviors by coupling the above information with the knowledge structure defined in the annotations. In particular, we examined the use of the tabs for navigating between resources attached to the same concept.

We observed in QBLS-1 that the tabs “formalization” and “precision” were not clicked as often as the other two: “definition” and “example”. The pending question was asked in the questionnaire: “*I find that the titles of the tabs like « precision » and « formalization » well indicate their content*”. The mean score of 4.4 to this question indicates a good understanding of the meaning of those terms, and the information they give about the content of the resource. By coupling the results of both evaluation techniques, it reveals that the only interpretation possible for the inferior use of those tabs is that students did not find the resource types “formalization” and “precision” useful in the context of their question-answering task.

In QBLS-2, the same study did not reveal such differences. We monitored that roughly 50% of the time the users visited the different tabs for a concept from left to right. This gives important information about the navigation strategy deployed by learners: Half of the time, they followed the order proposed, and the other half they decided to click specific resources and to follow the order. Both strategies are to be found in most individual records so it is not related to cognitive profiles. This emphasizes the need for both an *automatic semantic ordering* of the proposed resources and a presentation of explicit labels in the tabs that allow *users to make their choice*.

Glossary access

Finally, we monitored the use of the glossary in QBLS-1. Its role in the interface was to offer a direct access to the concepts without having to navigate the conceptual structure to find them. In this sense, an important use of the glossary would invalidate the navigation strategy chosen to perform the QBLS task.

The glossary has been used very few times and the answer to the question “*The glossary helped me during my task*” obtained a mean of 2.1.

There are differences in use between the groups: the first group revealed more curious about the interface as 65% of users opened the glossary at least once, only 40% and 39% respectively in the other groups. In average, the number of clicks is the same, around 3.

This very low impact of the glossary shows the importance the navigation space.

A result of this study is that the *structured navigation seems preferred over a simple flat model* based on a glossary access.

In addition, the proposed *navigation fulfils its role*. Both *domain and pedagogical information are exploited*, and the *structured navigation is preferred over a glossary access*.

8.4.2.3. Answering

A usage evaluation would not be complete without measuring if the task of answering questions could effectively be carried out. In QBLS-1, all the students completed the six assigned questions. Not all the results were good but this definitely shows the system utility.

The primary goal of *supporting question based learning is reached*.

In the questionnaire for QBLS-1, the evaluation of the perceived learning efficiency of the system is held by questions n° 1-7 and n° 18, 19, (see annex 2). To confirm that the answers to those questions converge towards the same conclusion, the value of the Cronbach’s alpha has been determined. The Cronbach’s alpha indicator reveals the part of the variance of the

true result with regard to the total error. In other terms, it allows establishing that the questions inside a group are correlated.

The analysis has revealed a value of 0.73, which is sufficiently high to consider those nine questions as correlated. The mean score of this group of questions is 4.25 out of 5. It is then a very good result concerning the efficiency of the system as perceived by the students.

A deeper analysis of this group of questions leads to the distinction between two pairs of questions:

➤ *“I’ve been able to realize the assignment without the paper handout of the course” and “I could easily find in the on-line course the necessary information to answer the questions”*

➤ *“Through the resolution of the questions I acquired a global vision of the course” and “I will revise my course for the final exam by trying to answer those questions again, I find it an efficient system for revising.”*

The means for those questions has revealed a significant difference ($t(89)=2,53$; $p<0,025$). The best score (4.43) is obtained for the first pair of question. This allows us to consider the QBLS-1 setting as more adapted to the realization of exercises than the theoretical learning of concepts. However, the mean for the second pair remains high (4.1).

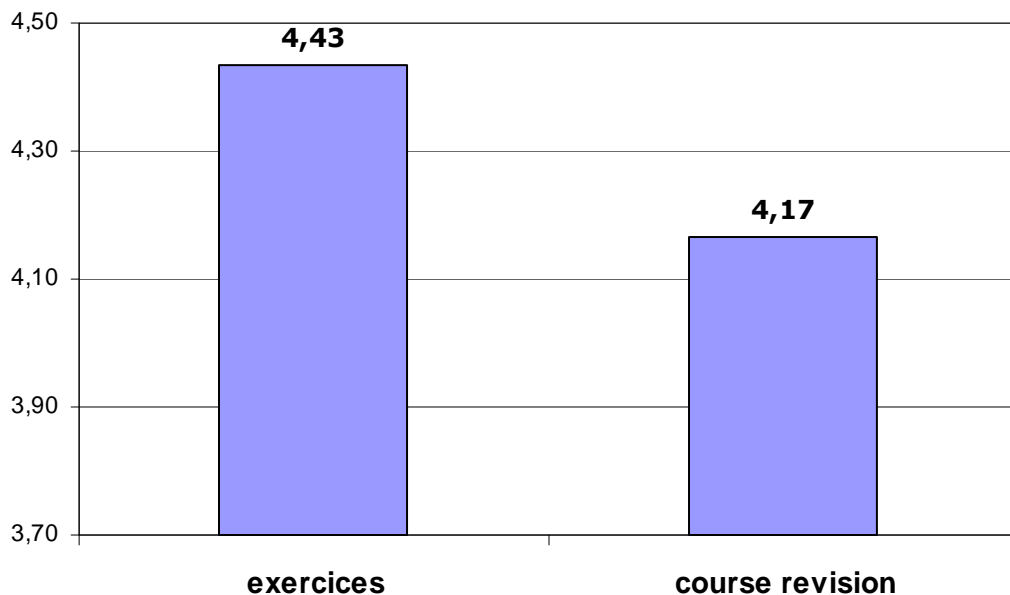


Figure 103 – User’s evaluation of QBLS support for two goals of on-line course systems

For QBLS-2, a similar group of questions obtains a mean of 3.7, with a large dispersion of the answers. This shows that in this second experiment less students were interested by the system.

QBLS targets *more the realization of exercises than the theoretical learning of concepts*.

8.4.3. Usability

From both observation and questionnaires results in QBLS-1 and QBLS-2, a good interaction with the interface was observed. The results of the SUS questions particularly highlight this (n° 20-24, see annex 2). The analysis shows that the average score for this group of questions is high: 4.3/5. It confirms that the level of usability of the interface obtained through the design phase (see 4.4) is good.

To track more specific issues about usability, we specifically evaluate three important points:

- the time necessary to understand the interface,
- the impact of the conceptual navigation,
- the way the system fits in the pedagogical scenario.

8.4.3.1. *Evaluating student's learning phase*

After a short introduction, given orally by the teacher, the student discovery and learning phase was quick (few minutes in general for the observed subjects).

In QBLS-1, we observed few cases where the list of questions was not found immediately leading to some incomprehension of the assigned task. After a short demo by the teacher, or by another student, the system did not reveal major usability issues.

Familiarization and learning time was reduced as a similar interface was used for the lecture, instead of the usual slide show. If this revealed difficult to handle for the teacher, it undeniably helped the student by demonstrating the interface principles.

For QBLS-2, the learning effort mostly focused on the use of the wiki. The QBLS interface seems well understood. Very few students used the table of content for navigation. This may be due to a lack of training about this feature but also confirms the result about the lack of interest for a decontextualized access like a glossary.

The confirmation of the easy manipulation of the interface can be found in the stable and affirmative answers given to the affirmations “*I could realize the assignment without the help of the teacher*” (3.9/5 and 3.8/5), and “*I think most users learn quickly how to use the system*” (4.5/5 and 3.8/5)

We conclude that *the interface is intuitive enough for the targeted students*.

8.4.3.2. *Impact of semantic navigation on usability*

Semantic navigation might be disorientating compared to classical hypertext navigation. In the questionnaires, students very positively rated the fact that they can easily access the relevant resources of the online course to perform exercises (result obtained in both experiments). On the contrary, they rather negatively rated the usefulness of the tab headings indicating the pedagogical type of the resources (e.g. Definition, Law, Example, etc.).

At first, this could be interpreted as a low interest of the pedagogical annotation. But looking at how this information is used we recorded that 50% of the time they didn't just visit the resources in order from left to right but made a deliberate choice, necessarily based on this pedagogical information.

The display of that semantic information might not be relevant for everyone but its interpretation by the automated system is crucial. Beginners, as we have observed them, cannot exploit this information at the early stages of their progression. They only benefit from it through the “intelligent” system that can handle this knowledge.

We conclude that *annotations are of prime importance to organize resources according to a pedagogical progression* and at the same time that *the complexity of the model must be hidden for the end-user*.

An unexpected behavior we noticed in QBLS-1 was an important use of the browser's “back” button, whereas a history list of the visited concept was available and offered to come back to the previously visited concepts. This highlights an important aspect for semantic web interfaces:

Existing practices of hypertext navigation plays an important role in the evaluation of usability of new navigation paradigms.

8.4.3.3. Coherence of the pedagogical scenario

In QBLS-2, two different tools are manipulated: the wiki, offering questions and a place for writing their answers, and QBLS to read and learn about the course. Both interfaces use similar colors, fonts and general layout. As a result, some students did not realize that they were using different tools (a wiki and QBLS). This is revealed by the free-text answers in the questionnaire.

In QBLS-2, usability issues are more related to the use of a wiki for displaying the question, which might be disorientating for students. Some of them declared that there was too many links on the page. This result is not directly linked with QBLS.

It shows that the integration of the tools is successful and that they do not present incoherencies or difficulties regarding the question-based task.

8.4.4. Acceptability

Globally, students positively evaluated QBLS. But in the perspective of further using this system in particular and on-line access course systems in general, further acceptability issues must be considered. Two indicators seem significant of the acceptance of the system: the satisfaction of students and the access to the system outside lab hours, out of any supervision.

8.4.4.1. Measuring student satisfaction

In the two separate experiments, users positively rated their satisfaction with the navigation system (4.3/5 and 3.9/5). They also found it easy to navigate (3.9/5 in both cases), which is very encouraging considering this paradigm was new to them.

None of the student was reluctant to use a computer to access the course. This result is largely influenced by the fact that we addressed a course for computer engineering students. However, most students are now acquainted with web browsing. As the system does not require any technical manipulation on the learner side and runs in most browsers, we claim that acceptability of such tools should be good for a large portion of university students. Usage figures concerning a third of them that accessed the system freely are also in favor of this conclusion.

The question of the paper version of the course remains unsolved. It has been requested by several students who feel safer with a non-volatile content. Students declared in the questionnaires that they needed paper versions even though they did not really use it during the labs (from the observers' point of view). In the questionnaires, the affirmation addressing their ability to perform the lab without paper support received a score of 3.9 out of 5.

8.4.4.2. Analysis of "free" access

Most of the students declared that they could perform lab sessions alone at home (3.8/5). This result can be checked against the monitored activity times. In QBLS-2 where the tool could be accessed freely 24/7, we recorded a lot of activity (40% of the total) outside lab hours (see figure 104). Days of the week are not represented, but on figure 102 some of the activity "peaks" happen on weekends. We interpret this result as a strong indicator of the acceptance of the system by learners.

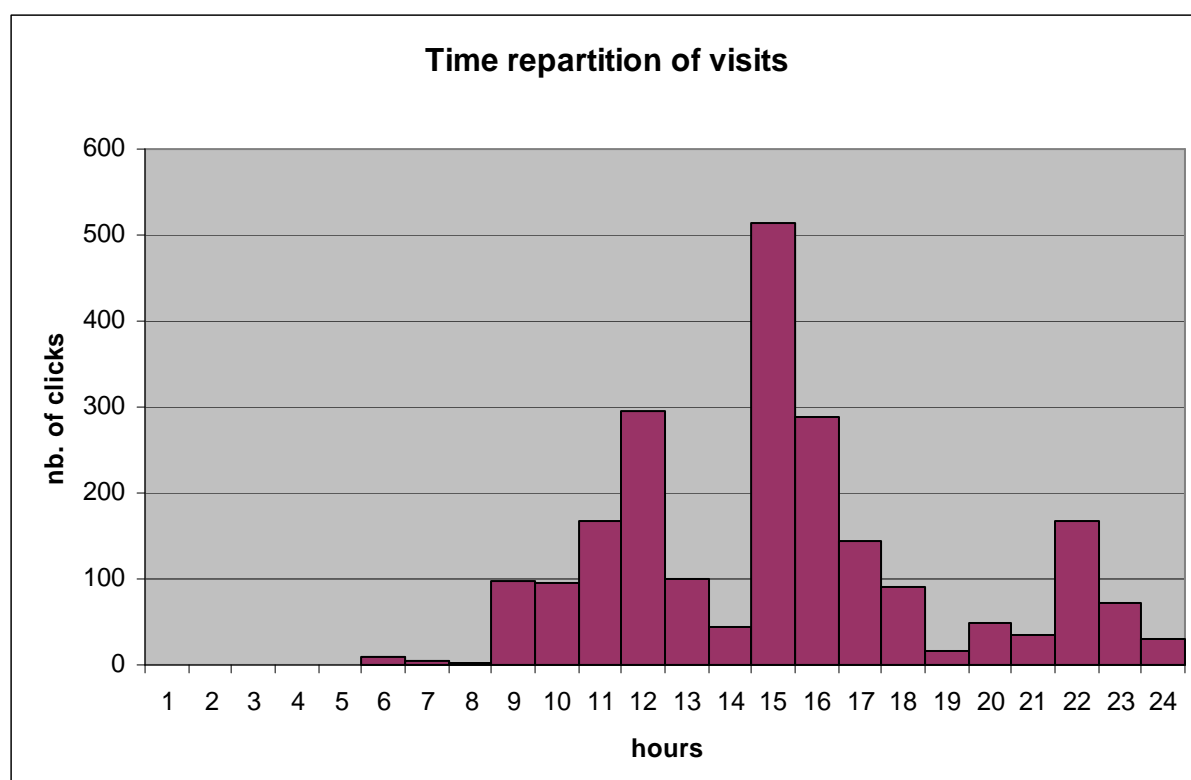


Figure 104 – Distribution of system accesses during the day.

Some students wished they had access the system from home but did not have internet connection, revealing that this figure could have been even higher, and will certainly increase in the future.

The level of acceptance among students is high, because *a noticeable part of them freely chose to use the system over books and paper handouts.*

8.5. Synthesis

The above evaluation helps us in determining the success of the deployment of QBLS in real teaching situations. It may also validate several aspects of the proposals we made in this thesis. In particular, the design choices made at the beginning can be reviewed according to these evaluation results. In particular, we focus on the chosen pedagogical scenario, the idea of reusing a course, the choice of the navigation paradigm and the global interaction model.

8.5.1. Pedagogical considerations

As explained at the beginning, the goal of the evaluation is not to monitor the progress in learners' knowledge but to focus on student's ability to perform the assigned task. Nevertheless, the final goal of the system cannot be completely ignored. We collected the following remarks about the potential efficiency for learning:

- First, the objective of the lab is well understood by the students. This is confirmed by both direct observation and analysis of the answer sheets. With QBLS, the activity resembles usual lab activity where students have to perform exercises with the help of their teacher. It seems important for the acceptance of a system that the task remains close to actual practices.
- Students faced a difficulty to get a general vision of the course. They expressed this concern in the questionnaires. However, they acknowledge that the system helps them to answer the specific questions. It appears that the lecture is still important for the overall comprehension. This is also linked to the specificity of the question-based approach, which

emphasizes practical knowledge over theoretical visions. This quote from a student interrogated at the end of the course (direct observation) « *when we have to answer problems we tend to only look for the necessary information to answer the question and not bother about the general subject of the course* ».

➤ Concerning the QBLS strategy, the motivation brought by the strategy proved effective. Moreover, the course material seemed to be more used than the previous years when only paper version was available.

➤ From the “performance” point of view of the students at exams, we notice no correlation between the use of the system and the ability to perform well at tests or exams. The cognitive profiles of learners that are helped by the system do not match the categorization brought by exam results. We expected this result and it confirms that the impact of the system itself is better evaluated through the techniques presented above than through pure performance tests.

8.5.2. Course reuse

The major paradigm underlying the design of QBLS is the reuse of existing material to save time and effort for teachers when setting up on-line courses. Two aspects must be considered here:

- How students cope with a reused material.
- How teachers feel about this reuse. Practical aspects of reuse have already been presented in chapter 5.

8.5.2.1. Student’ point of view

Few problems or incoherencies in the content have been observed during the experimentations. This is yet a partial result because the first course re-used was modified by the teacher in QBLS-1. Only the second one was reused from the web without modification of its content. The large size and common type of the course reused in QBLS-2, is at least a significant example of reusing existing material.

The question of distributing a linear paper version of the course document remains. Paper allows personal annotations for students, which cannot be performed here. In addition, students declare that they prefer paper for reviewing their course. This raises several difficulties linked to teaching practice that are not yet answered for the reuse of existing material: Which document should be printed?, What form should it take?, etc.

8.5.2.2. Teacher’s point of view

From teacher’s opinion, the system revealed easy to use during the labs. The new role it implies for him/her was quite disorientating at first, especially for an experienced teacher. However, assurance grew during the labs. It is clear from the observation that the familiarization phase takes longer for the teacher than for the student. For the teacher the hardest issue seems the acceptance of the reused material. When trying to reuse a course, even one from colleagues, teachers always want to adapt it to their personal viewpoint.

We claim that this is necessary when the course is used as a support to oral teaching during a lecture. For on-line courses used during labs there is no objective reason why a course could not be reused, and that is what QBLS-2 demonstrates. However even in QBLS-2 we faced this issue as the teacher in charge of the course started to enrich the questions so much with course contents that the usage frequency of QBLS remained stable whereas we expected a progressive increase of activity. There is no other explanation to this than a lack of acceptance of an “outside” course material by the teacher.

8.5.3. Navigation model

The navigation model seems well adapted to the chosen task. Learners rapidly understood the navigation features in the interface and did not express navigation difficulties. They also find the system easy to navigate and useful to find quickly the necessary information. This tends to validate the proposed navigation model.

The question of making visible the pedagogical type of the resource remains an open issue, as we could not determine exactly on which criteria students base their decision when choosing to visit a resource. More specific tests would be necessary for this. Nevertheless, we proved that the automatic ordering of the resources was important because of the “left to right” pattern observed in the visits.

8.5.4. Global setting

The efficiency of a learning setting depends on many parameters. In the learning situation we address, not all of them are linked to the system. “External” criteria should be taken into account (group interaction, previous knowledge, personal motivation, etc.). We just mention the following points as indicating the key aspects we have been able to evaluate:

- The “free” access to the system emphasizes that even when attending face-to-face courses in a regular location, distant on-line learning, especially from home, is now playing an important role in academic settings. For evaluation, it means that on-line teaching situations have to consider both environments (classroom and home).
- The system was differently appreciated by the teachers for whom it represents a far higher change of practices than for the students. We effectively observed a change in their role, becoming “guides”, peering over students shoulder to help them with their task rather than facing them. In the case of QBLS-2, where the system was introduced by one of the four teachers supervising the labs, we observed the difficulties of putting such new practice in place: having some teachers printing the questionnaires (thus removing any possibility for on-line access to the course), enriching questions with course content, etc.
- The QBLS system fits the chosen strategy and the overall setting (labs in small groups, etc.). The integration of different tools (wiki+QBLS) working over a distant connection proved successful to support the given learning task and we consider it as a major evaluation result supporting our proposal.

8.6. Evaluation framework and generalization of the results

The evaluation framework presented here introduces several aspects common to the evaluation of digital educational software in general. The different techniques of evaluation complete one another and scaffold the conclusions that emerge from the interpretation of those results.

8.6.1. Specific methods for specific conclusions

The experience acquired while evaluating the QBLS system allows us to precise the degree of generality of the proposed methods. In a nutshell, the evaluation techniques (questionnaires, observations, recording, etc.) are generic, but the evaluation of the learning task is very contextual. Especially the conclusions about the efficiency of the system regarding the task are closely linked to the studied implementation.

The evaluated task is different for each system, each environment, etc. Depending on the evaluation goal, different techniques might be adapted or emphasized. For example, most of the affirmations in the questionnaires are specific to the problems met with a specific interface. They target specific students, in a specific teaching and learning environment. Some

questions could be reused (like the SUS part) others need to be changed according to the context. The same remark applies to other evaluation means like observation and answer sheets.

Generic aspects are found in the way the questionnaire is written, and in most of the measures made on the activity traces (paths, times, etc.), but no generic conclusion could be drawn from one application that would be valid for all the others. Even if evaluation means are similar, the interpretation of the results can only be contextual. Evaluation takes into account large varieties of information that are not captured by direct evaluation tools but rather belong to the context. For example, students answer questionnaires considering the global setting, it might be difficult to separate the role of the different tools in such evaluation

The different techniques illustrated could be applied in many situations. As a general result we emphasize the need to use *several evaluation tools* to cross-validate the different interpretations.

The few conclusions that may possess a generic value should be understood as “best practices”. We present in the following two paragraphs the evaluations results that may have a generic implication. The distinction is made between their relevance for either e-learning or the semantic web domain.

8.6.2. Best practices for evaluating e-learning systems

The evaluation in e-learning is a critical issue. Serious evaluation is required to secure the major investments necessary to develop new tools. This encompasses many types of tools used in a learning situation (tutoring systems, adaptive hypermedia, course consultation systems, etc.).

We propose the following guidelines for such evaluation:

- First, direct observation by teachers is the first source of feedback. In distance learning such direct observation cannot be performed, but solutions must be found to place human observers together with the students at some point in the evaluation.
- Questionnaires are the most straight forward and directly informative tool about the user's feelings and reaction. Our evaluation shows the interest of a questionnaire for collecting student feedback that would be difficult to collect otherwise. It also stresses the importance of being able to quantify the answers to exploit them with statistical tools.
- Interaction traces between learners and the system are interesting to confirm interpretations from the observations, but alone they are not sufficient to interpret user activity. They can be collected automatically at run-time and thus can give instant feedback, whereas questionnaires and learners productions can only be analyzed afterwards.
- During an evaluation, it is important to collect all the documents produced in particular the documents produced by learners (answer sheets, digital productions, drafts, etc.). Other « contextual » documents (photos, environment plan, agenda, etc.) will help the interpretation afterwards. If this could reveal a very rich source of information the amount of work needed to exploit them is important compared to the quick feedback obtained with other methods.

The practices described above can be organized on two dimensions (depending on their capacity to deliver immediate result and on the level of involvement from the teacher.

Table 13 – Characteristics of different evaluation methods

	Instant feedback	Delayed feedback
Strong Teacher involvement	Observation	Learner's production
Weak teacher involvement	Log analysis	Questionnaire Log analysis

We do not intend to present here an exhaustive study of existing evaluation techniques. Many others have been proposed. This is just a panel of important ones, which brought interesting feedback in the context of the QBLS experiments and for evaluation of on-line course access systems in general.

Such techniques particularly focus on student activity. In the scenario of the QBLS experiments (1 and 2), a teacher also takes part in the lab. He/she is a specific user of the system and the system must be evaluated differently in this scope. Only one teacher or potentially a small team interacts with QBLS and observations are limited. On the other hand, it is possible for an external evaluator to interact directly with them. Other techniques must then be put in place (interviews, video, etc.).

8.6.3. Recommendations for evaluating semantic web systems

The general evaluation presented evaluates the semantic aspects at a high level. The technical aspects linked to the semantic web are hidden to the end-user and thus do not really play an important role in the global evaluation. The range of applications that can rely on the semantic web paradigm is wide. It is too ambitious to propose guidelines for evaluating them, especially as we stressed the importance of context.

Then, we limit our proposal to the evaluation of two important aspects: the use of generic ontologies and the more low level technical aspects

8.6.3.1. Users and ontologies

A major difference introduced by semantic web systems, compared for example to classical information systems, is the integration of generic ontologies to support their functionalities. The understanding of the user with regard to the concepts introduced by the ontology must be evaluated, and a specific attention must be paid to ontology-related tasks.

- First, the amount of disorientation brought by ontology-related tasks, if any, must be evaluated.
- The accordance of semantic representations with users' conceptualization (for example the adequacy of the labels) should be checked.
- Semantic web will remain a new paradigm for some time, and there is a good probability that users will feel disorientated anyway. Thus, it is important to evaluate the learning curve of the system, which will condition its further use.

In the course of our evaluation, we observed that existing "ergonomic" evaluation heavily relies on existing practices, and tend to favor well-known designs over new ones. An evaluation framework based on this idea will largely reject semantic innovations. Again, it emphasizes the necessity to evaluate familiarization phases.

8.6.3.2. Technical evaluation

Scalability

An important technical issue with semantic web applications is scalability. In an application like QBLS, large amounts of semantic annotations are already manipulated because of the

number of log statements generated. In a broad perspective, even larger sets should be considered (several courses, hundreds of students, etc.). Precise figures about response times and critical thresholds are necessary indicators. For QBLS, response time reached its acceptable limits when taking into account the full history of all students. The number of users remained comparatively quite low and thus was not a problem.

Expressivity

With regard to a semantic web based infrastructure, we report here on the lack of expressivity of the SPARQL language to extract information automatically from semantic logs for such analysis. Operations like counting, and aggregating numerical values are difficult to express in this language. Due to this limitation, the construction of interfaces for dynamic feedback based on the display of SPARQL results cannot be straight forward (as presented in chapter 6 for constructing user interfaces). The analysis of activity logs then offers an interesting use case where large amount of data can be easily collected and that requires a rich specific expressivity.

Other technical aspects should be hidden by the use of standards. Thus, ensuring interoperability would also simplify technical evaluation through standard procedures, unified benchmarks, etc...

8.7. Conclusion

The contribution of QBLS to the field of e-learning goes beyond the demonstration of semantic web technologies. It is a complete system that was used in real teaching situations. In this chapter, we present the evaluation framework applied to evaluate the real world applications of the QBLS system. Several evaluation techniques are introduced. Their results are presented along with the interpretations we made. We also draw some directions for evaluation in a more generic perspective.

The types of labs performed with QBLS take place in “classical” settings and we said that students are used to such labs. It is especially true for French high education, but the practices might be different in other countries. Therefore, we did not evaluate this last factor of cultural background. This factor is mentioned in ergonomic studies (Nielsen, 94) and certainly plays an important role in such evaluation.

After presenting the different proposals contained in our framework in the previous chapters 5, 6 and 7, the evaluation proved that they converged toward a useful, usable and acceptable system. In the next chapter, we will sum up our scientific contribution in a global method for setting up semantic on-line courses.

9. GENERAL METHODOLOGICAL GUIDE 199

<i>9.1. From annotation to log analysis: a complete method</i>	200
9.1.1. Method overview.....	200
9.1.2. Final outcome.....	203
<i>9.2. Positioning</i>	203
9.2.1. KM methodologies.....	203
9.2.2. A Semantic Web approach.....	204
9.2.3. Learning paradigm.....	204
<i>9.3. The ASPL integration</i>	205
9.3.1. Exploitation of existing documents.....	205
9.3.2. Semantization of contents.....	205
9.3.3. Exploitation through semantic services.....	206
9.3.4. Analysis.....	210
9.3.5. Perspectives.....	210

9. GENERAL METHODOLOGICAL GUIDE

In this last chapter, we expose a general method summing up the different contributions of this thesis. The proposition aims at defining a reproducible process for the creation and the exploitation of on-line courses, based on the reuse of existing material. The method integrates the different solutions exposed in the previous chapters through a unified approach. It is intended to serve as a guide for future research, but it also targets “pedagogical engineers” who directly address teachers’ needs.

The proposed method is composed of four main steps. The realization of the steps has been described in the different chapters. References to the relevant chapters will be made when appropriate.

The presentation is organized in three main sections:

- First, the proposed method is exposed and the role of each step is described.
- The positioning and originality of the method are presented. We discuss its specificities as a knowledge engineering method. We highlight the central role played by the semantic web and we review the method in a pedagogical perspective.
- Finally, to illustrate its realization, we present an example where the method was used, along with the QBLS implementation, as part of a semantic navigation platform for learning developed in the context of the Knowledge Web European network of excellence.

9.1. From annotation to log analysis: a complete method

9.1.1. Method overview

We propose a four steps process to create an on-line course consultation system from a reused document. The process itself does not commit to any specific implementation or pedagogical scenario, but it aims at providing helpful navigation for learners, with the assumption that formalized knowledge about the course will provide such support. In particular, the method targets the creation of conceptual navigation spaces.

Compared to existing approaches, the method targets the reuse of courses, typically found on the web. If the course is created from scratch, different paradigms may be more relevant.

Our method is based on the following four basic steps:

- Selection of the original document
- Semantization of the document
- Usage of the content through conceptual navigation by the learners
- Learner activity analysis of the use of the resources in a specific system

These four steps are chronologically ordered in a cycle presented in figure 105. The positions of the steps on the figure stress that each step is in fact related to the others through its previous and following neighbors in the cycle, but also through the central role played by formalized knowledge.

Each step acts like a piece of a bigger puzzle: it fits with other steps to build a coherent “picture” of course reuse, but it also has connections with potential “external” pieces that complete the puzzle of on-line teaching.

Black arrows linking the central “knowledge area” to each step indicates that the influence between them goes both ways:

- Each step influences the knowledge base through the selection of selecting specific models, by augmenting the base, etc.
- The formalized knowledge influences all the steps, from the original selection of the material, which might be ontology-guided, to the analysis of activity report relying on the conceptual course structure.

Arrows in dotted lines indicate the possible cycles in the process:

- Between the semantization phase and the actual usage of the system by learners, the teacher may go back and forth to validate the effect of his/her annotations.
- The final feedback may influence the selection of the next course and thus induce a global cycle for the process.

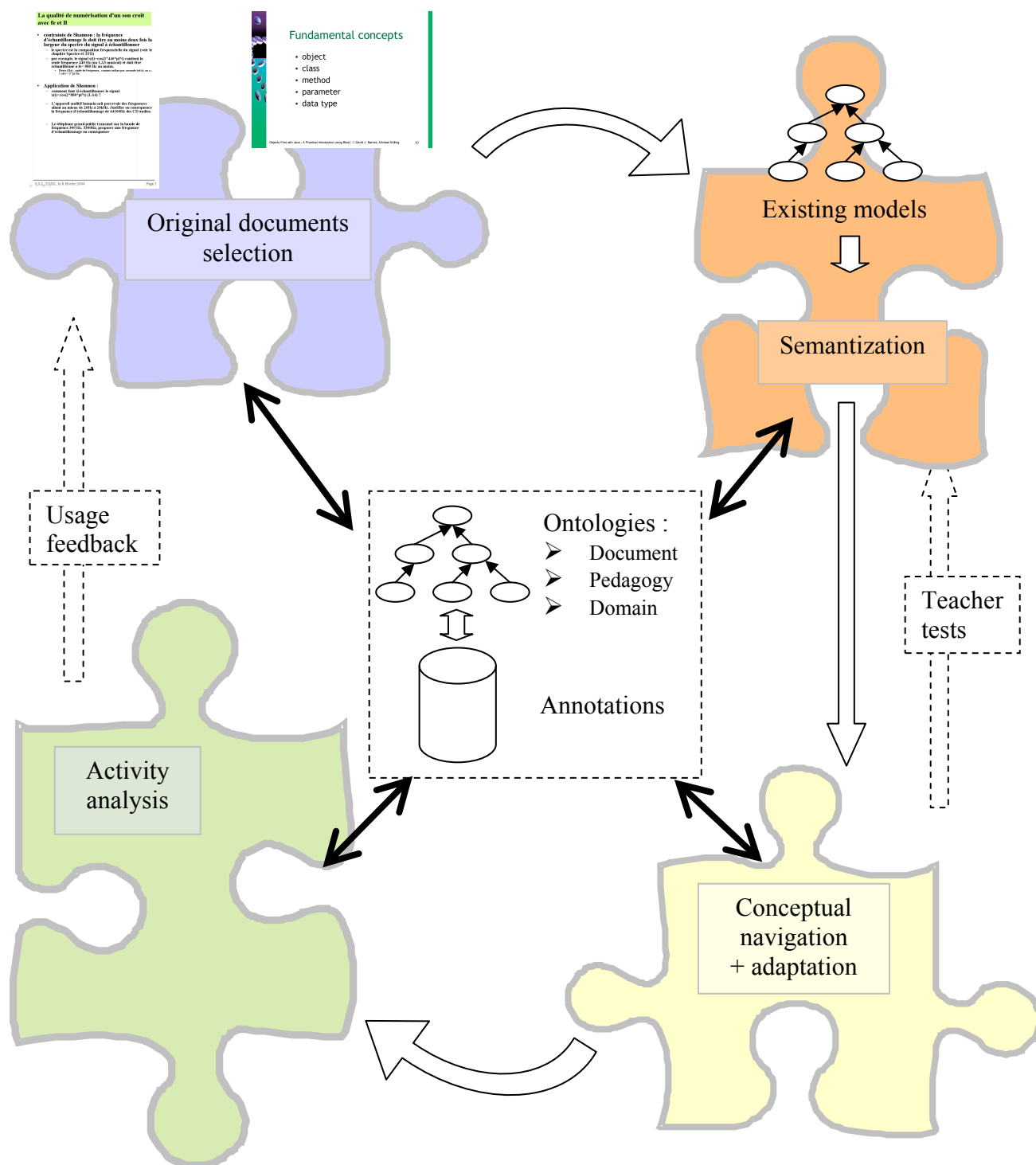


Figure 105 – Method overview for creating and exploiting on-line courses from reused documents

9.1.1.1. Selection

The selection of one or several resources is the first milestone in the process. The criteria for choosing a specific resource are multiple. Some of them are more or less objective; others are more subjective and depend on the teacher in charge of the choice. A number of criteria are summarized in table 14.

Table 14 – Criteria for choosing a resource to reuse

Objective criteria	Subjective criteria
<ul style="list-style-type: none"> ➤ Suitability of the material for web display ➤ Domain covered (potentially determined with an ontology). ➤ Size of the material with regard to teaching time ➤ Suitability of the material for semantic annotation 	<ul style="list-style-type: none"> ➤ Author's popularity ➤ Confidence in the source (organization, contact, etc.) ➤ Perceived strategy

Learning object repositories, such as Ariadne (Duval et al., 01), are potential places to find resources to reuse. The teacher may also choose to reuse his/her own course. In this case, the need for edition of the content may reveal higher.

9.1.1.2. Semantization

The semantization process has been described chapter 5. This step plays a major role in the creation of annotations and in the choice/evolution of the ontologies formalizing the different models. The main idea behind this step is to make explicit, as much as possible, the vision of the teacher on the material. Modification should focus on the layout and the presentation aspects, only minor modification of the content itself should occur (see chapter 5).

The different ontologies that have been identified as relevant to express this vision concern:

- the document model,
- the pedagogical strategy,
- the domain model.

All three models possess specific and generic aspects discussed in detail in chapter 5. It is not necessary to introduce to the teacher such high-level concepts such as ontology or reasoning, but their simple grounding in styles and ordering of resources should be enough for the teacher to perform the semantization process.

9.1.1.3. Usage

In an on-line course, learners interact with the course through a browsing interface. Depending on the strategy, the teacher's choices, etc., the navigation must be organized differently using the available knowledge. The proposed method does not commit to any particular implementation or design process. We experienced that the development of the interface was quite straightforward for a reasonably skilled technical staff once the knowledge models and navigation paradigms were fixed. The existing implementation of QBLS, presented in chapter 6, is easily reusable and customizable and may be exploited as a common basis for the applications of this method.

Teacher's exploration of the interface may lead to adapt/modify the knowledge base (models or annotations) in order to correct inappropriate navigation directions. This induces an inner cycle in the proposed process (see figure 105). Direct visual feedback during the usage step by learners may also trigger such cycle.

9.1.1.4. Usage analysis

The last step consists in analyzing the activity traces generated by learners while using the interface. Directions for such an analysis involving formalized knowledge about the course were presented chapter 7.

This step is not compulsory but it takes advantage of the semantization effort to produce original feedback for the teacher. He/she can evaluate how students visited the material. It also gives indications on students' profiles and their navigation strategies.

Such feedback may lead to improvements in the future choices of the material or in the way it is exploited. For example, the pedagogical strategy might be adapted according to those results.

9.1.2. Final outcome

This modular method is composed of several innovative contributions. Separately, each step can be integrated in other processes for annotation, conceptual navigation, activity analysis etc. However, they are especially interesting through the global approach proposed here, where all the steps contribute to and take advantage of a central “knowledge area”.

We sum up the expected benefits of this method as follows:

- It offers a complete process from creation of the annotations to the evaluation of learner activity. Still each step is independent. Their interoperability is enhanced by a standard based approach.
- Formalized knowledge is placed at the heart of the system through a knowledge-based approach. It allows teachers to construct a conceptual navigation and a conceptual analysis of learners' behaviors.
- It offers a teacher oriented resource reuse approach that takes into account feasibility and tries to minimize the time spent in the annotation phase.

The various tools developed in the context of QBLS are generic. The annotation processor, the log analyzer, etc. rely on semantic web standards and thus are reusable in different contexts. This emphasizes the modularity and potential applicability of the proposal.

9.2. Positioning

9.2.1. KM methodologies

The method borrows principles from existing knowledge management methods presented in (Dieng-Kuntz, 04). However, our proposal focuses on supporting a learning goal. This differs from building memories in general, even if the method leads to the creation of a knowledge intensive system and share many aspects with existing methods, like relying on experts (here teachers) to acquire knowledge.

A major difference comes from the nature of the users. Obviously, learners are not experts of the domain, but neither can they be placed on the same level as users of a knowledge base like in a corporate memory scenario (Dieng-Kuntz, 04). For example a typical practical scenario of knowledge management in a corporate memory context could be: “user X needs information about the concept of inheritance in Java”. By exploiting the memory, the user may find existing resources (projects, reports, persons etc.) that will help him/her find the information he/she needs.

In our learning situation (typically in the first years of university teaching), this classical scenario does not apply: learners ignore the existence of the concept and they cannot formulate such request. They do not only need to find information but first to learn about it.

As a result, the proposed method acquires knowledge from a specific type of experts: teachers. It also focuses on course documents as the major communication channel towards

learners and lectures, debates, etc. are not considered here. In the end, it offers *a knowledge engineering approach dedicated to the design of learning systems based on existing courses*.

9.2.2. A Semantic Web approach

The high-level description of the method presented in this chapter does not commit to any particular formalism. However, it implicitly targets the use of semantic web standards as a pivotal formalism.

The architecture proposed for QBLS relies on RDFS and a few predicates from OWL. Using these formalisms, the different existing tools can exchange knowledge between them. For example, ontologies can be developed in Protégé, integrated in the QBLS annotation tool and used in the Corese search engine. The central role played by formalized knowledge in the method can only be achieved using standard languages. The semantic web in particular justifies the connections of the steps with external components.

In terms of expressivity, important limitations are imposed by teachers who are not experts in knowledge engineering. They can only manipulate simple paradigms like concept vocabularies, and subject relations that find a direct expression in the interface. In this scope, the expressivity required in the semantic language is easily obtained with RDFS and predicates of OWL like transitive properties, inverse, etc.

9.2.3. Learning paradigm

9.2.3.1. Integration

From a pedagogical point of view, the proposed method only addresses a fraction of the process of setting up a complete learning activity. It focuses on the exploitation of course content by learners while performing a “learning activity”. As illustrated by the “puzzle” pieces in figure 105, the method is intended to articulate with other processes aiming at the design of a complete course (this includes scheduling, organizing activities, linking with other courses, etc.). In the context of a distance-learning course handled by a LMS (Learning Management System) like Moodle for example, the QBLS system would represent a “block”, along with the forum, calendar, or testing modules.

9.2.3.2. Motivation

The major motivation for using annotated resources is to propose an “enriched” navigation in the course. In this sense, this proposal is in-line with existing works on adaptive and intelligent systems. However, the specific focus we put on reuse of existing material is quite original.

Another motivation that distinguishes our method is that we want to propose a complete solution for navigation: starting from raw material to usage feedback. In particular, feedback is not considered as a “one time” evaluation means but it is part of the global method and can be obtained each time the system is used.

9.2.3.3. Application

The applications designed through the proposed process might produce rather classical systems regarding adaptive and conceptual navigation. Because the method addresses the difficulties of real teachers in expressing knowledge about their course, it cannot reach all the ambitious goals of some adaptation engines.

The originality of the method does not lie in its outcome or in the artifact produced but in the expression and exploitation of the knowledge involved in every step of the process.

The type of knowledge expressed potentially limits the interactions with the user to a simple request/answer dialogue. More complex interactions, like push mechanisms that provide dynamic feedback and not just “on-request” answers, certainly require different methods to gather the necessary knowledge to build such interaction.

9.3. The ASPL integration

As a final point to this summary, we present the integration of the QBLS system into the Advanced Semantic Platform for Learning (ASPL) developed in the context of the Knowledge Web network of excellence.

This work demonstrates the generic applicability of the method. It also emphasizes the potential various exploitations of the produced knowledge as soon as it is integrated in a semantic web environment. The presentation follows the four steps described in the method.

9.3.1. Exploitation of existing documents

In the context of the Knowledge Web network of excellence, a repository of learning resources, called REASE (<http://rease.semanticweb.org/ubp> at the time writing), has been set up to collect digital documents used for courses related to the semantic web.

Project members developed a platform, called ASPL, to help students enrolled in semantic web studies. The platform aims at integrating several tools that will guide students in their learning task using ontologies. In particular, the system can be used to exploit the content of the REASE repository.

The goal of QBLS in this system is to offer on demand information about available resources to other front-end interfaces that will enforce different pedagogical strategies.

9.3.2. Semantization of contents

On the REASE repository, we selected five documents available in Power Point format. Those course concerned different aspect of the semantic web domain: knowledge modeling, ontology engineering, OWL, rules and ontologies, and methods and tools for corporate memories. All of them had different authors but they share a common format and basic layout (title+content). The semantization process can be applied to those resources that are submitted in editable formats (here PowerPoint slideshows)

Following the process described chapter 5, we annotated those resources with regard to a hierarchy of concepts about the semantic web domain. The hierarchy results from a discussion among experts of the domain. It was originally designed to annotate the resources of REASE where contributors assign one several concepts to their course document in a “LOM-like” style. This reproduces a very classical use of hierarchies in LORs.

The hierarchy can be easily formalized in SKOS and integrated in our framework through this meta-model. The pedagogical ontology from (Ullrich, 04) was reused again to annotate each slide.

The usage scenario was not clearly defined. As a project member, I endorsed the role of a teacher and annotated the resources (at this stage of the evolution of QBLS we could not ask teachers to do it themselves). In this situation, not all the slides in the resources could be annotated with domain or pedagogical concepts. We already stressed the importance of contextualization when annotating and we experienced it again when annotating those documents.

We applied the extraction process to each resource. Both extracted content and annotations were stored on specific instance of QBLS. As several original documents are introduced in this application, we developed a multi-resource repository, allowing a “teacher” to submit several annotated files to the same instance of QBLS.

9.3.3. Exploitation through semantic services

We deployed two different services to exploit the resources and the extracted annotations: a consultation service called from the front-end of ASPL formed by the Magpie tool, and a querying service used by an intermediate integrator tool.

9.3.3.1. Integration with Magpie

The interface developed for QBLS-2 was reused with little modifications and coupled with the Magpie tool (Dzbor et al., 05) (see 3.3.4.5). A screenshot of this tool is presented in figure 106. This tool is a plug-in for browsers. It allows users to highlight in the current web page the terms identified as concepts from several reference ontologies. For example, in figure 106, terms highlighted in orange are concepts of the semantic web domain, taken from the semantic web topic hierarchy mentioned above.



Figure 106 – The magpie tool highlighting semantic web concepts on the ACACIA homepage

By right clicking on a term, Magpie proposes a list of associated services. This list depends on the selected concept and the ontology defining it. In figure 106 the last service in the list proposes to query a QBLS server, in this case about the concept of “Knowledge Management”. It would retrieve existing resources (slides) from the repository regarding this concept.

The domain ontology is shared between the Magpie tool and the QBLS tool. The recognition of terms in Magpie is completely automatic and only relies on the labels defined in the

ontology. Through the application of the semantization process, this ontology has been enriched to annotate the slides in QBLS. This enrichment now benefits the Magpie tool.

By selecting the QBLS service, the following window pops-up (see figure 107). It proposes two resources retrieved about the selected concept of “Knowledge Management”. The tabbed presentation reuses the design developed for QBLS-2.

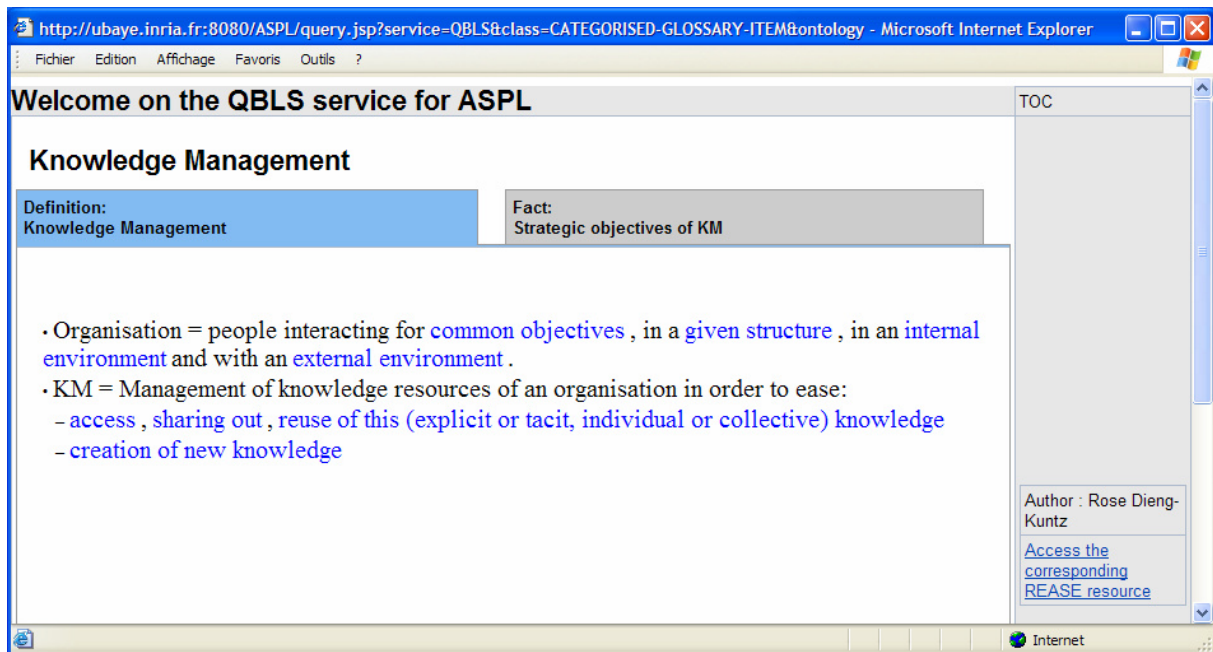


Figure 107 – QBLS interface for ASPL

The author of the resource is indicated on the bottom right for respect of authorship. A link to the original document, containing this slide on the REASE repository, is also proposed. Both indicators are dynamically extracted from an RDF base provided by the repository.

9.3.3.2. Integration at the web service level

Given the architecture of the QBLS system, the result returned by the semantic search engine can be transformed to any output format through an XSL transformation (see 6.5).

This is used in ASPL to combine the output of QBLS with other services, in particular a glossary service proposed by another project partner. For each request on a concept, a page is constructed by aggregating the results from both the glossary service and the QBLS service. The result of this aggregation is presented in figure 108. It combines a glossary entry, provided by a first service and a list of resources proposed by QBLS. Both contributions are related to the concept of “WEB” defined in the common semantic web topic hierarchy. In this example, the glossary provides the definition and QBLS provides resources of other types that will help in understanding and learning the selected concept.

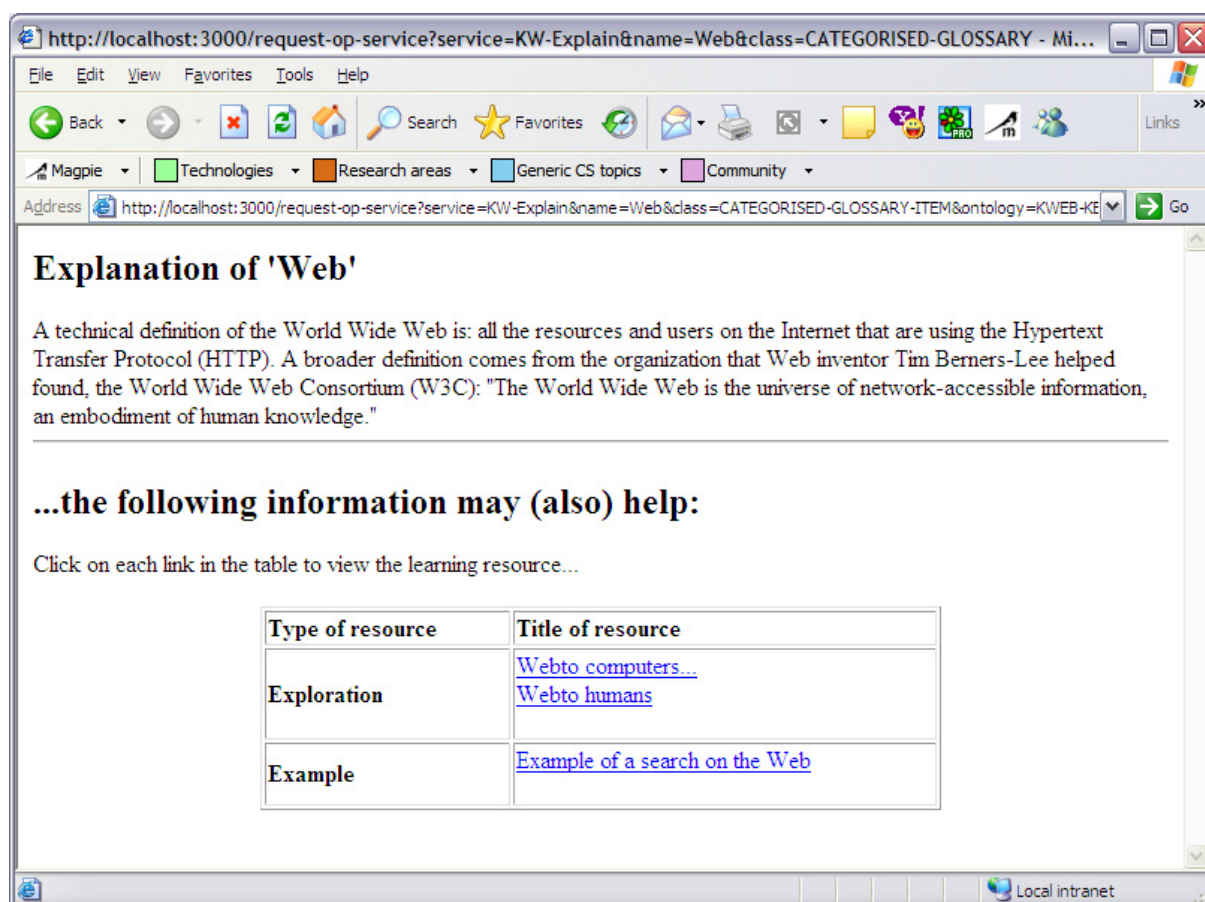


Figure 108 – Combination of two services informing about the concept of “web”

This example illustrates how different sources of information can be combined to build coherent interfaces when they rely on the same ontologies. Redundancy of the information is avoided by querying QBLS only for other resources than definition.

9.3.3.3. Overview

Figure 109 gives an overview of the part of ASPL that involves QBLS. Learners access the proposed services through Magpie. From this entry point, they can call different services that appear in separate windows (QBLS interface and Composite service). The domain ontology is shared between the different tools (Magpie, QBLS, REASE, Glossary service). For learners the underlying complexity of the system is hidden and navigation takes place only in standard browser interfaces.

The ASPL platform operates in the context of web navigation guided by semantic web concepts. The pedagogical strategy of QBLS, based on initial questions, can be applied. For example, Magpie can be used when browsing questionnaires. The QBLS service could then direct the users to existing resources that help them answering the questions. However, the framework is generic and other strategies might be supported.

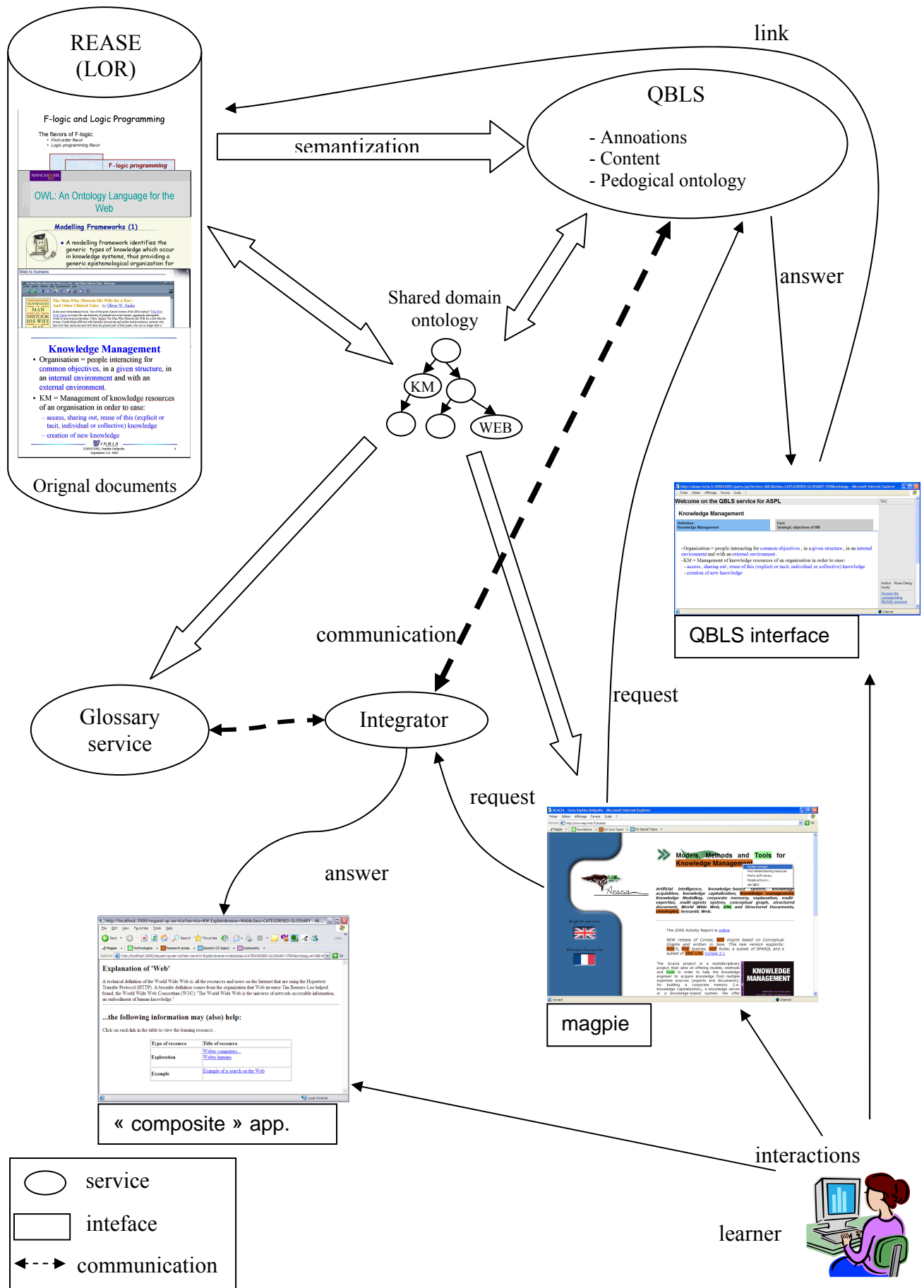


Figure 109 – Overview of the ASPL platform

Several resources from different sources may be presented to the learner and redundancy is difficult to avoid in this case. For example if the glossary entry also provides examples in the content provided, the clarity may be affected. That is an unavoidable drawback of reusing resources without modifying them. Globally, this type of system targets more advanced learners than those of QBLS-1 or QBLS-2. The later are less affected by information redundancy because they only target the reuse of a single document.

9.3.4. Analysis

This example of deployment of QBLS in a different context from what is presented in the previous chapters, illustrates the generality of the proposed method. However, the implementation of the last step of the cycle remains problematic. A distributed environment like ASPL, combines several distant services. It raises the problem of tracking user activity across the various tools. Following the proposed method would consist in defining a common log model and share the traces recorded by each service. Up to now this has not been done, and raises a number of issues about privacy, scalability, etc. Still, (Brooks and McCalla, 06) made some proposals in that direction and this difficult problem should now be addressed by frameworks like ASPL.

9.3.5. Perspectives

9.3.5.1. A link towards original resources

Annotating the material represents a cost. When dealing with many different resources this cost increases proportionally. In this context, we propose the following solution: Only the most important slides of a presentation are annotated.

In QBLS for ASPL, when displaying a resource, a link is proposed towards the original document. Then, if a learner really needs some specific information, he/she can always access the original material to find it. The “preview” given by the few annotated slides gives a good hint about the relevance of the original resource for the current task while greatly reducing the annotator’s work.

9.3.5.2. Building repositories of annotated resources

Such “preview mechanism” can also be used to take advantage of resources already annotated by teachers using QBLS. Even if the context in which the slides have been annotated is different, the system can direct students to “previews” and lead them to the original resource. Such resources could be compiled in a repository offering both the original document and the annotated versions. It would also reveal an interesting source of material for teachers.

If the same resource is annotated several times by different teachers for different contexts, the corresponding domain vocabularies can be merged or aligned to propose navigation based on several viewpoints. Compared to approaches like Ariadne (Duval et al., 01), setting up a repository of such annotated resources would present a much higher added value than actual solution relying on the LOM metadata that provide very little pedagogical and domain related information.

We conclude that the proposed method was successfully applied in this context of a complex distributed platform. Several resources have been reused and exploited. The distributed aspect raises issues for activity analysis. This problem illustrates the state of advancement of semantic web applications that must now try to share and synchronize knowledge on a larger scale.

10. CONCLUSION

The research contributions, presented in this thesis, focus on the application of semantic web technologies to the domain of e-learning. Issues about annotation, semantic web technologies and tracking have been answered. They lead to the proposition of a global method for reusing course documents and offering their content on-line through an enhanced navigation based on explicit knowledge.

This work is in large part based on the example of the QBLS system developed during this work. The QBLS implementation allowed us to conduct real world applications of the proposed method and highlights the potential of semantic web technologies for such e-learning systems.

10.1. *Scientific conclusions*

10.1.1. Semantics for learning

In this work, we presented how formalized knowledge can be used to support navigation in an existing course. Many proposals already exist around this idea, the originality of our proposal stands in the “practical” vision we adopt:

- We target the reuse of existing courses to reduce teacher’s authoring effort.
- We focus on realistic features, keeping feasibility constraints in mind.
- We propose a annotation process relying on popular existing tools like MS-Word or OpenOffice writer.

Given these hypotheses, we defend the following theses:

Knowledge expressed by teachers must be limited

During the three years of this project, we gained the conviction that actual solutions proposed for “intelligent” or “advanced” learning (using computers) are not sustainable because of the incredible effort required from the teacher. Such efforts, called “authoring” in the e-learning domain, combine creation of content with the expression and formalization of knowledge.

Authoring is often described as limited because of cost, but we experienced that it was often limited because of feasibility. For example, expressing relationships like prerequisites does not make sense in some situations.

Models must be limited to the teacher’s ability to instantiate them. Thus, we propose to reduce and focus the teacher’s task on the expression of limited models in direct relation with the proposed material.

Knowledge can be interestingly used only if it is contextual to the learning situation

We identified the models that could possess a universal value (Pedagogical ontology, SKOS, etc.) and stressed the fact that other models, like the domain in particular, mostly depend on the material.

By analyzing the domain knowledge necessary to offer a conceptual navigation, we show that such knowledge is mostly contextual and could not easily be reused. This characteristic is tied to the specificity of learning because Learning cannot be assimilated to a simple knowledge transfer where a single model of the knowledge to learn could be described. In learning systems the domain models carries a pedagogical vision.

The satisfaction obtained with learners when using QBLS is really encouraging for this type of applications, which up to now have been largely disregarded by practitioners as non-practical solutions. We did propose a satisfactory solution that was economically acceptable.

Knowledge based analysis of learner activity brings valuable pedagogical information

We proposed a navigation model that appeared efficient for both navigating the course and better interpreting the student's activity on such a course. These results are still preliminary but we propose to use them as a first practical mean to describe patterns, either behavioral ones, or temporal ones.

10.1.2. Semantic web approach

Semantic web technologies are efficient to implement knowledge based learning systems.

The first claim we make concerns the interest of integrating standards and tools of the semantic web. Deploying such technology requires some engineering efforts because of its novelty, but considering the lightweight, and customizable application developed (see 6.1), we consider it as a big step forward for implementation issues in intelligent learning systems.

The implementation of the QBLS is mostly generic. Not only does the system reuse in large part an existing generic tool with the Corese semantic search engine, but as most communications between the components the application rely on W3C standards (SPARQL, RDF, OWL) the implementation we produced is largely reusable. For example the ontologies are completely independent of the implementation and they could be exchanged without modifications.

The level of inference necessary in the context of learning systems is relatively low compared to the expressivity of OWL for example. An important result for further development of the semantic web standards concerns the expressivity of query languages rather than modeling languages. It illustrates a move towards the operationalization of those models.

Knowledge interoperability based on standards can be useful and effective

Secondly, we defend the position of the semantic web as a working solution for some aspects of knowledge interoperability. By reusing existing ontologies (domain, pedagogy) in OWL and using them directly in QBLS, we experienced the feasibility of this approach with actual tools.

Also, the ASPL platform also demonstrates the potential interoperability and collaboration between different systems when they are sharing a common representation of knowledge.

Learning systems would benefit of an “educational semantic web” approach

The proposed architecture and application scenario envisioned for the QBLS system proved useful by extending the “corporate semantic web” approach (Dieng-Kuntz, 04) to the domain of e-learning.

We showed that this approach of managing knowledge in the learning organization as we could do with knowledge inside a corporate organization brings interesting results but also possess its own specificities. Our vision of an “educational semantic web” is not a theoretical view where knowledge would be seamlessly exchanged at the scale of the web, but a practical approach, where knowledge is defined inside a small community and benefits from the standard languages and tools to manipulate it inside a well defined learning environment.

10.2. Results

Two major artifacts result from this work: a methodological guide and an implementation. Building on this experience, we fuel the debate on the concept of Learning Object.

10.2.1. A methodological guide

Creating on-line courses has been the subject of numerous books, essays or guides trying to answer the growing interrogations of practitioners facing this new task. The method we propose tackles this problem through *reusing material by annotating it*.

The role played by the different aspects of knowledge is emphasized and the practical approach we favored from the beginning gives a real user-oriented approach, which we hope, will benefit not only future research but also practices of tomorrow.

10.2.2. A showcase implementation

The QBLS system does not only demonstrate the efficiency of the semantic web approach in term of development. It is also a modular application which components can be re-used. Especially the annotation support module, for standard editors like MS-Word or Open Office, is a valuable tool with huge potentialities and little competitors.

10.2.3. Burying the Learning Object paradigm?

The learning object approach should now largely be rejected as a mere engineer's dream, far from actual practice and real human (limited) possibilities. Still many efforts pursuing its realization now provide solid building blocks on which our proposal relies. For example, the possibility to perform search on repositories for resources of large size is a potential good initiator of the method we introduced. The LO philosophy of cutting down resources into smaller bits is also a major inspiration and did prove useful. Of course, sharing and exchanging those resources must be limited to well defined contexts.

The vision of placing the material at the center of the communication between the instructor and the student should not be underestimated. If research is now turning to other aspects of the course "preparation" like learning designs, we should not forget the importance of the material, which remains the main communication interface for learning with computers.

10.3. Perspectives

10.3.1. Integration of annotation tools

In a close future, the convergence towards XML formalism will reach its paroxysm. In browsers, XHTML will be enriched with SVG or mathML content allowing teachers to put much richer content on line. Classical document editors will also rely on XML for storage. It will be possible to access any content for powerful manipulations, some of them already proposed with the latest XSLT 2.0 language.

However, the extraction of knowledge from such sources will always require important engineering efforts. The right goal to pursue seems rather to integrate more and more annotation tools in standard editors. Semantic annotation should become entirely part of the edition/authoring process of a document. It is already the case in authoring tools for e-learning but popular generic editors are for the moment mostly concerned with simple metadata, like authorship or dates.

We believe that such functionalities will find their place in future of "semantic-aware" applications replacing MS-Power Point, MS-Word, etc. Such tools will undeniably play a key role for integrating more formalized knowledge in our learning applications in the next few years.

10.3.2. Towards complex adaptive engines on the semantic web

Adaptation or personalization of web sites is becoming increasingly popular on the web in general. With the spread of the idea of Web 2.0 where web sites are much more reactive, we

are leaving the current scheme of query/answer dialog for much richer interactions. Coupling dynamic pages with adaptive engines will allow designers to produce much more complex interaction paradigms.

This has not been achieved yet on a large extent because of technical difficulties to run reactive clients on end-user's computers. With the integration of those functionalities in standard browsers, this will undoubtedly know a large diffusion. The architecture we propose on the server side, relying on a semantic middleware, can perfectly support such interaction. The production of services for dynamic interfaces should then increase dramatically.

In this "all web" world, issues like scalability and performance will become crucial for semantic web back-ends as the number of queries will increase along with the size of the knowledge bases.

10.3.3. The future of e-learning

We conclude this work by a personal word about the future of learning technologies in general. We have seen already that the definition of "e-learning" was quite unclear and that it now encompasses most learning situations because computers and digital media have percolated through mostly every day-to-day situation.

The biggest question, which largely outpasses the scope of this work is the cognitive implications of the use of such systems for learning. Impact on recall and memorisation are still largely unknown. Even more we are not sure we still need to learn as we used to. If information can be accessed quickly from anywhere at any time, does our memory needs to be trained for the same purpose? This is a question future practice will definitely need to address.

Finally, we do not see the further development of such systems being the fact of volunteer teachers and visionary practitioners, but rather as a consequence of "pervasive" computers. Our role as researchers is not to push for the integration of more digital support in education, but to prepare and propose solutions that will progressively become standard practice as we, learners, pursue our route through the digital age.

REFERENCES

10.4. Publications

(Abel et al., 04) Abel M.-H., Benayache A., Lenne D., Moulin C., Barry C., Chaput B., Ontology-based Organizational Memory for e-learning. In *Journal of Educational Technology & Society*, Special Issue on Ontologies and the Semantic Web for E-learning, vol. 7, issue 4, pp. 98-111, 2004.

(Allert, 04) Allert H., Coherent Social Systems for Learning: An Approach for Contextualized and Community-Centred Metadata. In *Journal of interactive Media in Education*, 2004.

(Altenhofen and Schaper, 02) Altenhofen M., Schaper J., Flexible Instructional Strategies for E-learning. In *Proceedings of 35th International Conference on System Sciences, HICSS-35 2002*, Big Island, HI, USA, Jan 7-10, 2002.

(Amanti and Ounis, 00) Amati G. and Ounis I., Conceptual Graphs and First Order Logic. In *The Computer Journal*, vol. 43, issue 1, pp. 1-12, 2000.

(Anderson, 01) Anderson L.W., A taxonomy of learning, teaching, and assessment: A revision of Bloom's taxonomy of educational objectives. In Krathwohl, D.R. eds., New York, Longman, 2001.

(Armani and Rocci, 03) Armani J., Rocci A., Conceptual maps in e-learning. How map based interfaces help the contextualization of information and the structuring of knowledge. In *Information Design Journal + Document Design*, vol. 11 (2 /3), pp. 171-184, 2003.

(Aroyo and Dicheva, 04) Aroyo L. and Dicheva D., The New Challenges for E-learning: The Educational Semantic Web. In *Journal of Educational Technology & Society*, Special Issue on Ontologies and the Semantic Web for E-learning, vol. 7, issue 4, pp. 59-69, 2004.

(Ausubel et al., 78) Ausubel D., Novak J., Hanesian H., Educational Psychology: A Cognitive View (2nd Ed.). Holt, Rinehart & Winston (eds.), New York, 1978.

(Azouaou et al., 03) Azouaou F., Desmoulins C., Mille D., Formalismes pour une mémoire de formation à base d'annotations. In Desmoulins C., Marquet P., Bouhineau D. (eds.), *Environnements Informatiques pour l'Apprentissage Humain 2003*, pp. 43-54, Strasbourg, 2003.

(Azouaou and Desmoulins, 05) Azouaou F. and Desmoulins C., Semantic Annotation for the Teacher: Models for a Computerized Memory Tool. In *Proceedings of International Workshop on Applications of Semantic Web Technologies for E-Learning, AIED'2005*, Amsterdam, July 2005.

(Baader and Nutt, 2002) Baader F. and Nutt W., Basic Description Logics. In Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F. (eds.), *the Description Logic Handbook*, Cambridge University Press, pp. 47-100, 2002.

(Baget and Mugnier, 02) Baget J.F. and Mugnier M.L., Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints. In *Journal of Artificial Intelligence Research*, vol.16, pp. 425-465, 2002.

(Barfurth et al., 94) Barfurth M.A., Basque, J., Chomienne M., Winer L.R., Les instruments de collecte de données de recherche qualitative dans des environnements pédagogiques informatisés. In Bordeleau, P. (eds.), *Apprendre dans des environnements pédagogiques informatisés*, Éditions Logiques, pp. 485-548, 1994.

- (Barrows, 86) Barrows H. S., A taxonomy of Problem Based Learning methods. In *Medical Education*, pp. 481-486, 1986.
- (Berendt and Bernstein, 01) Berendt B. and Bernstein E., Visualizing individual differences in Web navigation: STRATDYN, a tool for analyzing navigation patterns. In *Behavior Research Methods, Instruments & Computers*, vol. 2, issue 33, pp. 243-257, 2001.
- (Berners Lee, 94) Berners Lee Tim, Plenary talk by Tim BL at WWW 94: Overview. www.w3.org/Talks/WWW94Tim/
- (Berners-Lee et al., 01) Berners-Lee T., Hendler J., Lassila O., The Semantic Web. *Scientific American*, May 2001.
- (Berners-Lee, 01) Berners-Lee T. Conceptual Graphs and the Semantic Web, W3C Note, <http://www.w3.org/DesignIssues/CG.html>, 2001.
- (Biezunski et al., 99) ISO/IEC 13250 Topic Maps Information Technology -- Document Description and Markup Languages. Biezunski M., Bryan M., Newcomb S.R. (eds.), 3 Dec 1999.
- (Blanc, 01) Blanc D., La boîte à outils du formateur. Editions d'organisation Eyrolles, troisième édition, 2001.
- (Bloom, 56) Bloom B.S., Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. Longmans (eds.), New York, 1956.
- (Bouzeghoub et al., 05) Bouzeghoub A., Defude B., Duitama J.F., Lecocq C., Un modèle de description sémantique des ressources pédagogiques basé sur une ontologie de domaine. In *Revue STICEF*, Volume 12, 2005,
- (Brailsford et al., 02) Brailsford T.J.; Stewart C.D., Zakaria M.R., Moore A., Autonavagation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment. In *Proceedings of 11th International World Wide Web Conference*, Honolulu, Hawaii, 7-11 May 2002.
- (Brase and Nejdl, 03) Brase J. and Nejdl W., Ontologies for eLearning. *Handbook on Ontologies*, Heidelberg, Springer, 2003.
- (Brooke, 96) Brooke J., SUS: a "quick and dirty" usability scale. In Jordan P.W., Thomas B., Weerdmeester B.A., McClelland A.L.(eds.), *Usability Evaluation in Industry*, London: Taylor and Francis, 1996.
- (Brooks and McCalla, 06) Brooks C., McCalla G., Towards flexible learning object metadata. In *International Journal of Continuing Engineering and Lifelong Learning*, vol. 16, Nov. 1/2, pp. 50-63, 2006.
- (Brown et al., 05) Brown E., Cristea A., Stewart C., Brailsford T., Patterns in Authoring of Adaptive Educational Hypermedia: A Taxonomy of Learning Styles. In *Journal of Educational Technology & Society*, Special Issue, vol. 8, issue 3, pp. 77-90, 2005.
- (Brusilovsky, 99) Brusilovsky P., Adaptive and intelligent technologies for Web-based education. In *KI - Kunstliche Intelligenz*, vol. 13, pp. 19-25, 1999.
- (Brusilovsky and Vassileva, 03) Brusilovsky P. and Vassileva J., Course sequencing techniques for large-scale web-based education. In *International Journal of Continuing Engineering Education and Lifelong Learning*, vol. 13, pp. 75-94, 2003.

- (Brusilovsky, 03) Brusilovsky P., Developing adaptive educational hypermedia systems: From design models to authoring tools. In Murray T., Blessing S. Ainsworth S. (eds.), *Authoring Tools for Advanced Technology Learning Environment*, Dordrecht, pp. 377-409, 2003.
- (de Bra et al., 03) De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Santic T., Smits D., Stash N., AHA! The Adaptive Hypermedia Architecture. In *Proceedings of the ACM Hypertext Conference*, Nottingham, UK, August 2003.
- (Breuker and Bredeweg, 99) Breuker J. and Bredeweg B., Ontological Modelling for Designing Educational Systems. In *Proceedings of workshop on Ontologies for Intelligent Educational Systems*, AI-ED 99, Le Mans, France, July 18-19 1999.
- (Buffa et al., 05) Buffa M., Dehors S., Faron-Zucker C., Sander P., Towards a Corporate Semantic Web Approach in Designing Learning Systems: Review of the TRIAL SOLUTION Project. In *International Workshop on Applications of Semantic Web Technologies for E-Learning*, AIED'2005, Amsterdam, July 2005.
- (Burton, 82) Burton R.R., Diagnosing bugs in a simple procedural skill. In Sleeman D.H., Brown J.S. (eds.), *Intelligent Tutoring Systems*, Academic Press, London, 1982.
- (Catteau et al., 06) Catteau O., Vidal P., Broisin J., A Generic Representation Allowing for Expression of Learning Object and Metadata Lifecycle. In *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies*, ICALT06, pp. 30-33, Kerkrade, The Netherlands, July 2006.
- (Christiansen and Anderson, 04) Christiansen J. and Anderson, T. Feasibility of Course Development Based on Learning Objects: Research Analysis of Three Case Studies. In *International Journal of Instructional Technology and Distance Learning*, vol. 1, no. 3, 2004.
- (Colluci, 05) Colucci S., Di Noia T., Di Sciascio E., Donini F.M., and Ragone A., Semantic-based automated composition of distributed learning objects for personalized e-learning. In *Proceedings of the 2nd European Semantic Web Conference*, Grece, June 2005.
- (Conlan, 02) Conlan O., Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2002)*, Malaga, Spain, June 2002.
- (Conklin, 87) Conklin J., Hypertext: an introduction and survey. In *IEEE Computer*, pp. 17-47, September 1987.
- (Corby et al., 00) Corby O., Dieng R., Hébert C., A Conceptual Graph Model for W3C Resource Description Framework. In *Proceedings of the International Conference on Conceptual Structures (ICCS'2000)*, LNCS, Springer Verlag, 2000.
- (Corby et al., 04) Corby O., Dieng-Kuntz R., Faron-Zucker C., Querying the Semantic Web with the CORESE search engine. In Lopez de Mantaras R. and Saitta L. (eds.), *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'2004)*, subconference PAIS'2004, IOS Press, pp. 705-709, Valencia, 22-27, August 2004.
- (Corby et al., 05) Corby O., Dieng-Kuntz R., Faron-Zucker C., Gandon. F., Searching the Semantic Web: Approximate Query Processing based on Ontologies. In *IEEE Intelligent Systems Journal*, 2005.
- (Crampes et al., 00) Crampes M., Ranwez S., Plantier M., Ontology-Supported and Ontology-Driven Conceptual Navigation on the World Wide Web. In *Proceedings of HyperText 2000*, San Antonio, Texas USA, ACM Press, pp. 191-199, June 2000.

(Cristea, 04) Cristea A., What can the Semantic Web do for Adaptive Educational Hypermedia? In *Journal of Educational Technology & Society*, Special Issue on Ontologies and the Semantic Web for E-learning, vol. 7, issue 4, pp. 40-58, 2004.

(Dagger, 02) Dagger D., Wade V., Conlan O., Towards a Standards-based Approach to e-Learning Personalization using Reusable Learning Objects. In Richards G. (eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pp. 210-217, Chesapeake, VA, 2002.

(Dalgarno, 96) Dalgarno B., Constructivist computer assisted learning: theory and techniques. In A. Christie, P. James and B. Vaughan (eds.), *Making new connections, Proceedings of the thirteenth annual conference of the Australasian Society for Computers in Learning in Tertiary Education*, University of South Australia, Adelaide, pp. 143-154, 1996.

(Dalgarno, 98) Dalgarno B., Choosing learner activities for specific learning outcomes: A tool for constructivist computer assisted learning design. In C. McBeath and R. Atkinson (eds.), *Planning for Progress, Partnership and Profit, in Proceedings of EdTech'98*, Perth, 1998.

(Dalziel, 03) Dalziel J., Implementing learning design: the learning activity management system (LAMS). In *Proceedings of ASCILITE*, 2003.

(Dehors et al., 05) Dehors S., Faron-Zucker C., Giboin A., Stromboni J.P., Semi-automated Semantic Annotation of Learning Resources by Identifying Layout Features. In *Proceedings of International Workshop on Applications of Semantic Web Technologies for E-Learning*, AIED'2005, Amsterdam, July 2005.

(Delozanne et al, 07) Delozanne É., Le Calvez F., Merceron A., Labat J.-M. A Structured set of Design Patterns for Learners' Assessment. In *Journal of Interactive Learning Research (JILR)*, Special Issue Usage Analysis in Learning Systems: Existing Approaches and Scientific Issues, vol. 18, no. 2, 2007.

(Desmoulins and GrandBastien, 02) Desmoulins C. and Grandbastien M., Des ontologies pour la conception de manuels de formation à partir de documents techniques. In *Sciences et techniques éducatives*, vol. 9, no. 3-4, pp. 291-340, Hermès, 2002.

(Devedzic, 04) Devedzic, V. Education and The Semantic Web. In *International Journal of Artificial Intelligence in Education*, vol. 14, pp. 39-65, 2004.

(Dieng et al., 99) Dieng R., Corby O., Giboin A. and Ribière M., Methods and tools for corporate knowledge management. In *International Journal of Human-Computer Studies*, vol. 51, issue 3, pp. 567-598, 1999.

(Dieng-Kuntz et al., 05) Dieng-Kuntz R., Corby O., Gandon F., Giboin A., Golebiowska J., Matta N. and Ribière, M., Méthodes et outils pour la gestion des connaissances : Une approche pluridisciplinaire du Knowledge Management, Paris: Dunod, 2005.

(Dieng-Kuntz, 04) Dieng-Kuntz R., Capitalisation des Connaissances via un Web Sémantique d'Entreprise. In I. Boughzala et J-L. Ermine (eds.), *Chapitre 12 de Management des Connaissances en Entreprise*, Hermès, 2004.

(Dolog et al., 04) Dolog P., Henze N., Nejdl W., Sintek M., Personalization in Distributed e-Learning Environments. In *Proceedings of the 13th International World Wide Web Conference*, May 2004.

(Dicheva et al., 05) Dicheva D., Sosnovsky S., Gavrilova T., Brusilovsky P., Ontological Web Portal for Educational Ontologies. In *Proceedings of International Workshop on Applications of Semantic Web Technologies for e-Learning*, AIED'2005, Amsterdam, The Netherlands, July 2005.

- (Dicheva and Dichev, 06) Dicheva D. and Dichev C., TM4L: Creating and Browsing Educational Topic Maps. In *British Journal of Educational Technology - BJET*, vol. 37, issue 3, pp. 391-404, 2006.
- (Drucker, 00) Drucker P., Need to Know - Integrating e-Learning with High Velocity Value Chains. Delphi Group White Paper. www.delphigroup.com, 2000.
- (Duval et al., 01) Duval E., Forte E., Cardinaels K., Verhoeven B., Van Durm R., Hendrikx K., Wentland-Forte M., Ebel N., Macowicz M., Warkentyne K., Haenni F., The ARIADNE Knowledge Pool System. In *Communications of the ACM* 44, no. 5, pp. 73-78, 2001.
- (Duval et al., 02) Duval E., Hodgins W., Sutton S., Weibel S., Metadata Principles and Practicalities. In *D-Lib Magazine*, vol. 8, no. 4, April 2002.
- (Dzbor et al., 05) Dzbor M., Motta E., Stutt A., Achieving Higher-level Learning Through Adaptable Semantic Web Applications. In *International Journal of Knowledge and Learning*, vol. 1, no. 1/2, Inderscience Enterprises Ltd., UK, 2005.
- (Falmagne et al., 00) Falmagne J.C., Cosyn E., Doignon J.P., Thiery N., The assessment of knowledge in theory and in practice http://www.highedmath.aleks.com/about/Science_Behind_ALEKS.pdf.
- (Farrell et al., 04) Farrell R., Liburd S., Thomas J., Dynamic assembly of learning objects. In *Proceedings of the 13th international conference on World Wide Web - Alternate Track Papers & Posters*, WWW 2004, New York, USA, May 17-20, 2004.
- (Friesen, 04) Friesen N., The International Learning Object Metadata Survey. In *The International Review of Research in Open and Distance Learning*, ISSN 1492-3831, 2004.
- (Gandon, 01) Gandon F. Experience in Ontology Engineering for a Multi-Agents Corporate Memory System, in *Proceedings of Workshop "Ontologies and Information Sharing"*, Seventeenth International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, pp. 119-122, August 4th - 10th 2001.
- (Gandon et al., 05) Gandon F., Corby O., Giboin A., Gronnier N., Guigard C., Graph-based inferences in a Semantic Web Server for the Cartography of Competencies in a Telecom Valley, ISWC, Lecture Notes in Computer Science, Galway, 2005.
- (Garshol, 03) Garshol L.M., Living with topic maps and RDF. In *Proceedings of XML Europe 2003*, London, UK, 5-8 May 2003.
- (Giboin et al., 02) Giboin A., Gandon F., Corby O., Dieng R., User Assessment of Ontology-based Tools: A Step Towards Systemizing the Scenario Approach. In *Proceedings of EON'2002: Evaluation of Ontology-based Tools*, OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2002), pp. 63-73, 2002.
- (Gilly et al., 99) Gilly M., Roux J.-P., Trognon A., *Apprendre dans l'Interaction : analyse des médiations sémiotiques*. Presses Universitaires de Nancy et Publications de l'Université de Provence, 1999.
- (Goble et al., 01) Goble C., Bechhofer S., Carr L., De Roure D., Hall W., Conceptual Open Hypermedia = The Semantic Web? In *Proceeding of SemWeb2001, 2nd International Workshop on the Semantic Web*, Hong Kong, May 2001.
- (Gruber, 95) Gruber T.R., Toward principles for the design of ontologies used for knowledge sharing. In *International Journal of Human Computer Studies*, pp. 907-928, 1995.

(Hanisch et al., 06) Hanisch F., Muckenhaupt M., Kurfess F., Straßer W. The AHES Taxonomy: Extending Adaptive Hypermedia to Software Components. In *Proceedings of AH 2006*, pp. 342-345, 2006.

(He, 02) He S., Hong H., Patel A., Adaptivity in Problem-based Learning : use of granularity. In *Proceeding of International Conference on Computer in Education*, Los Alamitos, USA, IEEE Computer Society, pp. 711-712, 2002.

(Henze and Nejdl, 01) Henze N. and Nejdl W., Adaptation in open corpus hypermedia. In *International Journal of Artificial Intelligence in Education, Special Issue on Adaptive and Intelligent Web-Based Systems*, vol. 12, 2001.

(Henze, 05) Henze N., Personal Readers: Personalized Learning Object Readers for the Semantic Web. In *Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED'05)*, Amsterdam, The Netherlands, 18th - 22nd July, 2005.

(Heiwy and Ducateau, 03) Heiwy V. and Ducateau C-F., Un modèle de ressources pédagogiques pour la FOAD. In *Colloque Environnements informatiques pour l'apprentissage humain (EIAH'2003)*, Strasbourg, France, 15-17 avril 2003.

(Horst, 05) Horst H. J., Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. In *Proceedings of ISWC 2005*, LNCS 3729, pp. 668-684, Galway, Ireland, November 6-10, 2005.

(Jacquiot et al., 06) Jacquiot C., Bourda Y., Popineau F., Delteil A., Reynaud C., GLAM : A Generic Layered Adaptation Model for Adaptive Hypermedia Systems. In V. Wade, H. Ashman and B. Smyth., *Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2006)*, Springer, LNCS 4018, pp. 131-140, June, 2006.

(Koper, 01) Koper R., Modeling units of study from a pedagogical perspective - The pedagogical metamodel behind EML <http://www.eml.ou.nl/introduction/articles.htm>.

(Kolb, 76) Kolb D. A., The Learning Style Inventory: Technical Manual, McBer., Boston, MA, 1976.

(Kunze et al., 02a) Kunze T., Brase J., Nejdl W., Editing Learning Object Metadata: Schema Driven Input of RDF Metadata with the OLR3-Editor, in *Proceedings of the ECAI 2002 Workshop on Semantic Authoring, Annotation & Knowledge Markup*, Lyon, July 22-26 2002.

(Kunze et al, 02b) Kunze T., Richter C., Allert H., Dhraief H., Nejdl W., Instructional Models and Scenarios for an Open Learning Repository - Instructional Design and Metadata. In G. Richards (eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pp. 75-82, Chesapeake, VA, 2002.

(Leidig, 01) Leidig T., L3 towards an open learning environment. In *Journal of Educational Resources in Computing*, March 2001.

(Likert, 32) Likert R., "A Technique for the Measurement of Attitudes" *Archives of Psychology* 140, pp. 55, 1932.

(Marshall et al., 03) Marshall B., Zhang Y., Chen H., Lally A. M., Shen R., Fox E., Cassel L. N., Convergence of Knowledge Management and E-Learning: the GetSmart Experience. In *Proceedings of Joint Conference on Digital Libraries*, Houston, USA, 2003.

(Maurer and Sapeer, 01) Maurer H. and Sapper M., E-learning has to be seen as part of general knowledge management. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 2001.

- (Mazza and Dimitrova, 04) Mazza R. and Dimitrova V. Visualizing Student Tracking Data to Support Instructors in Web-Based Distance Education. In *Proceedings of 13th International World Wide Web Conference*, pp. 154-161, New York, USA, May 17-20 2004.
- (McEneaney, 99) McEneaney J.E., Visualizing and Assessing Navigation in Hypertext. In *Proceedings of Hypertext '99*, pp. 61-70, 1999.
- (Merrill, 99) Merrill M. D., Instructional Transaction Theory (ITT): Instructional Design Based on Knowledge Objects. In C.M. Reigeluth (eds.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory*, vol. II., chap. 17, pp. 397-424, Mahwah, NJ: Lawrence Erlbaum Associates, 1999.
- (Michau and Ploix, 03) Michau F. and Ploix S., Proposition de carte organisationnelle : décrire le rôle d'un environnement informatique au sein d'un dispositif pédagogique. In *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2003)*, Strasbourg, France, 15-17 avril 2003.
- (Millard et al., 06) Millard D., Tao F., Doody K., Woukeu A., Davis H., The Knowledge Life Cycle for e-learning. In *International Journal of Continuing Engineering Education and Lifelong Learning: Special Issue on Application of Semantic Web Technologies in E-learning*, vol. 16(1/2), pp. 110-121, 2006.
- (Mitrovic, 03) Mitrovic A., An Intelligent SQL Tutor on the Web. In *International Journal of Artificial Intelligence in Education*, IOS Press, vol. 13, pp. 171-195, 2003.
- (Mizoguchi et al., 97) Mizoguchi R., Ikeda M., Sinita K., Roles of Shared Ontology, in *Proceedings of AI-ED Research - Intelligence, Conceptualization, Standardization, and Reusability, AIED-97*, Kobé, Japan, 1997.
- (Mizoguchi and Bourdeau, 00) Mizoguchi R. and Bourdeau J., Using Ontological Engineering to Overcome AI-ED Problems. In *International Journal of Artificial Intelligence in Education*, vol. 11, no. 2, pp. 107-121, 2000.
- (Motik and Sattler, 06) Motik B. and Sattler U., Practical DL Reasoning over Large ABoxes with KAON2. Submitted for publication.
- (Mugnier and Leclere, 05) Mugnier M.L. and Leclère M., On Querying Simple Conceptual Graphs with Negation. Research Report LIRMM, RR05-051 (41 pages), July 2005
- (Murray, 03) Murray T., MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. In *International Journal of Artificial Intelligence and Education*, vol. 13, 2003.
- (Naeve, 01) Naeve A., The Concept Browser - a new form of Knowledge Management Tool. In *Proceedings of the 2nd European Web-based Learning Environments Conference (WBLE 2001)*, Lund, Sweden, October 24-26, 2001.
- (Nanard and Nanard, 03) Nanard M. and Nanard J., EadGen, un environnement ouvert de production en ligne basé sur XML et XSL-T, In *Proceedings of TICE 2002*, p. 157-166, Lyon, November 13-15, 2002.
- (Nejdl et al., 02) Nejdl W., Wolf B., Qu C., Decker S., Sintek M., Naeve A., Nilsson M., Palmer M., Risch T., EDUTELLA: A P2P Networking Infrastructure Based on RDF. In *Proceedings of 11th International World Wide Web Conference (WWW2002)*, Hawaii, USA, May 2005.
- (Nielsen, 94) Nielsen J., Heuristic evaluation. In Nielsen J., and Mack R.L. (eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, 1994.

(Niles and Piese, 01) Niles I., and Pease A., Towards a Standard Upper Ontology. In C. Welty and B. Smith (eds.), *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine, October 17-19, 2001.

(Nilsson et al., 03) Nilsson M., Palmér M., Brase J., The LOM RDF binding - principles and implementation. In *Proceedings of 3rd Annual Ariadne Conference*, Katholieke Universiteit Leuven, Belgium, November 2003.

(Nogry et al., 04) Nogry S., Jean-Daubias S., Ollagnier-Beldame M., Evaluation des EIAH, une nécessaire diversité des méthodes. In *TICE 2004*, France, 2004.

(Novak and Canas, 06) Novak J. D. and Cañas A.J., The Theory Underlying Concept Maps and How to Construct Them. Technical Report IHMC CmapTools 2006-01, Florida Institute for Human and Machine Cognition, 2006.

(Noy, 06) Noy N., Representing Classes As Property Values on the Semantic Web, *W3C Working Group Note*, <http://www.w3.org/TR/swbp-classes-as-values/>

(Otter and Johnson, 00) Otter, M. and Johnson, H., Lost in Hyperspace: Metric and Mental Models. *Interacting with Computer*, vol. 13, pp. 1-40, 2000.

(Palmer et al., 01) Palmér M., Naeve A., Nilsson M., E-learning in the Semantic Age. In *Proceedings of the 2nd European Web-based Learning Environments Conference (WBLE 2001)*, Lund, Sweden, October 24-26, 2001.

(Paquette et al., 97) Paquette G., Crevier F., Aubin C., Méthode d'ingénierie d'un système d'apprentissage (MISA). In *Revue Informations In Cognito*, vol. 8, 1997.

(Prié, 00) Prié Y., Sur la piste de l'indexation conceptuelle de documents. Une approche par l'annotation. In *Document Numérique, numéro spécial "L'indexation"*, vol. 4, no. 162, pp. 11-35, 2000.

(Quemada and Simon, 03) Quemada J. and Simon B., A Use-Case Based Model for Learning Resources in Educational Mediators. In *Educational Technology & Society*, vol. 6, issue 4, pp. 149-163, 2003.

(Quemada et al., 04) Quemada J., Huecas J., Miguel G., Salvachua T., Fernandez J., Simon B., Maillet B., Law K., EducaNext: A Framework for Sharing Live Educational Resources with Isabel. In *Proceedings of the 13th International World Wide Web Conference*, USA, May 2004.

(Reigeluth, 99) Reigeluth C.M., Instructional-Design Theories and Models, Volume II: A New Paradigm of Instructional Theory. Mahwah, NJ: Lawrence Erlbaum Assoc., 1999.

(Rigaux and Spyratos, 03) Rigaux P. and Spyratos N., Metadata Management and Learning Object Composition in a Self e-Learning Network. In *Proceedings of 1st Franco-Japanese Workshop on Information Search, Integration and Personalization*, Sapporo, Japan, June 30-July 2, 2003.

(Rodríguez and Verdejo, 04) Rodríguez-Artacho M. and Verdejo Maíllo M. F., Modeling Educational Content: The Cognitive Approach of the PALO Language. *Educational Technology & Society*, vol. 7, issue 3, pp. 124-137, 2004.

(Salmon, 01) Salmon G., It's teaching Jim, but not as we know it! The role of university teachers in the knowledge media age. In *Keynote presentation to ALT-C 2001: Changing learning environments*, Edinburgh, September 11-13, 2001.

(Savin-Baden, 00) Savin-Baden M., Problem-based Learning in Higher Education; untold stories, Buckingham, Open University Press/SRHE, 2000.

- (Scapin and Bastien, 97) Scapin D.L., Bastien J.M.C., Ergonomic criteria for evaluating the ergonomic quality of interactive systems. In *Behavior & Information Technology*, vol. 17, no. 4/5, p. 220-231, 1997.
- (Schneider et al., 03) Schneider D., Class B., Frété C., Girardin F., Lombard F., Morand S., Conception et implémentation de scénarios pédagogiques riches avec des portails communautaires. In *Communication au Second Colloque de Guéret "Les communautés virtuelles éducatives"*, Guéret, 4-6 juin 2003.
- (Schmidt and Winterhalter, 04) Schmidt A., Winterhalter C., User Context Aware Delivery of E-Learning Material: Approach and Architecture. In *Journal of Universal Computer Science (JUCS)*, vol. 10, no. 1, January 2004.
- (Schreiber et al., 99) Schreiber G., Akkermans H., Anjewierden A., de Hoog R., Shadbolt N., Van de Velde W., Wielinga B., Knowledge Engineering and Management: The CommonKADS Methodology, 1999.
- (Simon and Quemada, 02) Simon B. and Quemada J., A Critical Reflection of Metadata Standards based on Usage Scenarios. In *Proceedings der GMW Tagung*, Basel, Switzerland, September, 2002.
- (Sowa, 84) Sowa J.F., Conceptual Structures: Information Processing in Mind and Machine, 1984.
- (Squires and Preece, 99) Squires D. and Preece J., Predicting quality in educational software: Evaluating for learning, usability, and the synergy between them. In *Interacting with Computer*, vol. 11, issue 5, pp. 467-483, 1999.
- (Steels, 93) Steels L., Corporate knowledge management. In *Proceedings of ISMICK'93*, Compiègne, France, pp. 9-30, 1993.
- (Stojanovic et al., 01) Stojanovic L., Staab S., Studer R., ELearning Based on the Semantic Web. In *Proceedings of WebNet2001 - World Conference on the WWW and Internet*, Orlando, Florida, USA, 2001.
- (Stromboni, 00) Stromboni J.P., Enseigner l'automatique dans une école d'ingénieurs informatiques - Essais et expérimentations avec les NTIC. In *Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie (TICE'2000)*, pp. 197, 2000.
- (Stromboni et al. 01) Stromboni J.P., Sander P., Tigli J.Y., Some Experiments in Computer Aided Teaching. In *Proceedings of the 20th ICDE World Conference on Open Learning and Distance Education*, 2001.
- (Stromboni, 02) Stromboni J.P., Un cours introductif au traitement du signal à travers les applications audio de l'ordinateur multimédia. In *Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie (TICE'2002)*, 2002.
- (Stutt and Motta, 04) Stutt A. and Motta E., Semantic Learning Webs. In *Journal of Interactive Media in Education*, 2004.
- (Su, 01) Su H., Kramer D., Chen L., Claypool K., Rundensteiner E., XEM: Managing the Evolution of XML Documents, *11th International Workshop on research Issues in Data Engineering*, pp. 1-3, 2001.
- (Suzuki et al., 98) Suzuki J., Tashiro G., Abe Y., Yamamoti Y., Persona: a framework to provide adaptive presentation for web documents. In *Proceedings of the IPSJ Summer Programming Symposium, IPSJ Prosym '98*, pp. 101-108, 1998.

(Tattersall and Koper, 03) Tattersall C., Koper R., EML and IMS Learning Design: from LO to LA. Heerlen: OUNL, available at <http://dspace.learningnetworks.org>, 2004.

(Tchounikine et al., 04) Tchounikine P. et al., Platon-1 : quelques Dimensions pour l'analyse des travaux de recherche en conception d'EIAH. *Rapport de l'Action spécifique « Fondements théoriques et méthodologiques de la conception des EIAH. »*, STIC, CNRS, 2004.

(Tricot et al., 03) Tricot A., Plégat-Soutjis F., Camps J.-F., Amiel A., Lutz G., Morcillo A., Utilité, utilisabilité, acceptabilité : interpréter les relations entre trois dimensions de l'évaluation des EIAH. In *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2003)*, Strasbourg, pp. 391-402, 2003.

(Ullrich, 04) Ullrich. C., Description of an instructional ontology and its application in web services for education. In *Proceedings of Workshop on Applications of Semantic Web Technologies for E-learning (SW-EL'04)*, pp. 17-23, Hiroshima, Japan, November 2004.

(Uren et al., 05) Uren V., Cimiano P., Iria J., Handschuh S., Vargas-Vera M., Motta E., Ciravegna F., Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. In *Journal of Web Semantics*, vol. 4, issue 1, pp. 14-28, 2005.

(Uschold and Gruninger, 96) Uschold M. and Gruninger M., Ontologies: principles, methods and applications. In *The Knowledge Engineering Review*, vol. 11, pp. 93-136, 1996.

(Vargas-Vera et al., 02) Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., Ciravegna F., MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW 2002)*, pp. 379-391, Spain, 2002.

(Verbert et al., 05) Verbert K., Jovanovic J., Gasevic D., Duval E., Meire M., Towards a global component architecture for learning objects: a slide presentation framework. In *Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2005)*, Montreal, Canada, June 27 - July 2 2005.

(Villette, 99) Apprendre autrement aujourd'hui ? *10e Entretiens de la Villette*, Nov. 24 - 26 1999.

(Wiley, 00) Wiley D.A., Connecting learning objects to Instructional design theory: a definition, a metaphor, and a taxonomy. In Wiley (eds.), *The Instructional Use of Learning Objects*, 2000.

(Wiley, 03) Wiley, D.A.(2003) Learning Objects: Difficulties and Opportunities, http://wiley.ed.usu.edu/docs/lo_do.pdf

(Winter et al., 05) Winter M., Brooks C., Greer. J., Towards Best Practices for Semantic Web Student Modelling. In *Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED'05)*, Amsterdam, The Netherlands, July 18-22 2005.

(Woukeu et al., 04) Woukeu A., Wills G., Conole G., Carr L., Kampa S. and Hall W., Ontological Hypermedia in Education: A framework for building web-based educational portals. In *Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2004)*, Honolulu, Hawaii, USA, 2004.

(Zapata and Greer, 04) Zapata-Rivera J.D. and Greer J., Interacting with Inspectable Bayesian Student Models. In *International Journal of Artificial Intelligence in Education*, no. 14, pp. 127-163, 2004.

(Zimmermann et al., 06) Zimmermann B., Bergsträßer S., Rensing C., Steinmetz R. A Requirements Analysis of Adaptations of Re-Usable (E-Learning) Content. In *Proceedings of World Conference on Educational Multimedia, Hypermedia*, June 2006.

10.5. Web resources

All valid on 19th October 2006.

(ACM, 98) ACM Computing Classification System <http://www.acm.org/class/>

(Annotea, 06) Annotea Project, <http://www.w3.org/2001/Annotea/>.

(Ariadne, 06) Ariadne – Foundation for the European Knowledge Pool, <http://www.ariadne-eu.org/>

(ISO CG, 06) Common Logic Standard, <http://cl.tamu.edu/>

(CG, 06) Course Genie, <http://www.horizonwimba.com/products/coursegenie/>

(cohse, 06) Conceptual Open Hypermedia Service, <http://cohse.semanticweb.org/>

(DC, 06) DublinCore Dublin Core Metadata Element Set, Version 1.1, <http://dublincore.org/documents/dces/>

(Escot, 06) Educational Software Components of Tomorrow, <http://www.escot.org/>

(Foaf, 06) The Friend of a Friend (FOAF) project, <http://www.foaf-project.org/>

(Graphviz, 06) Graphviz - Graph Visualization Software, <http://www.graphviz.org/>

(IMS, 06) IMS Global Learning Consortium Inc., <http://www.imsglobal.org/>

(IsaViz, 06) IsaViz: A Visual Authoring Tool for RDF, <http://www.w3.org/2001/11/IsaViz/>

(Java, 06) The Java™ Tutorials, <http://java.sun.com/docs/books/tutorial/index.html>

(Jena, 06) Jena – A Semantic Web Framework for Java, <http://jena.sourceforge.net/>

(KW, 06) Knowledge Web FP6-507482, <http://knowledgeweb.semanticweb.org/>

(LOM, 06) LOM IEEE Standard for Learning Object Metadata (LOM): <http://ltsc.ieee.org/wg12/>

(Merlot, 06) MERLOT - Multimedia Educational Resources for Learning and Online Teaching <http://taste.merlot.org/>

(Ontobroker, 06) Ontobroker, <http://ontobroker.semanticweb.org/>

(OWL, 06) Web Ontology Language (OWL), <http://www.w3.org/2004/OWL/>.

(Protégé, 06) Protégé ontology editor, <http://protege.stanford.edu/>

(RDF, 06) Resource Description Framework (RDF), <http://www.w3.org/RDF/>.

(RIF, 06) Rule Interchange Format, W3C Working Group, <http://www.w3.org/2005/rules/wg>

(SCORM, 06) SCORM specification, The Advanced Learning Initiative, <http://www.adlnet.org/>.

(Sesame, 06) ...home of Sesame, <http://www.openrdf.org/>

(SKOS, 05) SKOS Core Guide, W3C Working Draft 10 May 2005, <http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510/>

(SPARQL, 06) SPARQL Query Language for RDF, W3C Working Draft 4 October 2006, <http://www.w3.org/TR/rdf-sparql-query/>

(SPARQL-XML, 06) SPARQL Query Results XML Format, W3C Candidate Recommendation 6 April 2006, <http://www.w3.org/TR/rdf-sparql-XMLres/>

(SVG, 06) Scalable Vector Graphics (SVG), XML Graphics for the Web <http://www.w3.org/Graphics/SVG/>

(Swebok, 06) Swebok Guide to the Software Engineering Body of Knowledge, <http://www.swebok.org>

(Swoop, 06) SWOOP - A Hypermedia-based Featherweight OWL Ontology Editor, <http://www.mindswap.org/2004/SWOOP/>

(SWRL, 04) SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission 21 May 2004 <http://www.w3.org/Submission/SWRL/>

(Triple, 06) TRIPLE Homepage, <http://triple.semanticweb.org/>

(vcard, 01) Representing vCard Objects in RDF/XML, W3C Note 22 February 2001, <http://www.w3.org/TR/vcard-rdf>

(Wordnet, 06) a lexical database for the English language, <http://wordnet.princeton.edu/>

(W3C, 06) W3C - The World Wide Web Consortium, <http://www.w3.org>

10.6. Index of Figures

Figure 1 – “iceberg”, retrieved from the World Wide Web, 26/07/2006	8
Figure 2 – Research context of the thesis	12
Figure 3 - Ontology of knowledge types, (Altenhofen and Schaper, 02).	14
Figure 4 - Constructivist pedagogical theories (Dalgarno, 96)	23
Figure 5 - Approaches of learning and instruction on a continuum of contextualization (Allert, 04)	23
Figure 6 – Example of an overlay model (Brusilovsky, 03)	32
Figure 7 - Representation of a knowledge state in the Aleks system (Falmagne et al., 00)	32
Figure 8 – Screen shot of the sql-tutor web interface	37
Figure 9 - An example of constraint in SQL-tutor (Mitrovic, 03)	37
Figure 10 – Prerequisite relationship between two resources in the AHA authoring environment	39
Figure 11 – Link recommendation in the AHA system	39
Figure 12 - Authoring of adapted content in AHA proprietary XML language	40
Figure 13 – Example of adaptation of the content	40
Figure 14 - Use of abstract conceptual representation, using abstract concepts in AHA authoring tool	41
Figure 15 - The knowledge triangle	42
Figure 16 - A logo for description logics	45
Figure 17 – Example of a conceptual graph, expressing that “Babar, an elephant, is grey”	46
Figure 18 – The vision of document and data base indexation with topic maps (Garshol, 03)	47
Figure 19 - Example of pedagogical ontologies (Ullrich, 04) (Leidig, 01)	51
Figure 20 – The semantic web layer cake (W3C, 06)	53
Figure 21 – A simple SPARQL query	56
Figure 22 - Screen shot of the protégé interface	57
Figure 23 - Screenshot of the Conzilla conceptual browser	59
Figure 24 - The magpie plug-in	60
Figure 25 – Example of a rule in TRIPLE	63
Figure 26 – Analysis dimensions for conception work in e-learning, after(Tchounikine et al., 04)	66
Figure 27 – From repositories to intelligent systems	68
Figure 28 - Navigation model based on domain ontology	74
Figure 29 - Navigation through concept identification in documents	75
Figure 30 - Link addition based on domain knowledge	75
Figure 31 - Path creation using ontological knowledge and reasoning	76
Figure 32 - Dynamic resource recommendation (or linking) using user’s path	76
Figure 33 – the global architecture of the QBLS system	79
Figure 34 – Dissociation of content and presentation in the semantization process.	85
Figure 35 – Overview of the annotation process	89
Figure 36 – Screenshot of the original course for QBLS-1	91
Figure 37 – The original slides of the course on Java programming	92
Figure 38 - The QBLS-1 Ontology (excerpt)	93
Figure 39 – The QBLS-2 ontology (excerpt)	94
Figure 40 - Layout of the final document in MS-Word	95
Figure 41 – Style definition in OpenOffice 2.0 (Oasis format)	95
Figure 42 – Style definition in MS-Word	96
Figure 43 - Annotation of learning resources with OpenOffice Writer and its style hierarchy	98
Figure 44 – Annotated content in MS-Word-XML format	101
Figure 45 - RDF annotation for QBLS	102
Figure 46 - XML format in OpenOffice Writer	103
Figure 47 – RDF generated for Java courses	103
Figure 48 - Conceptual structure in QBLS-1	109
Figure 49 – Conceptual structure in QBLS-2	110
Figure 50 – Comparison between the knowledge life cycle (Millard et al., 06) and the semantization process ..	115
Figure 51 – SPARQL query for structure matching	119
Figure 52 – The original document: a linear suite of resources	121
Figure 53 – Contextualization according to domain and pedagogical points of view	122
Figure 54 – Structuring knowledge to create conceptual paths in the content	122
Figure 55 – Impact of pedagogical knowledge on path creation	123
Figure 56 – Web-application architecture for learner access	126
Figure 57 – Interaction cycle between the learner and the system.	127
Figure 58 – Screenshot of the QBLS-1 interface, at the beginning of the learning task	129

Figure 59 – Simultaneous display of a question and a domain concept.....	130
Figure 60 – Screenshot of the learner interface for navigating a Java course	131
Figure 61 – RDF annotation about “carte son”	133
Figure 62 – Detail of an annotation in the editor.....	133
Figure 63 – Selection of a relation type in the editor	134
Figure 64 – Selection of an instance reference in the editor	134
Figure 65 – Excerpt of the SKOS vocabulary on Java programming	135
Figure 66 – A simple SPARQL query.....	135
Figure 67 – Result of a SPARQL query in the standard XML binding.....	136
Figure 68 – Querying to include narrower concepts	136
Figure 69 – Querying including pedagogical knowledge.....	137
Figure 70 – Propagation of the priority relation on the ontology.....	139
Figure 71 – Algorithm to compute a generic index on a partial relation order	140
Figure 72 – A query ordered according to the generated index	141
Figure 73 – Personalizing the query according to a user profile	142
Figure 74 – SPARQL query completed with information about labels and titles	142
Figure 75 – Additional triples queried to take into account contextual information	143
Figure 76 – Query determining the potential relations given a domain and a range concept	144
Figure 77 – JSP tag performing a query to the Corese search engine	147
Figure 78 – XML-binding for SPARQL results.....	147
Figure 79 – XSL stylesheet processing SPARQL result	148
Figure 80 – Generated HTML output.....	148
Figure 81 – Different information levels for learner tracking.....	153
Figure 82 – Example of a navigation path visualized on a graph of the course	154
Figure 83 – The RDFS log model	154
Figure 84 – Transformation pattern from two successive visits to a connected path.	155
Figure 85 – A screenshot of the QBLS-2 interface with indication of non-visited resources.	157
Figure 86 – Example of a conceptual space	158
Figure 87 – Visualization of the conceptual navigation.....	158
Figure 88 – Illustration of the back navigation on a conceptual space.....	159
Figure 89 – Query finding the previous document and concept indicated by the “back” button	160
Figure 90 – Process to generate graph representations from RDF files	162
Figure 91 – The log visualization interface.....	163
Figure 92 – SPARQL query to select paths spanning from one node.....	164
Figure 93 – Construction process for the visualization of the log analyzer.	165
Figure 94 – Selection of a single question and the associated conceptual path	166
Figure 95 – Visualization of a user path for a specific question	168
Figure 96 – Selecting steps from a group.....	170
Figure 97 – Comparison of two group behaviors on a focused part of the navigation space.....	171
Figure 98 – SPARQL query retrieving information about group behaviors with questions	171
Figure 99 – Comparison between a list display and a graph display for visits frequency.....	174
Figure 100 – Looking for behavioral patterns in the RDF graph	176
Figure 101 – Screenshot of the QBLS-2 interface accessed through a wiki page.....	182
Figure 102 – Usage statistics over the four months of the experiment.....	185
Figure 103 – User’s evaluation of QBLS support for two goals of on-line course systems.....	189
Figure 104 – Distribution of system accesses during the day.	192
Figure 105 – Method overview for creating and exploiting on-line courses from reused documents.....	201
Figure 106 – The magpie tool highlighting semantic web concepts on the ACACIA homepage	206
Figure 107 – QBLS interface for ASPL.....	207
Figure 108 – Combination of two services informing about the concept of “web”	208
Figure 109 – Overview of the ASPL platform	209

10.7. Index of Tables

Table 1 – Usage scenarios for each actor (adapted after (Azouaou et al., 03))	15
Table 2 – Role of the participants depending on the pedagogical artifacts, after (Schneider et al., 03).....	15
Table 3 – Bloom’s taxonomy of learning objectives	25
Table 4 – Base description logic (corresponds to the multi-modal logic K) Defines ALC:	46
Table 5 – Figures about QBLS-1 and QBLS-2	106
Table 6 – Comparison of different domain ontologies for Java programming	118

Table 7 - Concept overlap between the different ontologies.....	118
Table 8 - Match for hierarchical links	119
Table 9 - Rules used to infer priority relations between pedagogical concepts.....	140
Table 10 – Comparison between to different profiles in QBLS.....	142
Table 11 – Lines of code for theQBLS-2 learner interface	149
Table 12 – Different types of nodes in the course graph.....	163
Table 13 – Characteristics of different evaluation methods	196
Table 14 – Criteria for choosing a resource to reuse.....	202

Annexes

Annex 1 - Heuristic grids for evaluation

Nom et description de la grille	Auteur	Source
Heuristiques des dispositifs éducatifs		
Grille d'évaluation pédagogique et technique d'un site Web éducatif	Gilbert, D.	http://aptic.ulaval.ca/guidew3educatif/guide.php?HY=4.128 (dernièrement consulté en Janvier 2005)
<u>Conception et évaluation d'un outil virtuel de collaboration</u>	Charlier, B	http://tecfa.unige.ch/proj/recreasup/rapport/annexes_rapport_final_recreasup.pdf (dernièrement consulté en Janvier 2005)
Grille d'évaluation d' un site internet pédagogique	Perrot, T.	http://www3.unileon.es/dp/dfm/flenet/grilles2.html#TPerrot (dernièrement consulté en Janvier 2005)
Heuristiques "e-learning"		
Heuristic Evaluation Instrument and Protocol for E-Learning Programs	University of Georgia	http://it.coe.uga.edu/~treeves/edit8350/heureval.rtf (dernièrement consulté en Janvier 2005)
E-learning site usability checklist	Miller, D	http://www.stcsig.org/usability/resources/toolkit/e-learning-checklist.doc (dernièrement consulté en Janvier 2005)
Grille d'heuristique pour l'évaluation de dispositif de e-learning – aspects repérage	Bellino, C.	Mémoire de DESS d'Ergonomie Cognitive des NTIC (2002) mailto : catherine.bellino@dia-logos.net
Heuristiques et critères "génériques"		
Critères ergonomiques de Scapin et Bastien (appliqués au web)	Scapin, Bastien	http://ergolab.net/download/criteresergonomiques1_1.pdf (dernièrement consulté en Janvier 2005)
Guide pratique de conception et d'évaluation ergonomique des sites web	Millerand, F. & Martial, O.	http://www.crim.ca/rd/Guide%20ergonomique.PDF (dernièrement consulté en Janvier 2005)
Heuristiques de Nielsen adaptées au web	Keith Instone, d'après Nielsen	http://www.w3j.com/5/s3.instone.html (dernièrement consulté en Janvier 2005)
Xerox: Heuristic Evaluation: A system Checklist	Usability Analysis & Design, Xerox Corporation	http://it.civil.auc.dk/it/education/reports/he_cklst.pdf (dernièrement consulté en Janvier 2005)

Questionnaire pour le participant

(n'hésitez pas à utiliser les extrémités de l'échelle) :

- Page 231

- Si oui, pouvez vous donner 2-3 exemples de telles fiches ?
12. J'ai tiré parti des titres des onglets (Définition, Exemple, etc...) 1 2 3 4 5
pour me guider dans la navigation
13. Je trouve que les titres des onglets, tels que « Précision » ou 1 2 3 4 5
« Formalisation » correspondent bien à leur contenu
- Si non, quel(s) titre(s) proposeriez-vous ?
14. Les mots « Cours » « Thème » « Notion » utilisés dans l'interface 1 2 3 4 5
ne m'ont pas posé de problème de compréhension
- Si non, lesquels ?
15. J'ai été guidé par les titres des questions 1 2 3 4 5
- Si non, quel(s) titre(s) vous semblent ne pas correspondre au contenu de la question :
16. J'ai été gêné par le fait que les notions consultées remplacent l'une 1 2 3 4 5
par l'autre dans la même zone de l'écran
17. J'aurais souvent préféré avoir une zone supplémentaire en dessous 1 2 3 4 5
pour voir deux notions en même temps
- Si oui, pouvez vous donner 2-3 exemples d'une telle situation
18. Grâce a ce système, j'aurais pu faire les TD seul(e) a la maison 1 2 3 4 5
19. Je réviserai mes cours pour l'examen en essayant de répondre de 1 2 3 4 5
nouveau aux questions de TD avec ce système : je trouve en effet
cette méthode de révision efficace.
20. Je pense que j'aimerais bien utiliser ce système fréquemment 1 2 3 4 5
21. J'ai trouvé que cette interface était facile à utiliser 1 2 3 4 5
22. Je trouve qu'il y a beaucoup d'incohérence dans ce système 1 2 3 4 5
23. Je pense que la plupart des utilisateurs apprennent à utiliser ce 1 2 3 4 5
système très rapidement
24. J'ai trouvé que le choix de la couleur du fond permet de lire 1 2 3 4 5
facilement la page-écran
25. J'ai été gêné par d'autres couleurs présentes sur l'écran 1 2 3 4 5
- Si oui, par la couleur de quel(s) élément(s) ?

26. J'ai trouvé la présence des liens dans le texte très pratique 1 2 3 4 5
27. J'aurais préféré avoir la liste des liens à part au lieu d'avoir les liens dans le texte 1 2 3 4 5
28. La présence du glossaire m'a beaucoup aidé dans la réalisation de ma tâche 1 2 3 4 5
29. J'ai rapidement compris à quoi sert la zone « Historique des notions consultées » 1 2 3 4 5
30. Je trouve que cette fonctionnalité facilite beaucoup la consultation des fiches lors de la résolution des questions 1 2 3 4 5
31. J'ai trouvé que la disposition de cette zone est idéale 1 2 3 4 5
- Si non, quel(s) endroit(s) proposeriez-vous ?
32. Je trouve que son titre n'est pas suffisamment explicite 1 2 3 4 5
- Si oui, quel titre proposeriez-vous ?
33. Je n'ai pas besoin de différencier la zone supérieure de l'écran de la zone inférieure par le code couleur 1 2 3 4 5

RESUMÉ EN LANGUE FRANCAISE

1. INTRODUCTION

1.1. Historique

1.1.1. Les supports de la connaissance

En préambule à la présentation du travail réalisé pour ce manuscrit, un petit tour d'horizon historique s'impose sur la façon dont l'homme, au travers des âges, a transmis sa connaissance. Bien entendu il ne s'agit là que d'une vision très « occidentale » et partielle des choses.

Préalablement à l'invention de l'écriture, il est admis que les hommes échangeaient entre eux au moyen de la communication orale et gestuelle. Les historiens relèvent les premières traces d'écriture dans le croissant fertile il y a 9000 ans. Avec l'avènement des célèbres hiéroglyphes égyptiens, portés sur les rouleaux de papyrus, les connaissances (histoire, comptabilité, etc.) ont commencé à être stockées sur des supports physiques mobiles et inertes, dont certains sont encore lisibles aujourd'hui. Les bibliothèques, comme celle d'Alexandrie, ont tenté de rassembler les ouvrages les plus significatifs et construisirent les premières « mémoires » qui devaient perdurer au travers des siècles. Malheureusement la fragilité de telles entreprises ne leur a pas permis, dans leur majorité, de résister aussi longtemps. Plus de 1000 ans après, en Europe, les moines sont devenus les gardiens du savoir. Leur activité de réplique et de conservation des ouvrages a continué tout au long du moyen âge. En tant que gardien des supports de connaissance, ils étaient également en charge de l'enseignement de la lecture et de l'écriture, afin d'assurer que leur pratique ne se perde pas comme pour les hiéroglyphes. Malheureusement la copie manuelle était lente et pas toujours fidèle.

Le quinzième siècle marque alors le début d'une nouvelle ère avec la découverte en Europe de l'imprimerie. Avec la dissémination des livres la connaissance devient accessible à la masse des lecteurs. Les écoles se développent et les connaissances scientifiques de toute nature s'enrichissent irrésistiblement. Pendant plusieurs siècles cette situation reste inchangée. Le livre imprimé étant le principal dépositaire de la connaissance humaine, à l'image de l'encyclopédie de Diderot, il est utilisé comme outil d'apprentissage par excellence. Cependant, l'efficacité de l'apprentissage ne peut être obtenue qu'au travers d'un enseignement mettant en jeu d'indispensables professeurs. La relation maître/apprenant n'a quant à elle quasiment pas évolué depuis les premiers jours de la civilisation grecque.

A l'aube du troisième millénaire de notre calendrier (soit depuis une dizaine d'années) se déroule une révolution sans précédent avec notre entrée dans « l'ère numérique ». La possibilité de répliquer le contenu des livres dans la mémoire des ordinateurs devient quasi illimitée. Les données numériques s'échangent au travers du réseau mondial internet avec une facilité inimaginable quelques années auparavant. Cette percée technologique, tout comme l'imprimerie à son époque, déclenche de nouvelles pratiques dans notre rapport à la connaissance et à l'apprentissage. Aujourd'hui on peut considérer le World Wide Web comme l'accès privilégié à la connaissance humaine et à sa propagation. Ce constat sera le point de départ de notre réflexion.

1.1.2. Apprentissage et enseignement

La compréhension de l'intelligence humaine est une préoccupation récurrente des scientifiques. Ce sujet a connu de considérables développements, offrant une meilleure compréhension de l'apprentissage et de ses relations avec la connaissance en général, et dernièrement avec la connaissance sous forme numérique. L'acquisition de savoir fait parti intégrante de cette intelligence humaine et ne peut pas être séparée de la connaissance en elle-

même. Ainsi par connaissance on ne désigne pas uniquement ce qui est écrit dans les livres mais aussi les compétences, l'attitude et l'expertise nécessaires pour l'acquérir.

Les grecs avaient déjà analysé et discuté les pratiques d'enseignement. De nombreuses théories ont vu le jour depuis. Les toutes dernières, basées sur les avancées dans le domaine des neurosciences, en sont encore à leurs balbutiements. Ce que l'on sait c'est qu'il n'existe certainement pas une seule bonne façon d'apprendre. Un processus aussi complexe est profondément lié à des phénomènes aussi bien physiologiques que psychologiques qu'il est pour l'instant difficile d'appréhender.

Un autre aspect de l'apprentissage est son côté sociologique. La façon dont nous apprenons est motivée et guidée par la société dans laquelle nous vivons. Platon n'a pas appris de Socrate comme nos adolescents découvrent la géographie avec Google-earth. Il est de tout temps naturel que l'apprentissage s'effectue en utilisant les objets qui nous entourent. Cependant nous sommes seulement au commencement de cette ère numérique et de nombreuses interrogations subsistent.

Apprendre ne deviendra de toute façon jamais simple. Mais notre curiosité et notre imagination nous pousse à toujours chercher de meilleures façons, plus rapides et plus efficaces d'apprendre. Cette quête est peut-être caractéristique de la nature humaine. Au vingt et unième siècle on appelle cette quête la recherche. Elle est menée par des hommes en blouse blanche avec de grosses lunettes, enfin presque... Nous espérons que la courte introduction présentée ici donnera au lecteur au moins un aperçu de l'importance et de l'impact de ce sujet, développé plus en détail tout au long de ce manuscrit.

1.2. Le e-learning et la gestion des connaissances

Le terme « e-learning » désigne pour nous toute activité relative à l'apprentissage utilisant des supports numériques. A son apparition, l'agitation générée autour de ce terme a masqué l'innovation technologique qu'il recouvrait : l'augmentation de la puissance et la diffusion des ordinateurs personnels. Cependant cette innovation n'a jamais signifié qu'apprendre deviendrait sans effort et que ce mode d'apprentissage fournirait un soit disant « *just-in-time education integrated with high value chains*. » (Drucker, 00). Certes, l'étendue des possibilités maintenant offertes dépasse l'imagination.

Il y a donc à la fois une révolution et très peu de nouveauté dans ce terme. En ce qui concerne la façon dont les gens apprennent c'est une véritable révolution que nous détaillons par la suite. Le glissement de la relation maître/élève vers une interaction médiatisée par l'outil informatique constitue certainement un des changements les plus radicaux dans les pratiques d'apprentissage. Cependant le temps et les efforts demeurent nécessaires pour un apprentissage de qualité.

L'étude de l'impact de la société sur la façon dont les gens apprennent, montre des individus toujours plus mobiles, flexibles et adaptables. Ceci trouve un echo dans le besoin grandissant de programmes éducatifs s'adressant aux adultes, aux professionnels ou encore dans les universités du troisième âge. Le terme « life-long learning » a été introduit pour décrire cette évolution. Il suppose en particulier que nous ne nous reposons plus uniquement sur l'apprentissage à l'école. Le rythme des évolutions technologiques est tel que la période de validité d'un savoir ou d'une compétence est maintenant bien inférieure à la durée de vie des individus, qui en parallèle a augmenté. Cette perspective d'un apprentissage tout au long de la vie, aujourd'hui largement acceptée, constitue un des arguments majeurs en faveur de l'e-learning. Le slogan « *anytime, anywhere* » prend tout son sens ici lorsqu'on sait que

l'apprentissage doit prendre sa place au travail et à la maison, et ce à toutes les étapes de la vie.

L'industrie du e-learning est jeune et en constante mutation. Elle répond au besoin des entreprises pour mettre à jour les compétences de leur salariés et pour considérer les difficultés à combiner les contraintes de l'éducation « classique » (présentiels, exercices surveillés, etc.) avec les agendas professionnels. Les dernières années ont vu toute une industrie émerger des petites structures d'enseignement à distance. Les analystes économiques donnent différents chiffres concernant l'expansion de cette activité, difficile à surveiller du fait de ses multiples formes, mais qui connaît indubitablement une forte croissance. Par exemple des analystes technologiques estiment que le marché américain (US) pour le e-learning en entreprise approchera les 10,6 milliards de dollars en 2007, selon la compagnie IDC.

“The real bottom line is that whether they need to brush up their skills on a product or learn about a new technology, [...] professionals no longer must spend days away from work, sitting in a classroom. Thanks to a host of well-established and new companies, ambitious technology employees today have access to a wide world of information, labs, educators and research material that can help improve their productivity, capabilities and employability -- all without requiring them to leave their desks.”

Extrait de <http://www.technewsworld.com/story/32325.html>

Les philosophes (Villette, 99), voient l'utilisation des technologies de l'information pour l'apprentissage et l'enseignement, notre « e-learning », comme une révolution dans les pratiques :

Au niveau académique, à l'école ou bien à l'université, de telles modifications affectent l'activité d'apprentissage elle-même, mais également le rôle de l'enseignant. Lorsqu'un cours se déroule à distance en particulier, la plupart du temps les apprenants font face à un écran et non une personne physique. L'enseignant devient un médiateur entre la source de connaissance et l'apprenant. Lui/elle n'est plus le détenteur du savoir mais l'agent qui guide l'élève dans son apprentissage. Certains penseurs expriment leurs craintes au sujet de cette virtualisation de l'enseignement qui pourrait fausser le rapport au réel. De plus l'activité se focalise autour de l'apprentissage et non plus de l'enseignement, mettant presque l'enseignant de « côté », ce qui est une révolution comparé aux anciennes pratiques. Le e-learning se concentre donc d'avantage sur l'élève et son activité.

Au niveau professionnel, de telles technologies sont assez répandues. L'adoption de nouveaux outils y est certes généralement plus rapide que dans le milieu académique. Cependant ils apportent un autre type de modification dans le fonctionnement des entreprises. La stricte répartition des rôles et les conceptions tayloristes s'y révèlent inadaptées. Elles n'y subsistent qu'en raison des rigidités organisationnelles et laissent progressivement la place à une organisation plus ouverte dans laquelle les employés partagent un but de production commun. Ceci implique une formation continue, et une approche pédagogique ancrée dans la réalité. Des investissements sont donc nécessaires. Par exemple dans le domaine industriel les outils de production ne sont pas toujours adéquats pour suivre des formations, et des outils virtuels intermédiaires doivent être développés.

Le domaine professionnel utilise le terme d' « organisation apprenante » avec l'idée d'apprentissage tout au long de la vie. Les entreprises considèrent que la gestion des compétences et du savoir est stratégique. Elle constitue leur principal atout dans une économie de l'immatériel. Dans ce contexte, elles tentent de développer des mémoires

organisationnelles, qui doivent s'articuler avec les efforts de formation. Ces mémoires doivent évoluer avec l'entreprise et les évolutions sont maintenant tellement rapides que leur stabilité devient une question cruciale. De même les futurs collaborateurs y sont principalement sélectionnés sur leur attitude préférablement à leur diplôme et leur expertise pour s'assurer qu'ils vont s'adapter à l'organisation.

De telles organisations apprenantes montrent alors la connexion importante entre apprentissage et gestion des connaissances dans le milieu professionnel. Le terme « Knowledge Management » est habituellement utilisé pour identifier les pratiques de partage, de conservation et d'explicitation de la connaissance. Ces éléments sont partagés avec la formation professionnelle qui vise à donner aux employés de nouvelles connaissances pratiques pour leur travail.

Avec la percée des technologies numériques, la gestion des connaissances (KM) s'est développée comme une discipline technique visant à gérer les connaissances explicites d'une organisation. La boîte à outil du « knowledge manager » contient des programmes pour gérer non seulement le contenu mais aussi tout type d'information pouvant s'exprimer sous forme numérique (voir l'exemples des systèmes ERP, CRM, etc.). Ces outils implémentent des méthodes et des pratiques établies pour la gestion des connaissances. Elles peuvent être spécifiques à une organisation ou bien définies comme des standards. De tels programmes reposent sur des formalismes et des algorithmes spécifiques pour exprimer et manipuler les connaissances. C'est ce que nous étudions dans la suite pour le contexte spécifique du e-learning.

1.3. Plan

Tout en gardant à l'esprit les considérations mentionnées plus haut, nous utiliserons le plan suivant pour présenter notre travail et nos contributions.

- Tout d'abord, nous précisons le contexte scientifique de ce travail. Les questions clés sont présentées et nous identifions les inspirations et tendances majeures. Nous nous positionnerons dans ce contexte en donnant au lecteur novice une meilleure compréhension des propositions scientifiques.
- Ensuite un panorama des travaux existants est esquissé. La très large étendue du domaine de recherche du e-learning couplé avec une communauté de recherche sur la gestion des connaissances bien établie nous donne un nombre de publications très important sur lesquelles nous appuyer. Les contributions mentionnées sont sélectionnées en fonction de leur lien avec un ou plusieurs aspects technologiques ou conceptuels de notre travail. Uniquement les travaux les plus reconnus, basés sur leur publication dans les journaux et conférences du domaine sont mentionnés. Quelques sources provenant directement du Web sont également citées lorsqu'elles font l'objet d'une large reconnaissance de la part de la communauté.
- Le quatrième chapitre présente l'approche générale de conception du système de e-learning faisant l'objet de cette thèse. Nous présentons l'analyse des besoins exprimés par les enseignants rencontrés, et l'architecture générale que nous avons adoptée. Les détails de cette architecture qui donnent lieu à des contributions significatives sont décrits dans les trois chapitres suivants.
- Premièrement, nous expliquons pourquoi le processus d'annotation était nécessaire pour l'exploitation automatique des documents, et détaillons la solution que nous proposons, les expérimentations conduites ainsi que le principe général d'annotation proposé.
- Représentant une des motivations de ce travail, les technologies du web sémantique constituent le substrat sur lequel nous avons bâti notre application. Nous expliquons précisément les fonctionnalités apportées par cette technologie et comment elles se révèlent

efficaces. Cette section détaille leur implémentation, les améliorations potentielles et les bénéfices qui peuvent en résulter de manière générale.

➤ Finalement nous explorons l'idée que la visualisation d'un cours annoté fourni différentes façons d'apprendre mais aussi différentes façons d'appréhender cette activité. Nous proposons un modèle d'analyse pour mieux comprendre et interpréter les parcours des utilisateurs sur l'espace de navigation qui leur est offert. Ceci contribue à l'évaluation globale du système, mais apporte également des pistes intéressantes pour la modélisation de l'apprenant, le suivi et le contrôle.

➤ Une section séparée présente l'évaluation complète du système. Les questions de réutilisabilité et de généralisation de l'outil y sont en particulier discutées.

➤ Finalement, nous synthétisons l'ensemble des contributions en proposant une méthode générique pour la sémantisation de cours en ligne dans le contexte de l'apprentissage avec ordinateur.

Nous insistons sur le fait que, depuis le départ, notre vision est fortement guidée par les problèmes de faisabilité et d'application réelle. Etant donné l'état actuel des recherches dans le domaine, l'écart entre les avancées théoriques et les pratiques était tel qu'il nous a semblé opportun de placer notre contribution dans cette optique. Ceci est d'ailleurs illustré par le slogan du réseau d'excellence européen supportant ce travail : « Realizing the semantic web ». Les aspects de passage à l'échelle, de réutilisabilité et d'utilisabilité seront donc de toute première importance. Ainsi nous nous efforcerons de généraliser les expériences, outils et méthodes présentés.

2. CONTEXTE DU TRAVAIL

3. ETAT DE L'ART

4. CONCEPTION ET DESCRIPTION GENERALE DU SYSTEME

5. SEMANTISATION DE DOCUMENTS DE COURS

6. EXPLOITATION DES TECHNOLOGIES DU WEB SEMANTIQUE

7. VERS UNE ANALYSE SEMANTIQUE DES PARCOURS D'APPRENTISSAGE

8. EVALUATION : MÉTHODE ET RÉSULTATS

9. MÉTHODE GÉNÉRALE

10. CONCLUSION

Les solutions proposées tout au long de cette thèse nous ont amenées à définir une méthode globale pour la réutilisation de documents de cours, leur mise en ligne, et leur exploitation au travers d'une navigation basée sur des représentations explicites de connaissances.

L'implémentation de QBLS nous a permis de conduire plusieurs expérimentations en situation réelle de la méthode proposée tout en démontrant le potentiel technologique du web sémantique pour ce type de système d'apprentissage.

Au vu de l'application de cette méthode, un certain nombre de conclusions scientifiques s'imposent. Nous les présentons ci-dessous. Dans un second temps ces conclusions se retrouvent dans les résultats de la thèse ensuite exposés. Enfin, elles ouvrent des perspectives à plus ou moins long terme sur lesquelles nous concluons ce travail.

10.1. Conclusions scientifiques

10.1.1. Sémantique et apprentissage

Nous avons présenté dans les chapitres précédents comment des connaissances formalisées peuvent être utilisées comme support à la navigation au sein d'un cours existant. En dépit des nombreuses propositions existantes, notre travail est original par l'aspect « pratique » qu'il privilégie :

- Tout d'abord nous visions la réduction du travail de l'enseignant au travers de la réutilisation des cours.
- Ensuite nous nous sommes concentrés sur des fonctionnalités réalistes en gardant à l'esprit les contraintes de faisabilité d'un environnement réel.

En considérant ces hypothèses nous avançons les thèses suivantes :

Les connaissances à exprimer par les enseignants doivent être limitées :

Durant les trois années de notre projet, nous avons en effet acquis la certitude que les solutions actuellement proposées pour les systèmes d'apprentissage « intelligents » ou « avancés » nécessitent un effort bien trop important de la part de l'enseignant. Cet effort combine la création de contenu avec l'expression et la formalisation de connaissances relatives à l'apprentissage. Il se révèle bien souvent insurmontable. Bien que décrit dans plusieurs travaux de recherche comme une simple limitation financière, nous pensons à contrario qu'il s'agit là d'une limitation intrinsèque ne pouvant généralement pas être résolue. Par exemple nous avons montré que l'expression de relations de pré-requis était souvent inadaptée, et ce quelque soit l'effort consenti par l'enseignant.

En conséquence nous proposons de limiter l'expressivité des modèles, en rapport avec les possibilités qu'ont l'enseignant et le matériel pédagogique de les instancier.

Les connaissances formelles ne sont utiles que si elles ont été exprimées de manière contextuelle par rapport à la situation d'apprentissage.

Nous avons identifié les modèles qui pouvaient posséder une valeur plus ou moins universelle (ontologie pédagogique, méta model SKOS, etc.) et défendons que les modèles du domaine en particulier, sont dépendant en grande partie du contenu pédagogique employé et de la vision personnelle de l'enseignant.

En analysant les connaissances nécessaires pour offrir une navigation conceptuelle, nous avons montré leur nature essentiellement contextuelle, et la difficulté de toute réutilisation. Nous pensons que cette caractéristique est propre à la question de l'apprentissage. En effet apprendre ne peut pas être réduit à un simple transfert de connaissances décrites dans un modèle unique.

Une analyse de l'activité de l'apprenant basée sur les connaissances manipulées apporte un gain d'information.

Nous avons proposé un modèle de navigation qui présente un intérêt à la fois pour l'accès aux documents qu'il propose (navigation conceptuelle) et pour l'interprétation possible de l'activité d'apprentissage durant la navigation.

Au final nous pensons que les résultats obtenus au cours des expériences mettant en jeu de réels utilisateurs sont très encourageants pour le type d'applications visées, jusqu'à présent plutôt considérées comme illusoire par les enseignants.

10.1.2. Une approche basée sur le Web Sémantique

Les technologies du web sémantique se révèlent efficaces pour l'implémentation d'un système d'apprentissage « à base de connaissance ».

La première idée que nous défendons concerne l'intérêt d'intégrer les standards et les outils du web sémantique dans ce type d'application. Le déploiement de cette technologie requiert certes des efforts d'ingénierie, principalement du fait de sa nouveauté, mais les gains démontrés dans la simplicité et la souplesse de l'application développée nous incitent à considérer les technologies du web sémantique comme un grand pas en avant pour l'implémentation de système de e-learning dit « intelligents » ou en tout cas reposant sur des connaissances formalisées.

Nous considérons l'exemple du système QBLS, comme emblématique de cette avancée. En reposant sur les standards du W3C comme RDF et le futur SPARQL, la genericité des résultats obtenus est très importante. En particulier celle-ci ne repose à priori pas sur les particularités du moteur de recherche sémantique Corese employé dans notre cas.

Concernant le niveau d'inférence nécessaire dans le contexte du e-learning, nous avons trouvé que les besoins étaient relativement faibles comparés à l'expressivité du langage OWL. En revanche l'expression de règles, indispensable, reste une lacune en terme de standard. Les besoins penchent également clairement vers l'expression d'un langage de requête comme SPARQL (pas encore adopté comme standard), illustrant ainsi le besoin d'opérationnalisation des connaissances dans les technologies du web sémantique.

L'interopérabilité des connaissances formelles exprimées dans des langages standard est possible et utile.

Nous défendons également le rôle du web sémantique comme une solution effective pour certains aspects d'interopérabilité des connaissances formelles. En réutilisant des ontologies existantes exprimées en OWL et en les insérant directement dans QBLS, nous avons démontré la faisabilité de cette approche avec les outils actuels.

L'approche « web sémantique d'entreprise » peut bénéficier aux systèmes d'apprentissage

L'architecture et les scénarios envisagés pour le système QBLS étendent l'approche « web sémantique d'entreprise » au domaine du e-learning. Au lieu de viser la connaissance dans une entreprise, nous nous attachons à appliquer ces principes au sein d'une « organisation apprenante ». Notre vision du web sémantique n'est pas théorique. Nous ne défendons pas l'idée d'un réseau mondial où les connaissances formalisées pourraient être échangées et réutilisées à volonté dans différents dispositifs d'apprentissage. En revanche nous proposons une approche réaliste où les connaissances sont partagées au sein d'une petite communauté tout en bénéficiant des langages et des outils communs, mis en œuvre à moindre frais.

10.2. Résultats

Deux productions majeures résultent de ce travail : un guide méthodologique et un exemple d'implémentation. Ces deux résultats nous permettent d'intervenir sur l'idée de « learning object ».

10.2.1. Un guide méthodologique

La création de cours en ligne fait l'objet de nombreuses publications, essais et guides tentant de répondre aux interrogations grandissantes des enseignants face à cette nouvelle tâche. La méthode que nous proposons aborde la question sous l'angle de la réutilisation de matériel existant. Le rôle pouvant être joué par des connaissances formelles y est précisé et nous privilégions une approche pratique et orientée utilisateur, qui nous l'espérons bénéficiera non seulement aux futurs travaux de recherche mais aussi aux nouvelles pratiques des enseignants.

10.2.2. Une implémentation exemplaire

Le système QBLS démontre non seulement l'efficacité de l'approche web sémantique en terme de développement, mais constitue une réelle application modulaire dont les composants peuvent être réutilisés. En particulier, le module d'annotation, supportant les outils d'édition actuels constitue une brique logicielle sans équivalent et particulièrement utile.

10.2.3. La fin des « Learning Object »

L'idée même de « Learning Object » doit être maintenant largement rejetée comme constituant un rêve d'ingénieur bien loin des pratiques et des possibilités humaines. Néanmoins, les efforts consentis en vue de la réalisation de cette vision nous fournissent aujourd'hui les fondations sur lesquelles repose notre proposition.

Par exemple, la possibilité de rechercher des ressources de grande taille sur des portails dédiés constitue une possible première étape pour la méthode que nous introduisons. La philosophie de « découper » les documents en unités de plus petite taille constitue bien évidemment une inspiration majeure dans notre démarche, même si nous n'envisageons pas le partage et l'échange de ces unités.

Finalement, nous pensons que la vision des Learning Object qui place le contenu pédagogique au centre de la communication entre l'enseignant et l'étudiant ne doit pas être sous-estimée. Si les recherches actuelles se tournent vers d'autres aspects de la « préparation » d'un cours comme les « design pédagogiques », le contenu pédagogique reste la principale interface de communication et en conséquence garde un rôle prépondérant.

10.3. Perspectives

10.3.1. Intégration des outils d'annotation

Dans un futur proche, la convergence vers les formalismes XML devrait atteindre son apogée. Dans les navigateurs le XHTML s'enrichit de contenu SVG ou MathML permettant aux enseignants d'offrir des contenus plus riches « en-ligne ». Les éditeurs de contenu classique reposent également sur des formats XML, permettant de puissantes manipulations, comme celles proposées par le langage XSLT 2.0.

Cependant l'extraction de connaissances depuis ces sources requière toujours un important effort d'ingénierie. La bonne direction semble être d'intégrer de plus en plus les outils d'annotation dans les éditeurs eux-mêmes. L'annotation sémantique deviendrait partie prenante du processus d'édition/création d'un document. Si c'est déjà le cas pour les outils dédiés au e-learning, les éditeurs standard ne proposent pour l'instant que de simples métadonnées concernant le titre et l'auteur du document.

Nous envisageons donc l'évolution future des Power-Point et Word aujourd'hui en usage vers de réelles applications sémantiques. De tels outils joueront indéniablement un rôle capital en

vue d'intégrer plus de connaissances formelles dans les futures applications, et pas uniquement dans le domaine de l'enseignement.

10.3.2. Vers des systèmes adaptatifs complexes avec le web sémantique

L'adaptation ou la personnalisation des sites devient de plus en plus populaire sur le web en général. Avec l'idée de web 2.0 et des pages de plus en plus réactives, le dialogue classique requête/réponse se trouve peu à peu bouleversé au profit d'interactions beaucoup plus riches. Le couplage de pages dynamiques avec des moteurs d'adaptation va permettre aux designers de mettre en place des paradigmes d'interaction beaucoup plus complexes et centrés autour de l'utilisateur.

Cela n'a jamais été réalisé encore à grande échelle du fait des difficultés techniques pour faire fonctionner des clients réactifs chez les utilisateurs. Mais avec l'intégration de ces fonctionnalités au cœur des navigateurs, ces techniques vont connaître une prochaine large diffusion. L'architecture que nous proposons côté serveur, reposant sur un « middleware sémantique » peut très bien supporter ce type d'interaction. La production de services pour des interfaces sémantiques dynamiques devrait alors connaître un grand développement.

Dans ce monde « en ligne » les questions de performance, de passage à l'échelle vont devenir cruciales pour les plateformes web sémantique au fur et à mesure que le nombre et la complexité des requêtes croîtront avec la taille des bases de connaissances.

10.3.3. Le futur de l'e-learning

Nous concluons ce travail par une vision personnelle sur le futur des technologies liées au e-learning en général. Nous avons déjà mentionné que la définition de « e-learning » était vaste et rassemblait maintenant la plupart des situations d'apprentissage du fait de l'omniprésence de l'outil informatique dans notre vie quotidienne. De fait nous n'envisageons pas que les développements futurs de ces pratiques seront le fait d'enseignants volontaires ou visionnaires, mais découleront plutôt d'une présence encore plus importante de l'ordinateur et du réseau dans notre mode de vie. Notre rôle en tant que chercheurs n'est donc pas de promouvoir l'intégration de plus de support digitaux dans l'enseignement au nom d'une modernité qui nous a distancé depuis bien longtemps, mais bien de préparer et proposer des solutions qui deviendront progressivement des pratiques communes tandis que nous tous, apprenants, poursuivons notre route dans l'ère numérique.

Exploiting Semantic Web and Knowledge Management Technologies for E-learning

Abstract:

This work explores the potential applications of semantic web and knowledge management technologies for e-learning systems for accessing courses. It particularly addresses the difficult problem of reusing existing resources.

In this scope, we propose a methodology and an associated system (QBLS) relying on cutting-edge semantic web tools and formalisms, that particularly develops three aspects:

- Reusing existing pedagogical resources through a process called “semantization”. Pedagogical content is semi-automatically annotated with ontological concepts, from several points of view (domain, pedagogy, structure). It aims at contextualizing the material with regard to the teacher’s conceptualization and pedagogical strategy.
- Exploiting annotated pedagogical resources in dynamic web interfaces. We show how semantic web technologies efficiently perform inferences based on ontological knowledge to personalize and adapt courses. The interest of semantic web standards (OWL, RDF, SPARQL) is highlighted in a practical perspective, relying on the deployment of the semantic search engine Corese.
- Tracking learner activity and analysis of this activity. We explore the potential of graphic based visualization and inferences to manually or automatically analyze activity traces on a semantized course.

The theoretical proposal is supported by our implementation of the Question Based Learning System where learning resources available on the web are reused, annotated and exploited in real world experiments using semantic web technologies. The practical examples mentioned in this work are illustrated by two experiments conducted at the EPU of Nice Sophia-Antipolis.

Exploitation des Technologies du Web Sémantique et de la Gestion des Connaissances pour le E-Learning

Résumé :

Ce travail explore le potentiel des technologies du web sémantique et de la gestion des connaissances dans le cadre des systèmes de e-learning, notamment ceux présentant des contenus de cours « en ligne » de manière facilitée et éventuellement personnalisée, et dans lesquelles le problème de la réutilisation de ressources existantes demeure un problème difficile.

Dans ce contexte nous proposons une méthodologie et un système associé (QBLS), inspiré des pratiques en gestion des connaissances et reposant sur des outils et formalismes avancés du web sémantique. Notre proposition repose principalement sur trois points :

- La réutilisation de ressources pédagogiques existantes au travers d’un processus dit de « sémantisation ». Ce processus vise à contextualiser le contenu par rapport à la conceptualisation et la stratégie pédagogique de l’enseignant.
- L’exploitation de ressources pédagogiques annotées dans des interfaces web dynamiques. Nous insistons sur l’intérêt des standards du web sémantiques (OWL, RDF, SPARQL) dans une perspective pratique, reposant sur le déploiement du moteur de recherche sémantique Corese.
- Le suivi et l’analyse de l’activité de l’apprenant. Nous explorons le potentiel des visualisations et inférences à base de graphes pour l’analyse automatique ou manuelle des traces d’activités dans un cours sémantisé.

Cette proposition théorique est appuyée par notre implémentation du système QBLS, pour « Question Based Learning System », où des ressources pédagogiques sont réutilisées, annotées et exploitées dans le cadre d’expérimentations en situation réelle menées à l’EPU de Nice Sophia-Antipolis.