Université de Genève
Faculté de Psychologie et des Sciences de l'Education
TECFA (Technologies et Formation de l'Apprentissage)

EVA_pm:

Design and Development of a Scaffolding Environment For Students Projects

(EVA_pm: conception et réalisation d'un environment d'étayage pour des projets d'étudiants)

Mémoire en vue de l'obtention du DESS
en Sciences et Technologies de l 'Apprentissage et de la Formation

By

Paraskevi SYNTETA
(http://tecfa.unige.ch/~paraskev)
(Paraskevi.Synteta@tecfa.unige.ch)

Defended on 6 July 2001

Jury:

- Prof. Patrick Mendelsohn, Professeur, TECFA, Univ. of Geneva, (thesis director)
- Dr. Daniel Schneider, Chargé de cours, TECFA, Univ. of Geneva (thesis co-director)
- Dr. Gilles Falquet, Maître d'enseignement et de recherche, Computer Science Dept., Univ. of Geneva

TABLE OF CONTENTS

ABSTRACT vii

Resumé vii

ACKNOWLEDGEMENTS i

1 INTRODUCTION 1

1.1 Problem Statement 1

1.2 Similar efforts / attempts 2
1.2.1 Beginning inquiry 2
1.2.2 Directing inquiry 3
1.2.3 Drawing conclusions 3
1.2.4 Collaborating with others 3
1.2.5 Acquiring and presenting knowledge 3
1.2.6 Overall context 4
1.2.7 Summary 4

1.3 Thesis Purpose 5

1.4 Some history 7
1.4.1 How this project was born… 7
1.4.2 Results and problems 8
1.4.3 PBL at TECFA 8
1.4.4 Decisions taken 9

1.5 Concrete example 9

1.6 Thesis Organization 11

2 CONCEPTUAL FRAMEWORK 13

2.1 State of the art 13
2.1.1 Constructivism 13
2.1.2 Project-Based Learning 13
2.1.3 Net(worked) PBL and Technology Supported PBL 16
2.1.4 Engagement theory 17
2.1.5 Reflection (Meta-cognitive reasoning) 18
2.1.6 Scaffolding approach 18
2.1.7 Resource-Based Learning 19
2.1.8 "Genex" for GENerating EXcellence" 19

2.2 Synthesis - Conclusion 20

3 RESEARCH AND DEVELOPMENT (R&D) METHOD 23

3.1 Research Method 23
3.1.1 Field exploration 23

3.2 Development method 24
3.2.1 Cooperative Prototyping versus Traditional Prototyping 25
3.2.2 Using Cooperative Prototyping for EVA 26

4 EVA'S XML INFOSET 29

4.1 XML: The New Standard 29
4.1.1 Technical advantages of XML 30
4.1.2 Conceptual advantages of XML 30
4.1.3 XML deficiencies 31
4.1.4 XML initiatives for education 32
4.1.5 DTD vs. XML Schema 34

4.2 EVA's Grammars 35
4.2.1 Conception of EVA's grammars 36
4.2.2 EVA's Grammar Overview for project management (EVA_pm ML) 37
4.2.3 EVA's Grammar Overview for the resulting paper (EVA_pm ML) 38
4.2.4 Support for EVA's grammars 38

5 EVA'S TOOLSET 40

5.1 In Search of A Friendly Editor 40
5.1.1 Text editors 41
5.1.2 Graphical editors 42
5.1.3 Conclusions 44

5.2 EVA's commNcontrol portal 44
5.2.1 Authentication/Customization 45
commNcontrol - Content presentation (Dissemination of the data) 47
5.2.3 Group and global mailing lists 48
5.2.4 Semi-asynchronous persistent in time discussion tool 49
5.2.5 File transfer tools and versioning system 51
5.2.6 Audits and evaluations tools 53
5.2.7 Project management and Paper stylesheets 55

5.3 EVA's Virtual Book 55

5.4 Implementation 56
5.4.1 Java Servlets 56
5.4.2 XML files 57
5.4.3 Technologies used 57
5.4.4 Awareness tools 58
5.4.5 Putting all of them together … 59

6 TESTING AND EVALUATION OF EVA 63

6.1 Testing EVA 63
6.1.1 Choose a project, make groups 63
6.1.2 Presenting EVA 64
6.1.3 Download initial versions 65
6.1.4 Follow phases 65
6.1.5 Editing xml files 66
6.1.6 Coaching 66

6.2 Preliminary findings for EVA 66
6.2.1 Evaluation of the LE 67
6.2.2 Evaluation of the student learning 68

6.3 Evaluation of the student's use of the LE 69
6.3.1 General satisfaction – appreciation 70
6.3.2 Usefulness – short and long term scaffolding 70
6.3.3 Usefulness of the accompanying tools 71

6.3.4 Reflection 71
6.3.5 Managing time 72
6.3.6 Resourcefulness 72
6.3.7 Motivation from peering or coaching 72
6.3.8 Ergonomics - usability 73

6.4 Evaluation of tutor's use of the LE 74
6.4.1 General satisfaction – appreciation 74
6.4.2 Usefulness of the given tools 74
6.4.3 Awareness (Monitoring) 74
6.4.4 Scaffolding / Support 75
6.4.5 Ergonomics / Usability of the interface 75

6.5 Evaluation of the author of EVA_pm 75

7 CONCLUSIONS 76

7.1 Concluding remarks 76

7.2 Outlook 78

8 APPENDIX A: EVA_PM'S GRAMMARS 79

8.1 EVA_pm ML ("project.dtd") 79

8.2 EVA_paper ML ("paper.dtd") 84

8.3 Example file (xsl view) for project management DTD 85

8.4 Example file (xsl view) for paper DTD 85

8.5 EVA's URL's 85

9 APPENDIX B: QUESTIONNAIRE AND STATISTICS 86

9.1 Online Questionnaire for EVA_pm (to STAF-Gina students) 86
9.1.1 Statistics 86

10 APPENDIX C: EVA_PM MANUAL 87

10.1 User Guide for EVA 87
10.1.1 Framework 87
10.1.2 Target public 87
10.1.3 Collaboration 87
10.1.4 Coaching / Support 87
10.1.5 Technical requirements 87
10.1.6 Documentation for the Grammar 88

11 REFERENCES 89

12 TABLE OF FIGURES 94

13 GLOSSARY 95

14 INDEX **96**

To Vassili, for the four years being married that we close today

ABSTRACT

This study is centered on the design and development of a scaffolding environment for students' projects. It's a constructivist environment that aims to scaffold Project-Based Learning (PBL) strategies to improve their effectiveness and serve as a reflection tool to help students develop meta-cognitive skills. It leverages XML (eXtended Markup Language) and suggests a vocabulary (DTD) that describes students' projects (assessments, theses, etc.) as a prompting tool. The vocabulary is well accompanied with the proposal of an appropriate XML editor to diminish the cognitive load of editing and also with an online toolset ("commNcontrol" and "Virtual Book") to scaffold communication, visualization, peering, and progress tracking.

Place of research: TECFA, University of Geneva, Switzerland

Keywords: project, education, XML, thinking tools, scaffolding, Project-Based Learning (PBL), reflection, constructivism, and cognitive tools.

Resumé

Cette recherche est centrée sur la conception et la réalisation d'un environnement d'étayage pour des projets d'étudiants. C'est un environnement constructiviste qui a pour but de favoriser des stratégies tel que l'"Apprentissage par projet" (PBL) afin d'améliorer leur efficacité. De plus, il est utilisable comme un outil de réflexion pour que les étudiants développent des compétences méta-cognitives. Ce projet exploite la technologie XML et propose un vocabulaire (DTD) qui décrit les projets des étudiants (mémoires, thèses, etc.) comme un outil de "prompt". Nous proposons un éditeur XML (choisi parmi plusieurs) pour diminuer la charge cognitive induit l'édition en XML. De surcroît, nous proposons un outil en ligne ("commNcontrol" et "Virtual Book") comme support à la communication, à la visualisation et au suivi de l'avancement du projet.

Lieu de recherche: TECFA, Université de Genève, Suisse

Mots clés: projet, éducation, xml, thinking tools, étayage, Apprentissage Par Projet (APP), réflexion, constructivisme, outils cognitifs.

ACKNOWLEDGEMENTS

I would like to thank all students of STAF-Gina that tested EVA and for all the fruitful comments that made: René, Lionel, Dierk, Roberto, David, Catherine, Yves, Nathalie, Nicolas, François, Elodie and Sylvain.

Christine Diehl for reading and correcting English of this thesis. Thanks Christine, it meant a lot to me.

Bertrand Resin for helping me with French, for being the first that tested EVA, but most of all for his humor.

Anne Marquis for helping me with French and for the XML books that brought me from New York.

All my colleagues at TECFA. Especially Allison, Barbara, Bertrand, Cassandra, David, Didier, Fabrice, Olivier, Véronique. Each one in his own way, added something to this thesis.

TECFA that provided the means and space for this thesis to take place and my professors, Patrick Mendelsohn, Daniel Schneider, Pierre Dillenbourg and Daniel Peraya for their valuable contribution to this thesis.

Dr. Gilles Falquet for accepting to be part of the jury of this thesis and for the exact reading of it.

Dr. Patrick Mendelsohn for believing in me in a period that my moral was very low, for being the director of this thesis.

Dr. Daniel Schneider the person that gave birth to this method and for trusting it to me, the co-director of this thesis. For all the things that I learned next to him. For making working with him so much fun. I really appreciate you, it was a privilege to get to know you and work with you.

To the dark blue of Santorini, that helped me escape in the difficult times of this thesis by looking to the two postcards I have in my office and Vassilis that first offered me a trip there.

My family, my mother for all her support and the hundreds of phone calls, my father for paying the huge telephone bills, and my brother for being always there where I needed him. I miss you all.

Vassili, my husband for thousands of things… for explaining me so many times what "static" means in Java, for providing all the infrastructure ☺ to work on this thesis (portable computers, PC's, networks, etc…), for the script that he wrote that processed the data from the questionnaire, for all the nasty comments that made me become better, for all the endless nights that we shared the same desk working next to each other with our computers to give me courage to go on, for just being part of my life. I simply love you…

Chapter

1

1 Introduction

This first chapter describes the goal and the objectives of this research followed by a short presentation of all the chapters, giving the global articulation of this thesis.

1.1 Problem Statement

"Project-based learning" (PBL)[1] is a model that organizes learning around projects. According to different definitions cited by Thomas J., 2000 [64] projects are complex tasks based on challenging questions or problems that involve students in design, problem-solving, decision making or investigative activities; give the students the opportunity to work autonomously; and culminate in realistic products or presentations (Jones, Rasmussen & Moffitt, 1997 [27]; Thomas, Mergendoller & Michaelson, 1999 [63]) .

Some advantages of using PBL approaches, among others, are that PBL increases student motivation by engaging them in their own learning, it improves student problem-solving and higher order thinking skills, promotes metacognition and self-regulated self-learning by asking them generate their own strategies, and it provides opportunities for interdisciplinary learning.

"Project-Based Learning" is gaining place against other traditional methods of instruction and there is a strong belief that is beneficial and effective (in Conclusions of Thomas J., 2000 [64]). It's implementation, though, falls to several student deficiencies and PBL design problems.

There are several challenges faced by students and teachers during PBL implementation. In particular, students have difficulties benefiting from self-directed situations especially in complex projects. Some of the most important problems that they face are:

- **Initiating inquiry;** students have difficulties to come up with a coherent research design, define goals and objectives, break up the work into pieces. They have difficulty framing questions to guide their inquiry and in particular, developing questions that have scientific merit.
- **Directing investigations;** students need guidance locating the required resources. Lack of data and insufficient resources are some of the reasons that a project fails.
- **Managing time;** students often fail to keep up with deadlines. The most common case is that they underestimate the time that is needed for the realization of a task.
- **Overcome "writer's block";** students often face "writer's block" without being able to overcome it. Especially in the beginning, they are overwhelmed by the complexity of the task instead of breaking up their work into simpler parts.

[1] From now on, in this text "PBL" will stand for "Project-Based Learning"

- **Lack or limited experience on project design and generally on research skills;** the majority of the students have never being taught project management and methodology. So planning a project, organizing the co-operative work (in case it is needed), and distributing the tasks is a difficult task *per se*. The students don't break tasks into parts, they don't "debug" their work.

- **Collaboration & feedback;** students have difficulty on collaborating and articulating their work with that of others. They also are not very strong in giving regular feedback on their work.

- **Project follow-up;** acquiring critical thinking skills and cognitive self-awareness that support learning is difficult according to Schneiderman et al., 1998 [56] and students rarely revise their products.

Teachers on the other hand:

- have **limited prior experience** with PBL practice,

- have difficulty **monitoring if knowledge is being acquired** and deciding if and when to "coach",

- let alone the fact that **monitoring** several projects in parallel, to provide feedback and eventually evaluate them, is burden-some and time-consuming with traditional methods.

Because of all the above, important research has been conducted to improve the delivery or effectiveness of PBL by intervening in the practice of it. There are interventions that are designed to correct the weakness of some PBL feature or to remediate or accommodate some student deficiency relative to an aspect of project work. These interventions, which are designed to support PBL, have been referred to as **scaffolding** (Guzdial, 1998 [18]) or **"procedural-facilitation"** (Scardamalia, Bereiter, McLean, Swallow & Woodruff, 1989 [52]).

Scaffolding is a term used in education research to describe the kind of support provided to a student engaged in learning a process, particularly a design or problem-solving process (Wood, Bruner & Ross, 1975) that can fade progressively and disappear by the end of the process.

Procedural facilitation involves prompting students about their role in the collaboration and suggesting useful things to say.

In literature, there is a range of strategies for such interventions. Some of these attempts include the development of specific software that will facilitate the task for both students and teachers and make it more efficient.

1.2 Similar efforts / attempts

Several attempts have been made from many researchers, to improve the effectiveness of PBL. For the purpose of this research, a representative sample has been studied and is presented according to the deficiency that they try to remediate.

1.2.1 Beginning inquiry

Scardamalia and Bereiter (1991) [49]; Scardamalia, Bereiter, McLean, Swallow & Woodruff (1989) [52] and Sage (1996) [47], trying to help students initiate their inquiry,

developed **CSILE** (Computer Supported Learning Environment) [11], a student-constructed, collective database. The system incorporates "cognitive coaching" e.g. the use of steps to guide beginning inquiry, with peer of teacher feedback in order to make knowledge construction activities overt.

Barron et al., (1998) [1] in their intervention "**Learning Appropriate Goals**", tried to correct the students' deficiency of initiating their inquiry by introducing specifications, asking for design plans and generally by helping students to develop "driving questions".

1.2.2 Directing inquiry

Barron et al. (1998) [1] also provided students with practice in conducting problem-based learning activities prior to introducing PBL and embedded guidance on how to conduct an operation within the project materials.

Torp and Sage (1998) [65], in order to help direct student inquiry provided a structured set of inquiry steps for students to follow.

1.2.3 Drawing conclusions

Cognitive and Technology Group (1991) at Vanderbilt University [5], realizing that students have difficulty in constructing mental models to guide problem-solving, developed computer programs to help them construct and visualize ideas.

1.2.4 Collaborating with others

Guzdial (1998) [18], to help students better collaborate with each other, incorporated a computer-mediated "cognitive apprenticeship" model.

Hmelo, Guzdial & Turns (1998) [20] developed **CaMILE** , a "Collaborative and multimedia interactive learning environment".

Later on, (Guzdial and Turns, 1999) tried to encourage better team planning through use of a web-based tool "**Team Facilitator**" by prompting students to evaluate their performance individually and in groups and by present their planning and their strategies for the next phase. In addition, they developed "**Reflective Learner**", a web-based environment that supports students in writing learning essays to reflect on their design experiences. The main support of the system is the scaffolded writing area.

1.2.5 Acquiring and presenting knowledge

For the acquisition of knowledge, Blumenfeld et al. (1992) [2] intervened by asking students for explanations and justifications at different stages of the project. They emphasized learning vs. work completion and understanding vs. product quality goals for student work.

Barron et al. (1998) [1] helped teachers to better monitor projects and improve their "coaching", by incorporating "formative self-assessments", by creating a classroom culture that supports frequent feedback and assessment, and by finding ways for students to compare their work with others. They also made students take more seriously their work by incorporating opportunities that involve external audiences.

Klein et al. (1997) [30] required multiple criterion performances (e.g., collaboration, explanation, demonstration, self-report) to make students focus on their work and promote meta-cognition.

1.2.6 Overall context

"Project-Based Science (PBS)" [39] (http://www.umich.edu/~pbsgroup/) has as it's goal to improve the way science classes are taught by involving students in finding solutions to authentic questions through extended inquiry, collaboration and use of technology. PIViT [37] (Project Integration and Visualization Tool) (http://www.umich.edu/~pbsgroup/PIViT.html) is a project-planning software tool that has been developed in the framework of PBS project at the University of Michigan.

Project MOST [38] (http://tiger.coe.missouri.edu/~most/), (Think-ers, Tink-ers, Tech-ers) has as central goal for the students to learn how to do research, not as a process that you carry out in a sterile and methodical manner but rather as a challenging set of activities and problems that engage them intellectually, socially and viscerally in a meaningful pursuit. It prepares them to be competent and motivated to undertake project-based learning and computational science.

The "Project-Based Learning Support System (**PBLSS)"** [36], (http://tiger.coe.missouri.edu/~most), developed by the University of Missouri in Columbia has as it's objective to provide scaffolding and process support to students who are engaged in the study of math and science. The pedagogical support, scaffolding or coaching processes are featured along with the following four learning processes: the planning and pinpointing of resources, the mental representation of the knowledge, communication, collaboration and reflection (Laffey, 1998, pp. 73 & 75 [31]).

The "**Design Diary**" [42] (Puntambekar S., et al. 1998 [42], website http://www.cc.gatech.edu/edutech/projects/dd_icls98/icls_design_diaries.html), is a tool to support students in learning science by design. This scaffolding tool has prompts to help students carry out their design related activities and reflecting on them in order to learn. Four stages of the process are supported: analysis or problem understanding, exploration, solution generation and evaluation.

"Interactive Web-Based Learning Environment" **WBILE** [43], (Sadhana Puntambekar, University of Connecticut), uses a constructivistic problem-based approach. It provides tools to support individual and collaborative learning such as: **reflective notebooks** that will help students flesh out their ideas before discussing them with other students, **discussion tools** for small and large group discussions and **collaborative concept maps** for students to integrate their knowledge and see the connections between topics.

1.2.7 Summary

All the above interventions to improve PBL efficiency concentrate in a specific area (problem), which they try to remediate.

Despite the differences of each approach, it is interesting to focus on their similarities. First of all, most of them **use technology** to reach their goal. Second, there are several key terms that come up very often in most of them, e.g. **scaffolding**, **reflection**, **collaboration**, **constructivism**, **and meta-cognition** (cf. Chapter 2).

1.3 Thesis Purpose

This study is also an intervention for improving PBL efficiency. It entails the development of a Scaffolding Learning Environment (SLE[1]) that is trying to learn from the lessons of the past and leverage from stresses on new technologies like XML[2] and the World Wide Web making a lightweight and easily portable environment.

Most of the research for improving PBL efficiency tries to remediate specific weaknesses of PBL, but doesn't propose a complete system that supports a substantial student project through all it's phases and for all contexts.

Our key goal was to develop a constructivist environment and a method for scaffolding students' projects (assignments) from their management up to the writing of their final report.

So, the objectives of this SLE are:

to help students **develop scientific inquiry** and knowledge integration skills, to focus on important and investigate key issues;

to support them **directing investigations**;

to make students **better manage the time** and respect the time constraints;

to overcome possible **writer's block,** or even better to avoid it;

to help students acquire **knowledge on project design** and research skills;

to improve **team management** and collaboration (especially collaborative editing of student groups);

to make students **reflect on their work**;

to support the tutor's role in a PBL approach;

to facilitate monitoring and evaluation for the tutor;

to help the tutor **verify whether knowledge is being acquired**;

to **motivate the peers**, and eventually to **distribute** the results **to bigger audiences**.

In order to succeed in the above i.e., to scaffold and manage a PBL process, this environment has planning tools, communication and collaboration tools, knowledge presentation ways and resource deposits that can guide students and help to develop meta-cognitive skills in them through reflection.

Our proposal is mostly influenced by the "**Reflective Learner**" of *Guzdial's team* and the "**Design Diary**", "**Reflective notebooks**" and "**Reflective essays**" of *Puntambekar's team*.

We consider the online distribution of this environment as a special feature that we propose, which makes the task easily accessible from everywhere, i.e. less time-consuming, more cost-effective and definitely more collaborative.

[1] "SLE" from now on in this text will refer to "Scaffolding Learning Environment"
[2] XML stands for "eXtensible Markup Language" (see Glossary for the complete definition)

With this research, we target students that have none or little experience in how to design and process a project. Thus, it seems inevitable that we must find a way to teach them how they should work efficiently to achieve their task and to get the maximum profit of the Project-Based Learning method (better and faster implementations with a coherent research design).

More specifically, given the fact that the specific technology that we propose also entails the cognitive load of getting familiar with the environment, we limit our target public to high school and university students. Moreover, according to Jonassen et al. [26]

> "… constructivistic learning environments are most effective for an advanced knowledge acquisition stage of learning… Therefore, universities are among the most appropriate venues for implementing constructivistic learning…".

So, the academic projects can very well be projects inside the high school *curriculum* as well as term projects inside university courses, and to some extent also research projects like Master theses, depending on the level of their complexity.

The proposed environment that has been implemented for the purpose of this thesis is called **EVA_pm**[1] and consists of the following:

- The conception of a **grammar** for a markup language that models the semantics of an entire academic project like setting goals, breaking complex goals down into achievable objectives and planning for them (for scaffolding and coaching), and project management (including tracking, evaluation and feedback) for planning, communication, reflection and resourcefulness.

- The development of a **toolset** that complements the above language, including:

  the proposal of a suitable (for our goal) XML editor,

  an online monitoring and communication console (**commNcontrol**) for both students and their tutors, which distribute project information, facilitates the task of monitoring, evaluating and feedback, and enables team members and project directors to communicate and has a versioning system that keeps track of all the versions to facilitate the group work,

  and several stylesheets for visualization of the given information in beneficial ways for each of the interested parties (project authors, tutors, peers, etc.).

- The conception of a simplified **grammar** for a markup language that models the final result(s) of a project by structuring the project's product in terms of a semi-formal paper (i.e., having an abstract section, a main section, a conclusions section etc.). The main advantages of such presentation is that simplifies the dissemination of the results of many projects at the same time and it is easy to search and easy to extract parts of it, e.g., making a list of all the abstracts of a group of projects and creating a web portal, which in our case is called **Virtual Book**.

[1] The name EVA was taken after the name of the promotion of the STAF diploma that the author of this thesis has been part of and *pm* stands for "Project Management". There is a tradition for STAF diploma to give a feminine name after every promotion that should start with the corresponding letter of the alphabet. So, the promotion of the author was the fifth one, that's why it starts with the leltter E and the name that was chosen by TECFA was EVA. From now on in this thesis, EVA will stand for the method and the environment implemented here.

It is also important to note at this point, that using an xml file to write the final paper makes students focus on the content. It happens very often the students to put too much of their attention on making their text look beautiful rather than reflecting on it. XML can diminish this phenomenon by giving a stylesheet to the students that will take care of the formatting.

1.4 Some history

1.4.1 How this project was born...

This attempt has been strongly inspired by the method used at TECFA by Daniel K. Schneider, the co-director of this thesis for the management of the project STAF-18, to support the students during the realization of their projects.

D. K. Schneider is in charge of a course called STAF-18 in the framework of an innovative two-year post-graduate diploma (MSc) in Educational Technologies at TECFA, University of Geneva, Switzerland called STAF. STAF-18 is strongly based on PBL as an instructional approach. During this course, students have to choose among a variety of given projects and work alone or in groups (pairs are strongly advised).

Two years ago, a grammar (DTD) was introduced to the students to describe their projects in XML files in order to support the students during the realization of their projects and later on, to parse the data in such a way that would be able to deduct precious information for the progress of each project. This last one was the first project's "cockpit" made with a stylesheet (XSLT).

The writer of this thesis participated and experienced this very first attempt as a student, gaining precious information and eventually fascinated by this method. She even implemented an online tool (with PHP3) in the framework of her project, to facilitate the editing of the XML files through web forms. The reason for this was that XML was (and still is?) an emerging technology, and therefore, was lacking editing tools, sufficient enough to sustain an educational activity.

The second year, the writer had the chance to be actively involved in the conception of the course STAF-18, as a teaching & research assistant for STAF diploma. So, summarizing all the problems of the previous year, an effort was made to improve the grammar by adding elements that were missing or erasing elements that were confusing the students. There was also an opportunity to experiment with new techniques such as Cocoon (The Apache XML Project). As some XML technology was still changing and editors were always under continuous construction, this time XEmacs in PSGML mode was chosen as the editor in favor. The second project's "cockpit" that was made for this year, was in Java Servlets (parsing XML with SAX/DOM API), with more functionality and definitely with a more appealing presentation.

An extra characteristic of this year's projects was that students were also asked to write a report for their project in the format of a semi-formal scientific paper in an XML file. For this reason, a specific grammar (report.dtd) was given to the students for structuring their report into sections like abstract, preface, introduction, main body, conclusions and annex. For the formatting of the text inside of these sections, we included in our grammar a DTD called "**ibtwsh6.dtd**" by John Cowan that had most of the formatting possibilities of an HTML document (to be more precise, it was actually a subset of XHTML). This grammar was accompanied by a stylesheet that takes care of the presentation of students' papers through the web (in HTML) in a similar way. This

choice was also an effort to provide a prompting tool to the students, making them realize how they should structure such kind of documents and at the same time shifting their attention from the “decoration” of their paper to focus on the semantics of it.

1.4.2 Results and problems

The positive results, that came (though without an experimentation and measurement) from these 2 years of using such a method, are that students finished their projects in time (compared to before, that they had a tendency to finish their projects the following period), they reported that they now had a better idea of how they should organize their work for a project, and definitely student management monitoring of the project was much easier for tutors than before.

The problems that were reported and that we noticed were that editing was difficult (need of basic XML knowledge, and lack of user-friendly and error-free XML editors). This was an important problem, as the cognitive load for editing was so high that students didn’t put all their attention to the content, as they had to achieve also editing.

Another problem was collaborative editing. As some of the students were working in groups, and eventually had to edit the xml file, there was a problem of sharing this file. In the same way, as tutors were giving audits and the final evaluation through the same file, they also had to share the same file at some point.

Communication also was made with simple emails or with a separate forum, which resulted in all this valuable information being stocked separately in a forum list or spread along several emails without having the possibility to make references to specific elements of the projects.

An important feature that was also missing is that there was no tracking of different versions of the xml files. As students were changing the content of their project management file according to tutors audits and their natural evolution, there was no way to keep track of these changes unless copies were kept from time to time in a separate folder. This solution was not so sophisticated, though, as these copies could correspond only to time and not to the real changes that each group was making.

1.4.3 PBL at TECFA

At TECFA, PBL is one of the main methods used for STAF diploma. So, the rest of the professors also have created their own methods to support students during their projects. Such attempts are:

- “Student management” for STAF-15 of Prof. P. Mendelsohn,
- Grille d’analyse” for STAF-13 of Prof. D. Peraya, and
- “The collaborative studio” for STAF-16 of Prof. P. Dillenbourg (access through guest account from Virtual Campus of TECFA http://tecfa.unige.ch/campus/infospace/index.php).

Each method has it’s own advantages but also it’s own particular characteristics as they have been conceived for a specific course. Therefore, they cannot be easily generalized and most important they are not portable. Given the fact that XML technology is portable and complying with the stress for "open soft" philosophy in education, our method fitted well to the picture.

1.4.4 Decisions taken

So, we decided that after an experimentation of two years, we were ready to make a complete and improved method and to share our toolset with the rest of the educational community around the world. The goal was that it could be easily applied and adopted from other tutors for different projects, but at the same time be flexible enough to leave space for improvisation for both students and tutors.

That means, that we had to start all over from the beginning and redefine our goals and more importantly go over the literature that could support and contribute to our project. We hope that this attempt will be well accepted and we encourage and welcome all comments and suggestions that will improve our work.

1.5 Concrete example

For a better understanding of our project, we briefly present a typical use of our proposed environment by describing how it is used inside the university post-graduate course called STAF-18: "Teaching and learning in a virtual space". The course uses the Project-Based Learning as teaching method. Every year, approximately 15 students follow it.

The general objective of STAF-18 is to present theoretical, practical and methodological bases of Computer Mediated Communication (CMC) and more specifically, of virtual environments and GroupWare. Every year a different subject area is chosen. In our example (taken from STAF-Gina promotion 2000-2001), the theme chosen is "*What can one learn from gaming environments?*". The previous year's theme was "*Information Visualization*".

The interest is focused on several aspects of gaming environments like: motivation, ergonomics, communication tools, introductory tutorials, on-line help, game editors, etc. The most important research questions that arise are the following two:

(1) What lessons can we take from games in order to improve learning environments?

(2) Can we learn by playing? If yes, in which domains?

STAF-18 in the curriculum is called a project course. It includes a 4-5 half days of classroom teaching in the initial week and students have six (6) weeks to complete their project working independently.

The course is happening in the following way:

1) Course propaganda. In the teaching period before the course officially starts (six weeks before), the theme of STAF-18 is announced and a short introduction is made to it. To stimulate students on the subject, a group discussion follows and they are asked to think about what they would prefer to work on in the meantime until the course starts.

2) Presentation of topics. In the beginning of the course, the tutor presents the detailed theme and discusses the research questions that have to be addressed. Later on, gaming environments are presented, why they were chosen and what are their specific characteristics. A detailed list of suggested projects is described and discussed together with research questions that should be addressed and some important resources to start the research.

3) **Presentation and discussion of Project management.** In the next course session the tutor makes an introduction on how a research plan should be done (focused on project management). Distribution of games, projects and creation of groups follows immediately after.

4) **Presentation of tools that will be used.** At this point, the toolset that is going to be used for the course is presented (for year 2001, is the environment presented in this thesis). It starts with the presentation of the conceptual framework that supports it (PBL, scaffolding, reflection, etc.), and continues with a concrete presentation of the tasks to be done and technical details.

5) **Students make a draft specification of their project.** Students actually do their project management by editing an XML file, where they define goals, objectives, etc. This is a difficult task as has been already described in the introductory section of this thesis. That's why XML has been chosen to facilitate and scaffold the task by serving as a scaffolding and cognitive tool (explanations are given further in this study). The tutor gives audits to regulate students' work.

6) **Students present the final specification for their project.** Two weeks later, students have to present their final specification of their project (through the same xml file). The tutor again, here, gives audits to regulate students' work.

7) **Students work on their project.** According to the work packages that they have defined in their specification, they execute the project. The tutor gives audits to regulate students' work.

8) **Students write the results of their project.** The students have to edit a second XML file where they describe their results in a form of a semi-formal paper (preface, introduction, body, conclusions, annexes). At the same time, students reflect on their work by editing comments inside the project management XML file.

9) **Final evaluation.** The tutor evaluates the projects and gives the final feedback for the projects.

All the above are supported by a web portal that provides several tools that make the task easier and that help to overcome several problems that otherwise would have been obstacles. The graphical representation that follows shows the evolution of an XML file through our environment (cf. Figure 1.).

Figure 1. This image shows how the XML file that a student writes can be transformed through EVA.

1.6 Thesis Organization

Several chapters articulate the presentation of this study:

Chapter 1 In the beginning, the arguments for the importance of such a study is presented along with a short description of similar attempts.

Chapter 2 Later, the conceptual framework that this research is based on and that supports this thesis is described in detail.

Chapter 3 In this chapter, the research and development method is presented. In the research method, the field exploration is described together with the small survey research that has been conducted between several people at TECFA, which serves as preliminary research. It's here, where the analysis of the questionnaire results is taking place and the decision for the development method follows. Here it is explained why and how we tried to use cooperative prototyping in the implementation of EVA_pm.

Chapter 4 EVA's XML infoset is presented here. It starts with an introduction to XML, it's advantages and it's deficiencies (so far), and how it can become a cognitive tool. It continues with the review of the most important attempts to leverage XML for educational uses. Immediately after, comes the description of the Phase I of the implementation for this thesis: the conception of the grammars and their description.

Chapter 5 In this chapter, comes the description of the implementation phase II (EVA's toolset) that complements and supports the grammars, and refers to the monitoring console ("commNcontrol"), the "Virtual Book" and the research for a user-friendly and configurable XML editor.

Chapter 6 As a logical sequence, the way that this environment has been tested and evaluated is given in this chapter. The evaluation method and it's criteria are described. Projects of STAF-18 students were used as test cases to evaluate the implemented parts and to verify that the implemented grammar will be able to model a project delivered online and that they work as specified.

Chapter 7 Results and conclusions, on whether the initial goals have been reached or not, follow. And the chapter ends by giving an outlook of the perspectives of the method.

Appendix A The grammars that are conceived for EVA_pm, links to examples and documentation online are presented.

Appendix B The questionnaire for staf-Gina students and the descriptive statistics of the answers to it are given here.

Appendix C A user guide for EVA_pm in case that someone is interested to study further or reuse the method is attached.

References References (in alphabetical order) on which this thesis was based, by extracting precious information, is the last but not least part of this presentation, with the infinite gratitude of the writer.

Figures The figures presented inside this thesis are given here in the form of a table where their captions and the pages that they appear are given.

Glossary A small glossary is given to the reader, with a short definition of the most important terms that appear in this thesis and put in alphabetical order.

Index The index of all the keywords (in alphabetical order) and where they appear inside the text.

Chapter

2

2 Conceptual Framework

This chapter presents a review of the state of the art that inspired this thesis and concludes with a synthesis of these theories and how they have been used to promote the thesis goal.

2.1 State of the art

The challenge of increasing students' understanding cannot be met only by giving them more "hands-on" activities. Students are amazingly gifted at completing the task successfully while avoiding having to actually learn (Doyle, 1983; Ng & Bereiter, 1995), let alone the fact that it is difficult to monitor students throughout the accomplishment of their task. So, it is very difficult to tell whether a student has actually learned and what exactly has been learned. Moreover, in a PBL approach, it's not the assignment of a project that makes the difference in learning, but the way it is assessed.

Such activities are effective as long as they are embedded in a "minds-on" or "mindtool" learning environment (Glynn, Yeany and Britton, 1991; Mayes 1992 [33]), where students are encouraged and helped to develop deep understanding and the skills of problem solving and reflective thinking (Rakaw, 1985). Puntambekar (1999) [43] cites many articles to conclude that by integrating minds-on activities into a curriculum we can succeed in creating such an environment.

The goal of our research is to help students learn the content of their curriculum more deeply by conducting a project and at the same time, develop skills and understanding of how they should conduct a project. The conceptual framework for the design and implementation of our "minds-on" learning environment is guided by the following approaches:

2.1.1 Constructivism

In general, the approach that is being adopted is a constructivist approach. Constructivists believe that learning is constructing knowledge from one's experiences rather than directly receiving information from the outside world (e.g. Collins & Green, 1992; Resnick, 1987; Brown, Collins & Duguid, 1989 [4]; Collins, Brown & Newman, 1989).

2.1.2 Project-Based Learning

Project Based Learning (PBL) is a teaching and learning model (curriculum development and instructional approach) that shifts away from traditional teacher-centered teaching and emphasizes student-centered instruction by assigning projects. It allows students to work autonomously to construct their own learning, and culminates in realistic, student-generated products.

More specifically, project-based learning can be defined as (The Challenge 2000 Multimedia Project website [60]):

- Engaging learning experiences that involve students in complex, real-world **projects** through which they develop and apply skills and knowledge

- Learning that requires students to draw from many information sources and disciplines in order to solve problems

- Learning in which curricular outcomes can be identified up-front, but in which the outcomes of the student's learning process are neither predetermined nor fully predictable

- Experiences through which students learn to manage and allocate resources such as time and materials.

Project-based learning is a structure that transforms teaching from "teachers telling" to "students doing." Teachers assume the role of cognitive and meta-cognitive coach (asking, monitoring, probing, managing, keeping moving) rather than knowledge-holder and disseminator; students assume the role of active problem-solvers, decision-makers, and meaning-makers (by participating, being engaged and by constructing) rather than passive listeners; and a project serves as the initial challenge and motivation (appealing to be explored, setting up the context of learning).

Researchers have investigated the impact of project-based learning (PBL) in a wide variety of educational contexts ranging from early childhood education to medical and legal education. PBL (and related instructional approaches) has generally been shown to be effective in increasing student motivation by engaging them in their own learning and in improving student problem-solving and higher order thinking skills (Regie Stites, 1998 [59]).

Also, PBL promotes meta-cognition and self-regulated learning by asking students to generate their own strategies for problem definition, information gathering, data-analysis, and hypothesis-building and testing, comparing these strategies against and sharing them with other students' and mentors' strategies.

> "The collaborative nature of the investigation enhances all of these valuable experiences ... as well as promotes a greater appreciation for social responsibility (Scott, 1994)."

Teaching with the project-based method enables students to work cooperatively with peers and mentors in a student-centered environment where learners are encouraged to explore various topics of interest. Hence, PBL also provides opportunities for interdisciplinary learning by engaging students in applying the content of different subject areas during the various phases of the project.

PBL helps students develop real world skills like the ability to collaborate well with others, make decisions and take initiative, and face complex problems.

After completing a project, if students are asked to create a self-evaluation of the project, like writing a meta-report, this enables the students to focus on their learning process and allows them to see their progress. Self-evaluation gives students a sense of accomplishment and further instills responsibility for learning. And by documenting the learning process it also makes it easy to distribute results to bigger audiences, with all the obvious advantages.

Generally, with it's innovative approach to learning, PBL also requires an innovative approach to assessment. It requires varied and frequent assessment, including teacher assessment, peer assessment, self-assessment and reflection.

Although there is a lot of research that has been done on the impact of PBL, it is difficult to isolate the effects of it. Some of the reasons are, that it is applied most of the time with other strategies, limits with other similar approaches are blurred (like problem-based learning), it is implemented differently in different contexts and most of all, standard achievement tests cannot measure the higher order thinking skills developed (Regie Stites, 1998 [59]).

Thomas J., 2000 [64], in his review of research on PBL concludes to the following:

- There is evidence that PBL is challenging to plan and enact.
- There is some evidence that students have difficulties with self-directed situations, especially in complex problems. Some of these difficulties associated with initiating inquiry, directing investigations, managing time and using technology productively (already described in §1.1). Thus, the effectiveness of PBL depends a lot on the scaffolding provided to students to learn how to learn.
- There is indirect and direct evidence that PBL is a more popular method than other instructional strategies, as both students and teachers believe that is beneficial and effective.
- There is some evidence that PBL compared to other learning methods, enhances the quality of students' learning, increasing their capability for applying what is learned in novel problems.
- Finally, there is ample evidence that PBL is effective for teaching students complex processes like planning, communicating, problem solving and decision-making, but there is no comparison with other methods.

We have to clarify here, that with the abbreviation of PBL, we refer to **Project-Based Learning** and not to **Problem-Based Learning**. They both describe constructivist instructional strategies, that engage students in authentic student-centered tasks to enhance learning, and the line between them is frequently blurred, as they can be used in combination and play complementary roles, but they are not identical approaches (Camille Esch, 1998 cited in The Challenge 2000 Multimedia Project website [61]).

The project-based learning uses a **production model**. Students start by defining the purpose of creating the end-product, identify their audience, they research the topic, design the product, do the project management, solve the problems that arise and finish the product followed by a self-evaluation and reflection (Crawford, Bellnet website, Autodesk website, Blumenfeld et al. cited in The Challenge 2000 Multimedia Project website [61]). So, the driving force is the end-product, but the key to success is the skills acquired during it's production.

The problem-based learning uses an **inquiry model.** Students start with a given problem, make a plan for gathering information, pose new questions and summarize their research by presenting their conclusions (Duch, Delisle, Hoffman and Ritchie, Stepian and Gallacher cited in The Challenge 2000 Multimedia Project website [61]). In this case, the driving force is the problem given and the success is the solution of it.

2.1.3 Net(worked) PBL and Technology Supported PBL

Definitions of “Project-based Instruction” include also features (among others) relating to the use of cognitive (technology-based) tools (Krajcik, Blumenfeld, Marx & Soloway, 1994; Marx, Blumenfeld, Krajcik, Blunk, Crawford, Kelly & Meyer, 1994).

PBL is especially effective when supported by educational technology (Blumenfeld et al., 1991 [2]; Means & Olson, 1997; Coley, Cradler, & Engel, 1996). Evaluations of K-12 instruction have shown strong evidence of learning gains associated with PBL plus technology (Ryser, Beeler, McKenzie, 1995; Cognition and Technology Group at Vanderbilt, 1992 [5]; Pellegrino et al., 1992).

In one of the best documented programs combining PBL and technology, eighth graders in the Union City (New Jersey) Interactive Multimedia Education Trial scored approximately 10% higher than students from other urban and special needs districts on statewide assessments of reading, mathematics, and writing achievement (Education Development Center, 1994).

Kehoe, Guzdial & Turns (1998) [29] also clearly state that “*… technology can play an important role in structuring and supporting effective project-based learning …*” after long experience with “**Team Facilitator**” (team planning through web), **CaMILE** (Collaborative and Multimedia Interactive Learning Environment), **Reflective Learner** (web environment to learn writing essays) and **STABLE** (SmallTalk Apprenticeship-Based Learning Environment).

Jonassen et al., [26] also state in their manifesto that tools that meaningfully engage the learners should support constructivist environments and that technology-based environments can effectively support these activities.

Technology-based learning environments can and should support advanced knowledge acquisition. And that can be done by providing environments and **thinking tools** that engage constructivistic conceptions of learning (Kommers, Jonassen, & Mayes, 1992). Thinking tools are technology systems or applications that extend the intellectual functionality of the learner by engaging the learner to tasks that facilitate knowledge construction (e.g. semantic network software, expert systems, databases and microworlds).

The Internet also adds critical and valuable dimensions to a PBL experience. The management issues, for a start, that a teacher can face as the classroom-learning manager of a networked project can be dealt with easier than the ones on an "old-fashioned, low-tech" project (in case that teachers cannot meet the students frequently). Another advantage is that the dissemination of the projects is open to bigger and different audiences. More specifically, students have the opportunity to peer, review and browse other similar projects, motivating them and supporting them in the accomplishment of their own project.

And most important, the web can be used as a communications and collaboration medium to build ongoing dialogs between the project authors and their audience, especially their teachers. These “author-mentor” dialogs can be planned and organized to motivate students and establish increasingly high standards.

A networked project typically involves students in distant locations cooperating to research, exchange information, and learn from one another, although the distant partners may include experts. Students may conduct research, perform experiments

in their own community, and report their findings. They may pose questions to experts or exchange information with their peers.

But apart from the above advantages of a networked project, there is always the disadvantage of it's distant nature. So, in order for such a project to succeed, the teachers that conduct it have to put special attention on details such as meeting deadlines and using reminders of pending deadlines, animating the project through all it's phases, and finally, being ready to help and being flexible. While for students concluding such a project, it is important to share Project Results and stay in touch.

2.1.4 Engagement theory

The fundamental idea underlying engagement theory is that students must be meaningfully engaged in learning activities through interaction with others and worthwhile tasks. While in principle, such engagement could occur without the use of technology, technology can facilitate engagement in ways that are difficult to achieve otherwise. So engagement theory is intended to be a conceptual framework for technology-based learning and teaching (Kearsley & Shneiderman, 1999 [28]).

By engaged learning, we mean that all student activities involve active cognitive processes such as creating, problem solving, reasoning, decision-making, and evaluation. In addition, students are intrinsically motivated to learn due to the meaningful nature of the learning environment and activities.

Engagement theory is based upon the idea of creating successful collaborative teams that work on ambitious projects that are meaningful to someone outside the classroom. These three components, summarized by **Relate-Create-Donate** (Shneiderman B., 1988 [55]), imply that learning activities:

1. occur in a group context (i.e., collaborative teams)

2. are project-based

3. have an outside (authentic) focus

Engagement theory is presented as a model for learning in technology-based environments that synthesizes many elements from past theories of learning. The major premise is that students must be engaged in their course work in order for effective learning to occur. The theory posits three primary means to accomplish engagement:

(1) an emphasis on collaborative efforts

(2) project-based assignments, and

(3) non-academic focus.

It is suggested that these three methods result in learning that is creative, meaningful, and authentic.

The role of technology in the theory is to facilitate all aspects of engagement. The use of email, online conferencing, web databases, groupware, and audio-videoconferencing significantly increases the extent and ease of interaction amongst all participants, as well as access to information. The vast array of software tools available for analysis, design, planning, problem solving and making presentations

enable students to do sophisticated and complex tasks. Technology provides an electronic learning milieu that fosters the kind of creativity and communication needed to nourish engagement.

2.1.5 Reflection (Meta-cognitive reasoning)

Knowledge representation does not automatically engage students in looking back over their work and making critical appraisals of the work relative to their own internal cognitive model of expertise (Norman, 1993 cited by Laffey, 1998 [31])

Reflection is the procedure whereby learners are encouraged to review their performance critically and non-punitively. The key to reflective learning is to become a powerful observer of your own thinking, feeling, and acting. Some researchers call it also **meta-cognitive reasoning**.

This kind of “reification” (Collins, 1996) or “abstracted replay” (Collins & Brown, 1988 [6]) enables a learner to reflect in a systematic way on the problem-solving process and to reach new conclusions or make elaborations that they might otherwise not have made (Laffey, 1998 [31]).

Reflection is very important to enhance learning, especially during a PBL approach. Pirolli and Recker (1994) (cited by Puntambekar, 1999 [43]) suggested that reflection on problem solutions, which focuses on understanding the abstract relationships between problems, is related to improved learning. Students often concentrate on doing the task rather than reflecting on it and learning from it. It’s necessary to get students to reflect in order to learn from the task (Collins & Brown, 1988 [6]) and turn experience into learning (Dixon, 1991).

2.1.6 Scaffolding approach

Many researchers have studied scaffolding. Especially in the model of cognitive apprenticeship, developed by Collins, Brown and Newman (1989) [7], scaffolding is mentioned as a means of coaching students until students can perform intellectual tasks on their own.

According to theories developed by Lev Vygotsky, problem solving and other skills can be placed into three categories: i) those performed independently by the student; ii) those that cannot be performed even with help; and iii) those that fall between the two extremes, tasks that can be performed with help from others. This last category is what Vygotsky calls the “**Zone of Proximal Development**”, the domain in which the student is ready to grow.

> ..." The discrepancy between a child's mental age [indicated by the static test] and the level he reaches in solving problems with assistance is the zone of his proximal development. (Vygotsky 1986 p.186-7)"...

Scaffolding essentially means doing some of the work for a student who isn't quite ready to accomplish a task independently. Like the supports that construction workers use on buildings, scaffolding is intended to be temporary. It is there to aid the completion of a task and it is eventually removed. It's an instructional technique whereby the teacher models the desired learning strategy or task, then gradually shifts responsibility to the students.

One quite effective way through which thinking and communications skills can be nurtured is through scaffolding. The environment, proposed by this research, is going to serve as a medium to achieve scaffolding, i.e. to support the students during their first projects and after using this method for several projects, to be able to manage projects even without it. The goal is that students will learn eventually project design and conduction and, more important, that they realize that they should always reflect on their work.

An essential element of making the PBL approach work is to integrate the individual and collaborative learning as well as the scaffolding provided by the different tools. According to Greening, 1998 [16], Scaffolding is the key to Success in Problem-Based Learning.

2.1.7 Resource-Based Learning

Nowadays, society is experiencing the tremendous impact of the information age. As the web turned out to be a huge library of information, students more than ever need to have information-gathering skills. It is becoming ever more important that students develop sophisticated skills in identifying and using information sources rather than that they "cover" a syllabus.

Education must prepare students for learning primarily from resources after they graduate. Even if there were no library or resource problems, the rate of expansion of knowledge and rate of change in professions would necessitate the development in students of new and more extensive information-gathering skills.

Resource-based learning is the achievement of both subject and information literacy objectives through exposure to and practice with diverse resources. Students become active learners as they use a wide range of materials to investigate subject material prescribed within their classroom curriculum (cited in http://stauffer.queensu.ca/inforef/tutorials/rbl/rblintro.htm [66]).

Two essential features of resource-based learning are its flexibility in terms of adaptability to different learning styles and subject areas, and its promotion of student autonomy. Students develop information literacy skills through formal and informal practice with tasks requiring information from an array of resources.

The main goal of resource-based learning is to provide the opportunity for all students to develop independent learning skills, in conjunction with the acquisition of a basic body of knowledge, which will enable them to become life-long learners.

But simply providing resources for learning does not ensure that learning takes place. Students need specific guidance on how to use resources effectively though.

2.1.8 "Genex" for GENerating EXcellence"

Schneidermann (1998) [57] in his article "Codex, memex, genex: The pursuit of transformational technologies" states strongly his belief that:

> "… well-designed technologies can promote excellence. Users, who experience empowering designs that are comprehensible, predictable and controllable, may be inspired to pursue quality in their work products. A genex would be an integrated family of direct manipulation tools that supports users in creating innovations… A well-conceived and clearly-stated framework could guide design efforts, coordinate planning and speed development… "

So, being fascinated by the above article, we wanted our toolset to be a genex toolset that would encourage creative work, if not generate excellence ☺. That's why we chose to follow the genex framework that consists of the four phases "**Collect**", "**Create**", "**Consult**" and "**Disseminate**" (cf. Figure 2.).

Genex framework

Foundational beliefs	Creative Phases	Genex Tools
1. New knowledge is built on previous knowledge	Collect information from an existing domain of knowledge	Digital libraries, Search services Dynamic queries Information visualization Multimedia research
2. Powerful tools can support creativity	Create innovations using advanced tools	Document assemblers Art, design & architectute tools User interface builders Simulations, Models Templates, History, Macros
3. Refinement is a social process	Consult with peers or mentors in the field	Listservs, Newsgroups Conferencing, Groupware Presentation, Annotation Tele-democracy
4. Creative work is not completed until it is disseminated.	Disseminate the results widely	Email, Electronic publications Narrowcasting Affiliation networks, Niche lists E-communities

Figure 2. Genex framework: Foundational beliefs are tied to creative phases and sets of existing tools that can be refined and integrated into a genex (Schneidermann B., 1998).

This meant that we had to develop our genex tools according to the following phases:

- the XML grammar for a mark-up language to model projects (phase "Collect") because "… *powerful tools can support creativity* …". It is interesting what Schneiderman et al. wrote in 1998: "… *Web pages are currently produced by low-level HTML statements that could be extended to contain more explicit semantics and structural information. Richer document description and structuring languages will facilitate generation, sharing and searching of complex information* …" [56].

- the grammar for a mark-up language where one can also archive a project's resources (phase "Create") because "… *powerful tools can support creativity* …" and "… *new knowledge is built on previous knowledge* …".

- the "project cockpit" for monitoring, progress tracking, evaluation and communication (phase "Consult") because "… *refinement is a social process* …".

- and online dissemination through the web (phase "Disseminate") because "… *creative work is not complete until it is disseminated* …".

2.2 Synthesis - Conclusion

In an effort to put all the above theories and approaches together, synthesize them and come up with a concrete list that summarizes the basic points, we ended to the following:

- Project-based learning affords rich opportunities for learning by developing skills and abilities in students that are difficult to acquire with traditional instructional approaches,

- Networked and technology-supported PBL adds the advantages of peering, collaboration, easier management, distribution to bigger audiences; let alone the fact that technology can be used in PBL approaches as a cognitive tool, and *engages* students in their learning,

- Scaffolding is critical to help students learn within a complex project by supporting them all along the project to assure it's completion in the best terms possible as it supports "Proximal Learning",

- Reflection promotes meta-cognition, enhancing students, depth of learning,

- Resource-based Learning strengthens students by making them develop information literacy skills and independent learning skills. Eventually creating life-long learners, a valuable feature in our information age.

So, the conceived method (EVA_pm) corresponds to the above conceptual framework in the following way:

- The **markup-language (EVA-pm ML)** for the project management, enables learners express and concretize their ideas, explore, articulate and reflect on solutions to problems that come out, and provide explanations and justifications, making them better understand the domain knowledge.

  The xml files derived from this grammar will serve as a repository of students' thoughts and ideas, forcing them to think about the issues in depth, structure their work and eventually produce a coherent research design. Based on the notion that "making covert, abstract processes visible, public and manipulable, serves as a necessary catalyst for reflective metacognitive activity" (Derry, et al., 1994 [13]), these virtual "notebooks" (xml files) will be a vehicle for providing hints. And that's exactly where scaffolding emerges. Tags provided with semantics will scaffold students all the way through their exploration for knowledge building. At this point, scaffolding is at the cognitive level.

  Tags for meta-reports are provided also, to encourage reflection and provide the opportunity to generate richer understanding and integration of the knowledge in research. So, scaffolding also occurs at the meta-cognitive level as students gain deeper conceptual understanding and monitor their own understanding of the domain.

- The **editing toolset**, with:

  The proposal of an **XML editor in favor**, chosen among several ones that exist in the software production, which fits into the needs of our approach,

  "stylesheets", so that all interested parties can visualize the contents of the xml files through a common web browser (even if not xml-compliant) and profit from peer interaction .

- The **"cockpit"** or **"Comm&Control"** console, an online monitoring console that enables instructors and students to share the information and visualize it in more concrete ways, e.g. via graphical representations.

  This console includes several **awareness tools** like progress bars, images for audits and evaluation to promote motivation from peering.

It includes also, **"group" discussion tools** to promote collaboration within members of each working group and discussion tools between the instructor and each group to scaffold the interaction (the discussions are asynchronous). Each discussion is "attached" (referred) to a specific semantic, thus the aim is to make these discussions more productive so that the semantic context can be used to augment the cognitive benefits provided by the approaches followed.

commNcontrol has also a **versioning system,** which supports the group editing and keeps track of the evolution of the project. With this feature, tutors can monitor all the steps of the project and students can go back and restart, working from an older version in case it is needed.

The versioning system is accompanied by **File Transfer Tools** that complement it's role.

Moreover, **online forums and mailing lists** are available also for collaboration, as well as, the tutors' feedback and eventual evaluation. Here, scaffolding is at the collaborative level with the discussion tools and at the cognitive and meta-cognitive level for the rest of the tools.

- The **"meta-language (EVA-paper ML)",** defined by a combination of grammars (DTDs), models the paper that the students should write for the final product of their project. This serves the final phase of meta-reflection and gives the student the chance to reconstruct his learning and reflect on it. This also helps the homogeneous dissemination of similar projects, making them highly searchable.

  Virtual Book, a web portal that unites all the papers and complements this grammar, so that peers can profit from it. With this feature, the goal of the whole course becomes more concrete as students can distance themselves from their own project and see it as a part of a general theme.

Chapter

3

3 Research and Development (R&D) Method

This chapter presents the research and development method that was used for this study.

3.1 Research Method

3.1.1 Field exploration

A very important part of this research was to conceive a grammar that would model the work of an academic project. There are different sources of information that have been used to achieve this goal.

The very first steps were to go over the **notes of a methodology course** (Schneider, 2000 [53]) and try to outline the elements that should describe a project.

Then, go over several **academic projects that have taken place at TECFA** (students theses, research projects) and try to decompose them in a way to model the field and draw a project's skeleton (cf. §1.4.3). Themes of the projects had to be different, but thanks to the enormous heterogeneity of this special unit (TECFA), this was from the very beginning an easy target.

Also, the writer had already been actively involved in the project **STAF-18** that is considered to be the birthplace of this research (cf. §1.4.1). This has been an enormous source of ideas and improvements, as well as the birth of new questions for this thesis.

Another important source of information were **EC projects** and the research guidelines that are given with them. Studying how EC projects are organized, how their management has been done and, more importantly how they should be conducted, helped in the choice and naming of the most important elements that had to be put inside the grammar. In this context, projects financed by the European Community, like SEED (http://tecfa.unige.ch/tecfa/research/seed/) and LEARNETT (http://tecfa.unige.ch/proj/learnett/) have been studied.

Moreover, the consultation of two **books,** one about Research Design (Quivy R. and Van Campenhoudt L., 1995 [44]) and the other with advice on how to succeed in a thesis (Fragnière J-P., 1996 [15]) completed the global view of the situation.

A **survey of needs** had been conducted at the same time at TECFA, so that precious information like needs from key persons such as teachers and their assistants would be written down.

Survey of needs

Questionnaire. In order to gather precious information from the key persons involved in projects, like professors and their assistants, a questionnaire was articulated in such a way that would provoke a productive discussion, leading to comments and

suggestions that would improve this research. The idea was to give the questionnaire to a small sample of the unit and stop the survey when the same answers came up again.

Survey analysis. The results of the survey described above, are very interesting and correspond well to the findings that are mentioned in the introduction of this thesis.

The very first remark, that came out of the above discussions, which was very encouraging, was that teachers totally supported the idea of doing such a research and implementing a toolkit for project management. They kept saying that they are in need of such methods for their work.

The arguments presented were that, students need such a tool, both to learn how to work on projects and to help them during a specific project. As for the teachers, the existence of a powerful tool would facilitate their task to follow students' projects, especially many in parallel.

The majority found that the most difficult task is to make students respect the time constraints during their work on a project. This comment had an impact on the final implementation, such that emphasis has been put on awareness tools that provide an extra help, and such that careful attention was given to the definition of the work packages.

Another comment that came out of the discussions is that it is difficult to make students have a coherent research design during the realization of their project. That's why important elements for the grammar had to include their objectives, their approach, and resources that would force students to go over the literature.

Moreover, monitoring and progress tracking of a project is considered important, a fact that led the writer to focus on the implementation of a Monitor Device ("commNcontrol") through which teacher(s), but also students, will eventually be able to monitor several projects in parallel. This will not only facilitate the teacher's task, it will hopefully also motivate students watching the progress of the others' projects (peering benefits) and will serve as a portal for the communication between teachers and students.

3.2 Development method

The development method that has been adopted corresponds to **participatory design** and specifically to **cooperative prototyping**.

"Cooperative prototyping," is an excellent example of how participatory design perspectives can change the way system development is practiced (Grønbæk, 1991 [17]).

Both "prototyping" and "user involvement" (or "user centered design") are concepts that have frequently been suggested to address central problems within system development in recent years. The problems faced in many projects reduce to the fact that the systems being developed do not meet the needs of users and their organizations. The reasons are that user needs and expectations cannot be completely uncovered, written down, and conveyed to the designers in a specification early in the development process. The literature documents examples of such problems, which lead to overrun of project budgets and even to projects being stopped without delivering the requested system.

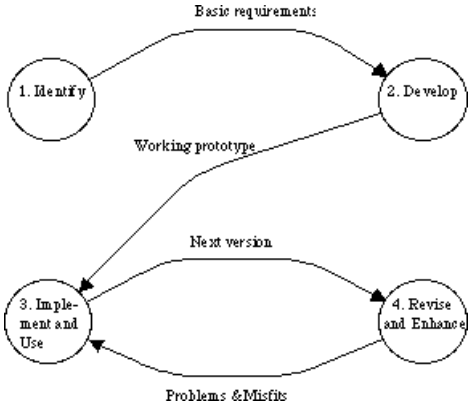


Figure 3. The prototyping model by Naumann & Jenkins (1982)

It is stressed that the initial prototype should be developed in a very short time ("overnight"), and only needs to be based on "half specified" requirements. The prototype should be provided for "hands-on" use by the users. Recognized undesirable and missing features should lead to revision and enhancement of the current prototype to produce a new version with a short turn-around time. The steps 3 and 4 (cf. Figure 3.) are repeated until they produce a system satisfying the users. Hence the generic prototyping model represents an evolutionary prototyping approach.

In a paper by Kraushaar & Shirland (1985) (cited by Grønbæk, 1991 [17]) a "two-prototype" approach is presented, i.e., two prototype versions are made before the final system is implemented by reusing major parts of the second prototype. An interesting point in their description is that they point to the possibility of making modifications of certain parts of the initial prototype together with users: "These 'on-the-fly' changes further encouraged the experimental tone and user involvement." These observations resemble observations made when applying the cooperative prototyping approach.

3.2.1 Cooperative Prototyping versus Traditional Prototyping

The authors who introduce prototyping ideas and approaches mainly put their emphasis on prototyping tools, technical issues, or project models displaying iterative development of prototypes. The users' contributions are typically restricted to initial analysis and a few evaluations of prototypes, as illustrated in Figure 4. In contrast the authors who focus on user involvement are mainly concerned with organizational issues, such as how users get political influence or how to "design" the workplace as a whole. In the approaches described in this literature the users' contribution, however, varies from being sources for knowledge extraction to being the driving force in the approaches. Only late compared to the short history, the concerns of the two research areas are brought together.

Cooperative prototyping brings together the qualifications of users and designers throughout system design processes. Ideally it allows the designers and users to transcend their current qualifications (in cooperation) and to be creative in the process of developing a new computer system. As suggested in Figure 5. , the degree of cooperation changes during a development process, dependent upon the kind of activities undertaken. For instance, users are more active in analysis, design and evaluation activities and less active in implementation activities.

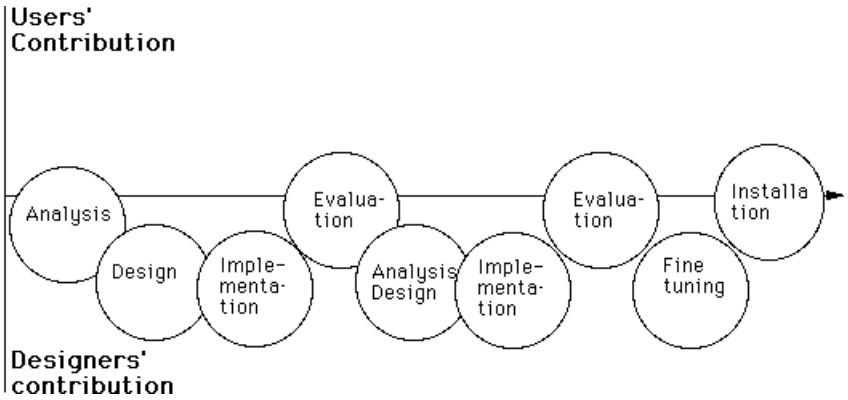


Figure 4. Typical process according to traditional Information System prototyping approaches. Few activities where users participate (from the Ph.D thesis of Grønbæk, 1991).

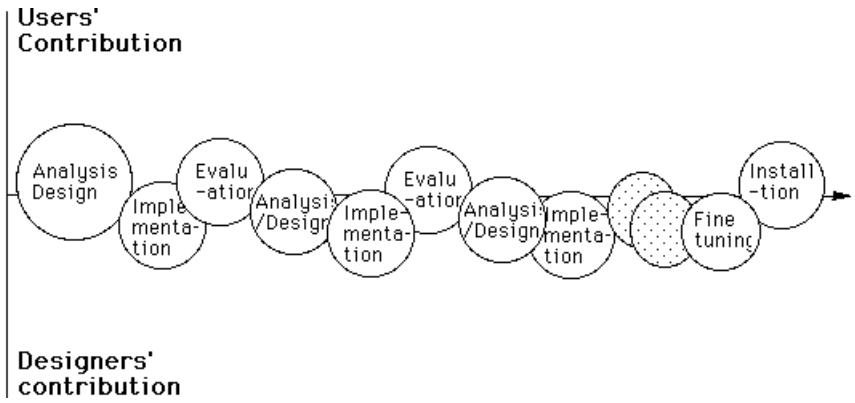


Figure 5. A typical prototyping process as suggested by the cooperative prototyping approach. More activities where users participate, and more interplay between activities (from the Ph.D thesis of Grønbæk, 1991).

Figure 4. and Figure 5. serve as a simplified illustration of the difference between traditional prototyping approaches and the cooperative approach with respect to user's and designers' contributions.

3.2.2 Using Cooperative Prototyping for EVA

In respect to theories for Cooperative Prototyping, EVA has been designed following iterative cycles of user involvement.

The first version of EVA_pm method started on 1999 and was tested with staf-eva students (http://tecfa.unige.ch/tecfa/teaching/staf18/staf18-overview-e.html) for the STAF-18 course. The second one the next year, 2000 and was tested with staf-fanny students (http://tecfa.unige.ch/tecfa/teaching/staf18/staf18-overview-f.html) for the same course to result to the version that is being presented in this thesis, which was tested with staf-gina students, 2001. In all of these three versions, students and tutors contributed by giving feedback on using the method.

One can see the steps in evolution to today's **commNcontrol** and **Virtual Book.** In the first attempt (staf-eva) there was a simple web portal for the projects (cf. Figure 6.). In the second attempt (staf-fanny) the portal became more sophisticated and a web portal was added for the final papers (cf. Figure 6. & Figure 7.).

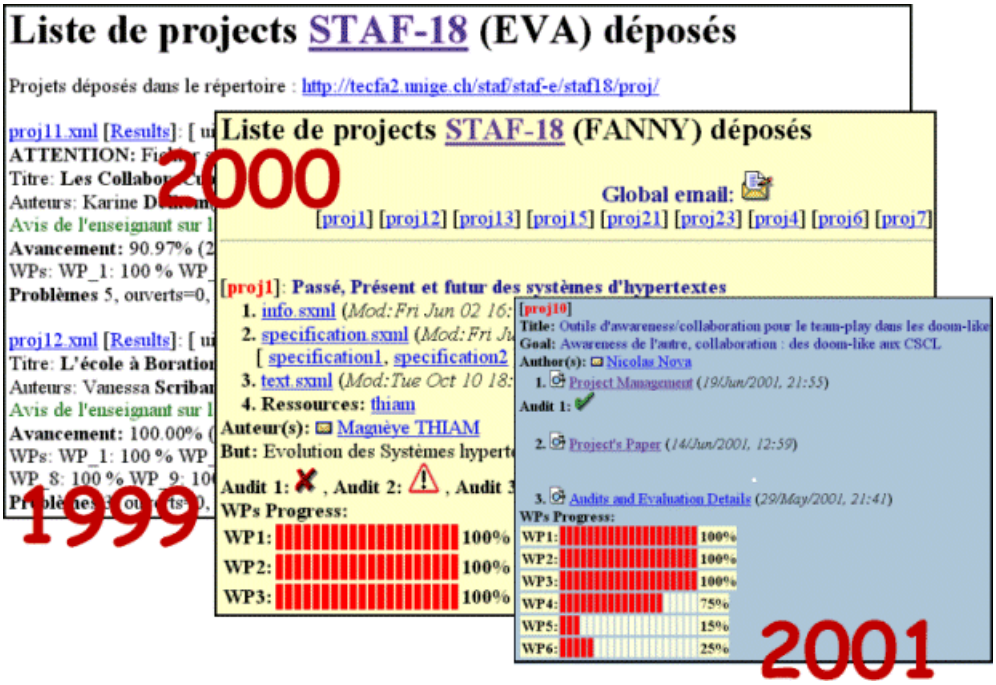


Figure 6. EVA's evolution of versions for commNcontrol (project's portal) for staf-eva, staf-fanny and staf-gina (STAF-18 course, http://tecfa2.unige.ch/staf/staf-e/staf18/proj-liste1.php, http://tecfa2.unige.ch/servlets/staf18FList2, http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/commNcontrol.html?action=commNcontrol).

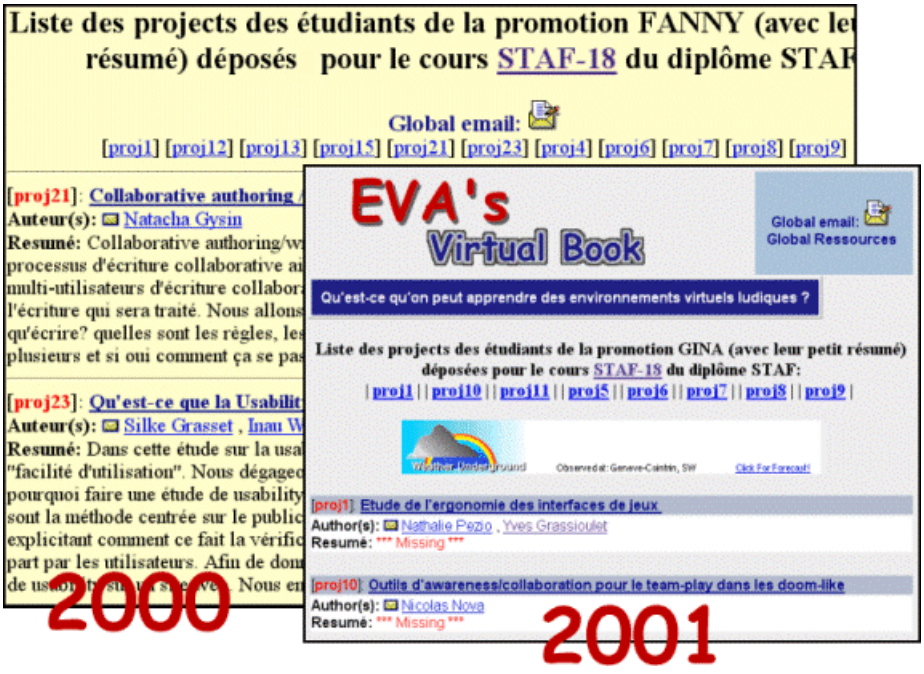


Figure 7. EVA's evolution of versions for the final papers' portal for staf-fanny, staf-gina (STAF-18 course, http://tecfa2.unige.ch/servlets/staf18FList3, http://tecfa.unige.ch/staf/staf-e/paraskev/servlet/journal).

In the same way, the XML grammars used to support the method have evolved according to feedback from the users (students and tutors) among with other.

So, the steps that were followed for the development part of this research are the following:

- Review previous versions of EVA for staf students. Keep track of problems, misconceptions, and users' feedback (students and tutors)

- Review similar initiatives (Thinking tools, mindtools, design diaries, reflective notebooks etc.) and review of the state of the art that they are based on.

- Conduct a survey at TECFA, about what information need to be stored for a project.

- Re-study the grammars of the previous versions and develop the current version of the markup language (DTD) for academic project management.

- Test the grammar within a teacher-less community. Two students (Bertrand Resin and Olivier Clavel), who work as assistants at TECFA, tested the grammar while trying to model their master theses with it.

- Correct mistakes, re-analyze etc…

- Develop supporting tools (editor+profile, stylesheets for the grammar, cockpit-commNcontrol, versioning system and supporting tools, Virtual Book portal)

- Test within a teacher-learner community (students of staf-gina, STAF-18 with Daniel Schneider as tutor).

- Redesign according to comments, likes and dislikes of the testers (students and tutors).

- Testing, evaluation and refinement.

In conclusion, the last three years (1999-2001) we have experimented with the method that is presented in this thesis and used cooperative prototyping to develop it. That's why every year we redesigned the method and tried to profit from the experiences and the feedback from the previous years.

Chapter

4

4 EVA's XML Infoset

This chapter starts with a short presentation of XML and a list of the most important initiatives for education. The conception of EVA's grammars follows and their elements are presented in detail.

4.1 XML: The New Standard

XML (e**X**tensible **M**arkup **L**anguage) is a markup language for documents containing structured information, which means that is a mechanism to identify these structures.

Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure. The XML specification defines a standard way to structure documents (XML.com, [69]).

The innovative part of XML is the XML "data formats". These include vector graphics, e-commerce transactions, mathematical equations, object meta-data, server APIs, and a thousand other kinds of structured information.

XML is a method for putting structured data into a text file. It's a set of rules, guidelines, and conventions for designing text formats for such data, in a way that produces files that are easy to generate and read (by a computer), that are unambiguous, and that avoid common pitfalls, such as lack of extensibility, lack of support for internationalization/localization, and platform-dependency.

XML looks a bit like HTML but isn't HTML. Like HTML, XML makes use of tags (words bracketed by '<' and '>') and attributes (of the form name="value"), but while HTML specifies what each tag & attribute means (and often how the text between them will look in a browser), XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it.

XML is new, but not that new. The designers of XML simply took the best parts of SGML, created a subset of it (trying to overcome the deficiencies of HTML), and produced something that is no less powerful than SGML, but vastly more regular and simpler to use.

XML is a family of technologies XML, XLink, Namespace, DTD, Schema, CSS, XHTML,... That's why we refer to it as an **XML infoset**.

To understand better XML, someone should look the **XML framework** of W3C. This consortium has defined XML languages for transforming in different formats of documents (XSL/FO, XSLT) and languages for the manipulation or association between them (XPointer, XLink, XQL). In the near future this give us the possibility to create powerful hypertexts or to generate new documents by merging parts of existing ones.

By now, there are already such languages for multimedia like SMIL, for mathematical representations like MathML, for 3D representations like X3D (that replaces VRML) or for vector graphics like SVG.

4.1.1 Technical advantages of XML

- XML is license-free, has built-in internationalization (full Unicode) and platform independence
- XML is used to exchange data between incompatible systems.
- XML can be used to share data, as it is simple text (ascii) files i.e software and hardware independent with minimal bandwidth requirements. One of the foremost issues is preservation and archiving in the inevitably changing arena of digital technology. Flat data (e.g., ASCII text) is easiest to preserve and "migrate" or convert, to later innovations in formats or technology without significant risk of data loss.
- XML can be used to create other languages that correspond to our needs.
- As it structures the information, it makes it easily searchable.
- It is well accepted as the new standard for the Internet by many big vendors…

XML is verbose, but that is not a problem (W3Schools, [68]). Since XML is a text format, and it uses tags to delimit the data, XML files are nearly always larger than comparable binary formats. That was a conscious decision by the XML developers. The advantages of a text format are evident (see above), and the disadvantages can usually be compensated at a different level. Disk space isn't as expensive anymore as it used to be, and programs like zip and gzip can compress files very well and very fast.

4.1.2 Conceptual advantages of XML

XML is a markup language for documents containing structured information. A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents according to the meaning of the information included and not according to the way it should be rendered, e.g. in a browser.

In fact, XML is really a meta-language for describing markup languages. In other words, XML provides a facility to define tags and the structural relationships between them.

XML has quickly become the accepted format for information interchange over the Web and for structuring information. It's standards based. It separates style from content. It describes information in one part (xml) and it’s layout in another part (xsl). It lets categorize, search, process and combine documents, and parts of documents from both internal and external sources so that we can quickly deliver the most targeted information possible.

So, its main characteristics can be summarized as:

- XML adds **semantics** to the data (makes data self-describing) (“The semantic Web” as Tim Berners-Lee calls it),

- XML contains **no fixed set of markup tags** – one can define his own tags, tailored for his kind of information and, therefore, create different vocabularies,

- XML **separates** data from it's presentation and process,

The benefits to developers and users are:

- More meaningful searches; if commonly accepted grammars are adopted (e.g., Docbook for all books).

- Development of flexible Web applications; once data is found, different applications can undertake and process them in various ways.

- Data integration from disparate sources; data that come from different sources can be combined.

- Local computation and manipulation of data; once data are available locally to the client, the client can manipulate them without the server's support.

- Multiple views of the data; local data can be presented dynamically in a manner determined by client configuration, user preference, or other criteria. CSS and XSL provide declarative mechanisms for describing a particular view of the data.

- Delivery of data on the web; as xml is an open text-based format, it can be transferred using HTTP protocol without need of changing existing networks. Not to mention that it compresses extremely well due to the repetitive nature of the tags used to describe data structure.

XML was created so that richly structured documents could be used over the web.

The formalisms, that exist so far for XML grammars, are several: DTD, XSchema, RELAX, etc. The most popular ones, though, are DTD and XSchema, with the latter one being more recent, thus more powerful but became definitive specification only recently.

But, why should one need structured text in education? First and most important reason that we support within this thesis is for **scaffolding,** as we have already suggested. Providing a grammar for structuring a document helps students or teachers think about what they are doing and helps them to get the task done. In that way, XML becomes a **cognitive tool**. A second obvious reason is to **leverage** from the XML framework. XML is here and we have to profit from it as much as we can, why not then also for education?

This project has been designed to take advantage of how eXtensible Markup Language (XML) answers multiple issues confronting academic institutions considering electronic thesis/ dissertation (ETD) projects.

4.1.3 XML deficiencies

The full XML Framework is quite recent, and no major manufacturer has released a full-blown browser to access its various capabilities. There are some problems that need to be addresed like quality of grammars, lack of sufficient tools (especially for editing), and eventual rendering of content (Schneider, 2001 [54]).

Grammars need more documentation (both Internal and External) and apparently there are some contradictory criteria that need to be overcome, like richness vs. simplicity and human vs. machine usability.

But grammars are not enough. Tools are essential for supporting such grammars. The ones that exist are still in a premature state and some are yet to be created. Editors have to become user-friendlier for end users, and we are in need of various tools like Visualizers, Analyzers, tools for Document Management, on-the-fly assemblers, and much more.

And there is still, of course, the content presentation. We are in need of several stylesheets at least that will accompany each grammar and support multiple publishing channels like web (HTML, WAP), PDF, PostScript, etc., and of course XML browsers more than anything.

If we go back again to education, there are even more things that we are in need, because XML doesn't change the fact that students don't learn by just reading. There must always be a pedagogical scenario that comes along with a schema, as well as tools for collaboration, reflection, etc. That's why in the learning environment that is proposed in this thesis, we focus not only on the elaboration of the grammar, which of course is very important, but also on the tools that will accompany it and also to the pedagogical scenario that this will fit in.

We study the instructional approach within which our environment will be used (PBL in our case) and go over all the details that will ensure this use to be beneficial.

4.1.4 XML initiatives for education

XML can become a cognitive support for the editing process. Schemas can help teachers to plan their courses and write educational texts. Students can learn to write reports or research plans following specific norms defined by XML grammars. Structured texts are easier to be indexed, especially for the web and create powerful documentation systems with less effort and cost.

Markup Languages have been created for various areas. There are already several initiatives for such grammars like IMS Metadata, DocBook (for books), TEI (Text Encoding Initiative), MathML, IEEE, ADL/SCORM etc.

This section reviews a few of them that are close to our approach, in order to show the potential of XML in education, and how it has been used to markup a wide range of information. In education, specifically, there are several vocabularies that came up like Tutorial Markup languages, Quiz Markup Languages, Student Management ML's, and so on with the most popular ones being IMS, IEEE LTSE and MathML.

Instructional Management Systems (IMS): This is a US initiative that started work with the academic community in developing and promoting open specifications for facilitating online distributed learning activities such as locating and using educational content, tracking learner progress, reporting learner performance, and exchanging student records between administrative systems. They are mostly focused on Metadata and Content-packaging. IMS Project Group has developed grammars for several areas: Multiple-choice test, multiple-choice test answers, Performance data, Notifications and Navigation and in particular, for Meta-data (together with IEEE) and content-packaging. (http://www.imsproject.com)

IEEE LTSC (Learning Technology Standardization Committee): This is the only body engaged in the educational domain, which has a recognized formal standing. That's why many of the other groups (IMS, CEN/ISSS, AICC, etc) participate in the IEEE process and aim to progress their working specifications through the IEEE adoption procedures. It has several working and study groups such as: General groups, Learner-Related Groups, Content-Related Groups, Data and Metadata and Management Systems and Applications. Among them, the most advanced areas are the last two, and more specifically, the "Learning Objects Metadata (LOM)" and "Computer Managed Instruction (CMI)". (http://www.manta.ieee.org/p1484)

DC Education (Dublin Core Group): The Dublin Core Group defined elements to describe on-line educational resources. The Dublin Core Metadata (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards and developing specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems. (http://purl.oclc.org/metadata/dublin_core)

ADLnet SCORM (Sharable Content Object Reference Model): The Sharable Content Object Reference Model (SCORM) is a set of interrelated technical specifications built upon the work of the AICC, IMS and IEEE to create one unified "content model". These specifications enable the reuse of Web-based learning content across multiple environments and products. Currently, SCORM consists of three main sections: an Extensible Markup Language (XML)-based specification for representing course structures (so courses can be moved from one server/LMS to another); a set of specifications relating to the run-time environment, including an API, content-to-LMS data model, and a content launch specification; and a specification for creating meta-data records for courses, content, and raw media elements. (http://www.adlnet.org/index.cfm)

LMML (Learning Material Markup Language): This is an XML-based markup language designed for educational contents (teachware). It provides sub languages for various educational fields and yet remains further extensible. Therefore it represents not only one single markup language but rather a family of various languages. LMML-CS is one member of this family and thus called an instance of LMML. It is applied to describe topics from the field of computer science (CS). (http://daisy.fmi.uni-passau.de/pakmas/lmml/11/doc/en/html/)

Tutorial MarkUp Language (TML): Tutorial Markup Language (TML) is an interchange format designed to separate the semantic content of a question from its screen layout or formatting. The language is designed to support several different types of questions within the same content model. TML 4.0 is essentially a super-set of HTML, with new elements added to describe question information. TML was developed by Daniel Brickley at the University of Bristol and it handles 4 types of questions. It handles different actions whether the user answers the questions right or wrong and also provides hints. (http://www.ilrt.bris.ac.uk/netquest/about/lang/)

Quiz MarkUp Language: Developed by Robert Bamberger, Christopher Shorey and Richard Simpkinsson at the Washington State University, this is a markup language designed to quickly and easily create quizzes in Assymetrix Toolbook for use as applications or as internet based documents. (http://ems.cea.wsu.edu/QML/)

Question MarkUp Language (QML): QML is copyrighted by the Tekamah Corporation and is a markup language for multiple-choice questions. (http://www.qmark.com/perception/help/v2manuals/auth_22_html/auth22_app_a_start.html)

QuizzIt: This was developed by Lucio Tinoco at Virginia Technical Institute to provide a powerful and easy way to use authoring tool for writing web-based, multiple-choice tests. It provides means for adapting quizzes based on the performance of the students. (http://pixel.cs.vt.edu/~ltinoco/quizitdocs/index.html)

SATML (Standardized Assessment and Testing Markup Language: This is a markup language created to enable instructors to share instructional material. The language is developed as an Extensible Markup Language (XML), and it is intended to be used as a common file format for Internet distance learning tools. The markup language can model the information of an entire course but the research has been focused towards creating a markup language for requirements of questions and quizzes. (http://132.170.199.196/satml)

All of the above attempts have used XML to markup instructional data (management or educational). The areas that have been exploited so far involve mostly **content packaging**, **student management**, **metadata** and **quiz markup**. Our approach will add a new unexplored domain that we call **educational project-management,** which goes beyond the goal of marking up educational data for sharing. We are leveraging XML to create a "**thinking tool**" that will scaffold PBL strategies, and this makes our initiative innovative and distinguishes it from all of the above.

4.1.5 DTD vs. XML Schema

Grammars are ways of describing classes of XML documents (like the grammars for other languages). From all the formalisms that exist, so far, for XML grammars we support the DTD (Document Type Definition), which is the most popular one, and XML Schema, which seem to be its successor.
DTDs are formal document grammars that employ various types of rules, called markup declarations. They provide document designers with a means of defining structures for capturing content in efficient and appropriate ways.

Unfortunately, the DTD suffers from a number of shortcomings and has several weak points such as:

- too simple: not enough expressive power
- too complex: difficult syntax (not even XML :)
- mixed into the XML specification

Many of the issues stem from the fact that a DTD specification is not hierarchical. Finally, DTDs do not allow one to formally specify field-validation criteria, such as the 5-digit (or 5 and 4) limitation for the zip code field. To remedy these shortcomings, a number of proposals have been made for a more database-like, hierarchical "schema" that specifies validation criteria. May 03, 2001, W3C XML Schema published this as a W3C Recommendation (http://www.w3.org/XML/Schema).

A W3C 'Recommendation' indicates that a specification is stable, contributes to Web interoperability, and has been reviewed by the W3C Membership, who is in favor of supporting its adoption by academic, industry, and research communities. XML Schemas define shared markup vocabularies, the structure of XML documents that use those vocabularies, and provide hooks to associate semantics with them. With over two years of development and testing through implementation, XML Schema provides an essential piece for XML to reach its full potential. The XML Schema specification consists of three parts. One part defines a set of simple datatypes, which can be associated with XML element types and attributes; this allows XML software to

do a better job of managing dates, numbers, and other special forms of information. The second part of the specification proposes methods for describing the structure and constraining the contents of XML documents, and defines the rules governing schema-validation of documents. The third part is a primer, which explains what schemas are, how they differ from DTDs, and how someone builds a schema. XSchema proposal specifies validity constraints for an XML document using XML. It is one of several proposals that aim to be the successor to DTD. It is not yet clear what the final validation standard will be. (OASIS, The XML Cover Pages, *W3C XML Schema Published as a W3C Recommendation.* Retrieved from the Web on 08/06/2001 http://xml.coverpages.org/ni2001-05-03-c.html).

XML Schema is more recent, thus more powerful than the DTD. It defines the types and the format of the elements and the attributes, and accepts explicit comments that describe the elements as hints, which are clear advantages over DTD.

Unfortunately, because of lack of tools needed due to the immaturity of XMLSchema, we decided to conceive the grammar in the format of a DTD, which will be easily transformed in the near future to it's corresponding XML Schema.

4.2 EVA's Grammars

EVA has defined several xml grammars (DTD's) in order to better describe a student's work on a project and to support him during this work.

The grammars, which have been implemented with the goal to serve as **cognitive support** to the students for their projects, are the following two (explained later in this chapter in detail):

- the **project.dtd** that describes the project management procedure
- the **paper.dtd** that describes the final paper including the results on the student's research

Both of these grammars are accompanied by a third grammar that is included in them as an external entity:

- the **IBTWSH** (Itsy Bitsy Teeny Weeny Simple Hypertext DTD), a slightly modified version of John Cowan. The Itsy Bitsy Teeny Weeny Simple Hypertext DTD (IBTWSH) is an XML Document Type Definition (DTD), which describes a subset of XHTML Basic for embedded use within other XML DTDs. Its purpose is to enable documentation within an XML document that offers the enhanced readability of rich text, as opposed to documentation that ends up being #PCDATA only. When the IBTWSH DTD is incorporated by reference (as an external parameter entity) into another DTD, that DTD inherits the capabilities of it. If HTML-compatible elements and attributes are used, the documentation can be displayed by HTML renderers, which is it's most important advantage. So, elements of the above grammars that serve for long descriptions can be enriched by giving the possibility to the users to insert images or links inside their text, emphasize it, make lists and tables, etc. (http://home.ccil.org/~cowan/XML/ibtwsh6.dtd)

Apart from these grammars that are the most important ones in our research, there are several others that have been conceived in the aspect of using xml as the standard format for storing data. These grammars are used by the tools that complement EVA_pm (i.e., for discussion, for the tutors' evaluations, for versioning, for groups and accounts - management of the project) and they are the following:

- the **dicussion.dtd** for storing the discussion between group members and the tutor, organized around open-questions and comments originated from the project management xml file,

- the **evaluations.dtd** where audits and evaluations for each project are being archived along with their metadata (author, date, for whom),

- the **versioninfo.dtd** that keeps track of all versions uploaded (metadata and locking information) and supports the versioning system of EVA,

- and finally, the **authentication.dtd** that stores group and tutor information and their accounts and supports the authentication system of EVA.

All of the above DTD's (6 in total) are accompanied by the corresponding stylesheets in XSLT, in order to visualize the documents in HTML format in a web browser (since for the moment there doesn't exist an XML browser).

4.2.1 Conception of EVA's grammars

Practically speaking, students that use EVA should collaborate with the other members of their group for the editing of two (2) xml files, the *project.sxml* and *paper.sxml* (the extension "sxml" is for the Cocoon processor). The first file, *project.sxml,* will store all the information about the project's specification and work management and the second one, *paper.sxml,* the results of the project in the form of an official paper.

Actually for the first grammar (project management), XML plays a significant double role. On one hand, it serves as a **prompting tool** that scaffolds students during their work on the project (by help them define their specification and manage the work), and on the other hand, it **structures the information** in order to extract subsets of it.

The first goal, **scaffolding**, is achieved by providing sections for helping students initiate their inquiry (like the sections for the main goal and objectives), direct investigations (like section for resources), and manage their time (by defining the sections for the work packages). So, the tutor, instead of teaching students a subject, has only to coach them in answering the given questions/sections.

As for the second goal, **structuring the information**, it is achieved by default, as it is the main advantage of xml. The consequence of this feature is that one can extract and visualize the stored information in many different ways (by applying different stylesheets or by using scripts that parse the information).

For the second grammar (final paper), the goal is to make students concretize their work and culminate in a realistic result, which will make students enhance learning.

For this second grammar, another characteristic of xml is being leveraged. The power and beauty of XML is that it maintains the separation of the user interface from the structured data. In XML, stylesheets such as Extensible Style Language (XSL) and Cascading Style Sheets (CSS) are used to present the data in a browser. XML separates the data from the presentation and the process. That means that the display and processing of the data can be done by applying different style sheets and applications.

Students often have the tendency to spend a lot of time in formatting their text instead of focusing on the content. In the case of EVA, students get a grammar that helps

them to structure their ideas and to present them in a coherent way, while a stylesheet is provided that undertakes the formatting. Another stylesheet merges information from another xml file and eventually adds the title, authors and references to it. In addition, a java servlet extracts information from several papers and creates a web portal in the form of a Virtual Book.

Special attention has been given, following the rules of authoring a DTD (for it's design and customization), so that the grammar will be easily expandable, updated, flexible, and so that the text that will be inserted will be as rich as possible (by inserting ibtwsh6.dtd).

4.2.2 EVA's Grammar Overview for project management (EVA_pm ML)

So, the resulting grammar (**project.dtd**), that corresponds to our research goals and objectives, defines eight parts of information that are the basic components of a student's project:

- **project-info;** all information about the project such as the title, acronym, type, description, authors names and info, dates, supervisors, versions, keywords, URLs.

  These comprise administrative information that must follow each project as usual. Keywords are important information to pointed out here, as they make the project's information easily searchable.

- **project-specification;** like main goal, objectives, target public, approach used, scenarios that apply, feasibility study, deliverables, work packages with the expected results, discussion about it and final report on it.

  A very significant part for the design of the project, as it serves as prompt to the student(s) to reflect on the goal, the objectives, the approach that they are going to use, the feasibility study that they should conduct before starting the project, the deliverables that they promise to provide at the end, and they provide a sketch of the work packages by splitting the tasks into meaningful sections. By describing their objectives also for each work package, that correspond to the general objectives make more sense to their work packages, and the usefulness of each work package is ensured.

- **project-problems;** detailed description of the problems faced and possible solutions, discussion about them and final report on them.

  This serves as a detailed list of all the problems faced during the realization of the project followed by all the possible solutions presented from group members or the project's supervisor. Keeping track of the problems faced helps students to go over them once they have finished the project and to meta-reflect on them.

- **project-meetings;** description of the meetings and information like participants, discussion about it and final report.

  This section ensures the successful achievement of the project, as it forces the students to keep in touch with the other work members of their group (if there are any) and most importantly with their supervisor. These are iterated cycles that allow them to go over their work and re-evaluate it, giving them the possibility to present what they have done already and to discuss it.

- **project-results;** the list with project results (according to promised deliverables)

This is the section where the goal of the project has been reached: the end product(s) has finished. So, a description of this product is presented together with other information that complement it like URL's, source code, PDF files etc. depending on the nature of the product.

- **resources-for-project;** the list of the resources used for the project (books, articles, URLs, etc) with all the corresponding information, a discussion about them and a final report.

  Another part that is crucial for a project is the detailed list of all the resources that are needed to realize it. This is accompanied with a citations session where a short description of the contribution of the resource is given. There is also a discussion session where the student can get immediate feedback from all the interested parties concerning the resources mentioned. Through this feedback, the tutor can guide the students and implicitly drive them to the right field.

- **comments;** a final report for every part or for the whole project.

  This part should always follow a PBL approach to provoke reflection and meta-cognition by asking the student to go over every part or the whole project and evaluate the procedure and the end product(s).

- **open-questions;** a dialog between the authors and their supervisors that can be inserted in almost every section.

  This provides a very productive dialog where the authors can exchange ideas between them and with their supervisor. Again, the tutor has the chance here to guide the students.

4.2.3 EVA's Grammar Overview for the resulting paper (EVA_pm ML)

The second grammar (**paper.dtd**), that complements EVA, is a simplified grammar for the resulting paper, which presents the project's results. It structures the information in eight parts that are the basic components of a semi-formal paper:

- **abstract;** the summary of the paper.

- **preface;** personal information like acknowledgements.

- **introduction;** a short introduction to the project where it's purpose, it's organization, it's hypotheses, methodology, definitions, etc. is presented.

- **main;** the main discussion of the project, the conceptual framework that it is based on, the work that has been done, and the final results go here.

- **conclusion;** this section presents the results of the project, the conclusions that arise and gives it's perspectives.

- **annex;** all information about the project that are secondary and don't fit to the other sections go here, such as program listings, tables, figures, glossary, index, etc.

4.2.4 Support for EVA's grammars

A problem that we face with xml grammars is that the users have to learn the grammar in order to know which elements are allowed and in which combinations.

Official grammars like DocBook for books require time for learning a tutorial or reading a book (a quite big one for DocBook, actually) so that users will learn how to use it and profit from it, and still users need to have always a reference manual available. So, to make our grammar easier to learn and to make a reference manual, two applications were used.

The first was "**DTDparser**" by Mark Wutka (http://www.wutka.com/dtdparser.html). DTDParser is a Java module for parsing XML DTDs separately from any XML document. The parser can parse from any kind of stream and builds an object structure representing the DTD. This structure can then be accessed to extract information about the DTD and can also be modified. For EVA's grammars, DTDParser automatically created online a web site with the whole structure of each grammar decomposed into it's elements and it's attributes and can be found in the following addresses:http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/dtdparseproject/, http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/dtdparsepaper/.

As this application didn't provide a way to add important features such as giving a short description of each element and attribute of the grammar, a second application was used. This application is called "**dtddoc**" (DTD Doc generator) by Lars Marius Garshol (http://www.garshol.priv.no/download/software/dtddoc/). dtddoc is a DTD documentation generator which can read a DTD and associated documentation written in XML and generate nicely formatted HTML documentation or, experimentally, DocBook RefEntry documentation. It is an implementation in Python that reads an xml file with the description of each element and attribute of the grammar, parses the grammar and creates an online documentation about it. The documentation created for the project management grammar (project.dtd) can be found here: http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/projectDTD.html

Together with the above two reference manuals online, a tree representation of the grammar was given to the students in format MS-Word (*.doc) together with an example xml file. All three of them can be found in the Appendix A: EVA_pm's grammars.

Chapter

5

5 EVA's Toolset

Chapter 5 concerns the tools that support EVA_pm. It starts with the search for a friendly xml editor that will facilitate the editing of xml files and continues with a detailed presentation of EVA's web portals (commNcontrol & Virtual Book) and their tools.

5.1 In Search of A Friendly Editor

As the goal of this thesis was to provide a toolset that would facilitate student projects, we had to choose an editor that would make it easy for the students to create and work with XML content. This was a very important parameter for the success of our method. The reason being that, although XML offers many advantages, it also entails the problem of editing.

Even though XML Framework, is considered to be the evolution to the web, standards are still under continuous change. Only two years that passed since the official announcement of XML but it has been well accepted and almost everyone is positive about it. The advantages that it offers are obvious as the standard for transferring information and also because of the structured presentation of the information that offers.

Unfortunately, the weak point so far is the existing editors. There are quite a few that exist (and many that emerge from time to time), but we don't have still a leading editor so far. Of course, having an editor for XML is not as easy as having one for HTML, for example. XML is not a language but a meta-language. An xml editor has to go over the specified grammar (DTD or Xschema) every time, which is a subset, adapt its interface to correspond to it and, eventually, validate the xml file created against this grammar.

In consequence of the above, our situation was very delicate: we had to find a way to lift the cognitive load of editing as much as possible. So, we tested several well-known editors in order to choose the most appropriate one. We knew from the very beginning that a complete xml editor didn't currently exist. We were ready to make compromises, as long as we didn't have to betray our objectives.

Having in mind all the above, we set our criteria and our expectations for a "suitable xml editor" for our method and split them in "major" and "minor" (according to their importance).

Major criteria:

- being aware of the grammar (DTD)

  so that can propose elements and attributes to edit at a given point of the document;

- possibility to **validate against a grammar** (DTD and/or XSchema) and find errors,

  to avoid errors in editing by the students i.e., "valid" xml documents;

- **friendly and ergonomic interface** for end-users (like students in our case),

  making easier the authoring of XML documents, shifting the cognitive load of the editing to concentrate on the project;

- **ability to configure** according to the grammar (a bit like a WYSIWYG editing),

  so that we could provide extra help to the students by making the presentation of the interface more appealing with the addition of colors, icons, ToolTips etc.

Minor criteria:

- good documentation and help,

  to learn the interface faster and more efficiently;

- **portability** to many platforms,

  to avoid restrictions of platform and succeed in the portability of our method;

We have to note here that we didn't take into consideration the cost of buying an xml editor. We consider that such a parameter, even though it plays a significant role in general for comparisons, is far less important for an educational framework than the other criteria that we had set.

There are two kinds of XML editors that exist: **text** editors and **graphical** editors. Below follows a short description for each editor that we tested (always the trial or demo versions for the commercial ones) in consultation with the article "Editeurs XML" (**.netpro**, no26, october 2000 in French – L8290).

5.1.1 Text editors

Text editors are the economy tools of XML. They display everything in one-type face (although different colors can be applied), cannot separate out the markup from the content, and seem in general more difficult to people who are used to graphical word processors. However, these surface details hide the secret that good text editors are some of the most powerful tools for manipulating text (Learning XML, Erik T. Ray, 2001, Eds. O'Reilly). The most popular text editor, Emacs, is presented here.

Emacs

Emacs is a text editor with brains. Emacs has been the favorite editor of developers for decades. It comes with a built-in programming language, many text manipulation utilities, and extra-added modules that can customize Emacs for XML, XSLT and DTD. A must-have is Lennart Stafflin's *psgml* mode (http://www.lysator.liu.se/~lenst), which gives Emacs the ability to highlight tags in color, indent text lines and validate an xml document against it's grammar.

At TECFA, we are using XEmacs (http://www.xemacs.org/), a powerful, extensible text editor with full GUI support, initially based on an early version of GNU Emacs 19 from the Free Software Foundation and since kept up to date with recent versions of that product. It is not fully aware of the DTD as it proposes elements or attributes to add, but allows the user to add them in wrong places.

5.1.2 Graphical editors

The majority of computer end-users prefer to write their documents using graphical editors (word processors). Easy editing and WISIWYG display are mostly the things that make editing with these tools more appealing than text editors. SGML editors seem closer to this principle but there are very expensive and difficult to maintain (e.g., Adobe Framemaker+SGML). A comparative review of several graphical editors that are commonly used follows:

XMETAL2.0 beta version by Softquad (http://www.xmetal.com/)

Plus: The editor has the **friendliest interface** compared to the others (It is the closest XML editor that looks and feels like a word processor) that **can also be configured** for a specific grammar (very important feature), although the configuration is quite difficult and only computer specialists can do it.

Something that hasn't been tested to see how efficiently it works was that **it doesn't validate against the grammar defined, but against a file of rules that it creates automatically from the DTD**.

XMetaL allows system administrators to provide **alternative names for elements** that are more human friendly. These names are displayed instead of the actual names in various places and allow users to understand what is going on without knowing the element names. For instance, after customization, the element name displayed below the document window can be changed (for an HTML document) from something like "HTML / BODY / DIV / H1 / EM" to "Document / Body / Section / Level 1 Heading / Emphasized Text", which can be more useful. **Descriptions** can also be given for attributes and elements that give users clues as to how to use them.

Moreover, XMetaL comes with a **built-in spell checker and thesaurus** that can be used to spell-check and enrich documents. Dictionaries in American and British English, French and German are provided.

Minus: Only for Windows platforms, uses a CSS stylesheet to create a formatted display or XSLT (with MSXML) with bugs ☹ and is not DTD aware.

EpcEdit (http://www.epcedit.com/)

Plus: epcEdit is a **cross-platform** that contains an integrated validating parser XML/SGML editor, an editor for CALS and HTML tables, an attribute editor, and an element manipulation tool. The hierarchical structure of a document is represented by a tree view that is updated while the document changes. It **validates** a document against it's DTD and it is **DTD aware** as it highlights if the user adds a wrong element.

epcEdit's unique railroad diagrams support the user in **understanding the intended structure of unknown or rarely used DTD** constructs, which is a unique and valuable feature. epcEdit focuses on providing the user with information about the document, the **allowed content** of XML/SGML elements, their attributes, and their content model as defined by the underlying DTD.

epcEdit's powerful attribute editor supports conformance checking of attribute values and has **special support for ID and IDREF** attributes. Together with the ability to **verify and enumerate the usage of IDs and IDREFs** in a document, documents with references to undefined IDs can easily become a thing of the past.

Minus: Unfortunately, it doesn't accept relative paths for the declaration of the DTD, and the only solution seems to be to give the URL of it instead.

Xeena by IBM

Plus: A Java-based XML editor (i.e., **cross-platform**) from the IBM Haifa Research Laboratory that includes syntax directed editing based on DTD recognition. It reads the DTD in order to provide context-sensitive icons for inserting new elements and **validates** the document against it's DTD.

Editing is done in a **tree view**. The XML attributes of the elements are edited via a table. Each attribute value is entered using an editing GUI component (e.g. combo-box, text-field), which is also derived from the DTD. The editor guides the user in inserting elements into the tree in a correct order (according to the DTD) by making the elements palette sensitive to the current selected tree node and by not allowing the insertion of elements in an invalid order. Users can thus create/edit/expand any document derived from that DTD, by using a visual tree-directed paradigm. The visual paradigm requires a **minimum learning curve** as only valid constructs/elements are presented to the user in a context-sensitive palette. Xeena is a Multiple Document Interface application (MDI) with full-fledged support to **edit multiple XML documents** and copy, cut and paste from one document into another. Unfortunately, somehow some of the above don't work exactly as it should …

Xeena **can be customized** to use application-specific icons and a color scheme for a specific grammar. (It is this optional customization that distinguishes Xeena from XML Pro and XML Spy and Xeena was the first editor that appeared with this feature.)

Minus: Unfortunately, it lacks critical flexibility. For instance, to open a new DTD, you need to reinvoke the program from the command line, referencing the appropriate DTD. The installation does provide some pre-set-up DTD and profile editors, however, to simplify this somewhat. Editing is very slow (one of the Java deficiencies).

In configuration, it shows elements with alphabetical order and not by the order that they appear on the dtd (very inconvenient for understanding where you are exactly). Also, it seems impossible to configure attributes, only elements.

In the editor, for the PCDATA one has to add as a child a TEXT element in order to edit the element, which normally shouldn't be necessary and moreover, the window for TEXT is inconvenient for writing a big text (always the case with tree-editing through forms).

Besides the configurable environment, it still remains an editor for experts and seems difficult to offer as a tool to an end-user for editing the xml files.

XML WRITER by Wattle Software

It doesn't validate automatically against the grammar and supports only the windows platform. We **cannot configure** the interface against a specific grammar, which we consider as an important feature for our project and that is why we exclude it.

XML Spy by Icon Information Systems (http://www.xmlspy.com/)

A nice interface with forms that becomes too complicated compared to the others if we have long grammars to edit. It is **impossible to be configured**, but **it does validate**

against the grammar. Supports only the windows platform and there is NO documentation.

XML Authority 1.2 by TIBCO eXtensibility

It has a **nice interface** compared to the others that **can also be configured**, but **it doesn't validate** against the grammar and supports only the windows platform.

5.1.3 Conclusions

The result, we have to admit, is quite disappointing. An editor that fulfills all our requirements doesn't exist for the moment. At this time, XML editors do not tend to meet our criteria. Many are written in Java, which leads to some interesting, slow, and non-standard interface behaviors, but on the other hand ensures portability.

But we know pretty well that this situation will very soon change, as the successor of DTD grammars, XSchema will make easier the implementation of such tools. The major problem with DTD, is that it is not written in XML, while XSchema is. So until this moment comes, we have to choose the best among the ones that we have so far.

As a general conclusion, we would say that most of tree editors are inappropriate for editing text as the text areas provided are too limited. An obvious way around this problem would be to use a standard text editor with some XML plug-ins (like XEmacs in *psgml* mode). The problem is that there are critical features that text-editors don't tend to provide. For instance, the ability to see the DTD structure, to see what entities are permitted inside a document at a given point, or to validate the document prior to saving.

Leaders so far, seem to be **XMetal** and **epcEdit**. Both fulfill almost all of our requirements, but they still remain difficult the use by a novice user. And to be sincere, the ideal would be to have an editor that has in one the positive features that each one of them provides: the friendly interface and the possibility for configuration (but easier) of XMetal together with the possibility of epcEdit to view the structure of the grammar in a tree-like format and also the ability to verify and enumerate ID's, not to mention the portability that epcEdit guarantees.

So, the editors that would fit well with our grammar, seem to be XMetal, and epcEdit, even though they are both very expensive to use in education (around $500) and eventually the editor that had been used by STAF-Gina students who tested EVA was XEmacs in *psgml* mode. This choice was possible only because these students had already some knowledge of XEmacs.

5.2 EVA's commNcontrol portal

A very important part of EVA's environment is the "**commNcontrol**[1]" portal. *commNcontrol* is EVA's web portal for a list of projects. This is a web-based environment that serves as the cockpit of one or more projects that have a common feature that unites them. This can be a common theme, the same tutor, the same interests of tracking or even the same period of realization.

commNcontrol distributes over the World Wide Web the latest revisions of each group project, the tutors' audits and evaluations, discussions between the group and with the tutors and provides **file transfer**, **discussion** and **evaluation** tools.

[1] commNcontrol stands for Communication and Control.

The role of this console is to promote community building around a course, improve collaboration between group members and generally the project's community, motivate students to give regular feedback, facilitate collaborative editing, make students profit from peering others' projects.

This console facilitates the task of monitoring for the teachers by making the visualization more overt, thing that will make their "coaching" role even better. Also facilitates the evaluation by giving the chance to compare between projects.

Peering is much easier with all the gain that entails, including communication between team members and with tutors for better collaboration and feedback. Also the use of awareness tools for better coaching of the projects.

Technically speaking, *commNcontrol* is an implementation with Java servlets, XML as the format for storing information and in conjunction with XML parsers and writers and occasionally Cocoon from Apache XML Project as the XSL processor.

More precisely, the components of *commNcontrol* are:

5.2.1 Authentication/Customization

EVA can be accessed via an HTML (web) form (cf. Figure 8.) where someone can be authenticated either as tutor, either as student or as a guest (http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/commNcontrol.html). The role of this authentication is to customize the next component (portal commNcontrol) according to this authentication by providing different tools and/or different representations of information according to the user.

Figure 8. EVA's authentication form

5.2.2 commNcontrol - Content presentation (Dissemination of the data)

Figure 9. EVA's commNcontrol console for a tutor

According to "Genex, Memex, Codex… " to complete our environment we need to disseminate it to bigger audiences.

Leveraging from XML would be incomplete if we didn't get the maximum out of it. This can only be done by taking advantage of the fact, that XML **can and should** separate style and content. Thus, we provide different ways to visualize the information stored in the xml files depending upon the audience interested.

More concretely, we have created 3 views:

- One, for the tutor(s) of the project(s), that give the full presentation of the information of each project, followed by discussion and evaluation tools (cf. Figure 9.)
- Another one for the team members of each project, also with the full presentation of the information but with discussion and file transfer tools (cf. Figure 10.)

- And finally, one for third parties (called “guests” in our case) that might be interested, but where there is no interaction with the data (no given tools). According to specific case, we could think of it as a portal where some information that is considered private is missing and some other information is presented differently (though not in our case).

Actually, it’s about a summary list of all group projects and their important information like authors, project titles, project goals, work package progress plus the audits and evaluations given by the tutors implicated to the project. According to the authentication, the corresponding tools for each user follow (tutor or student).

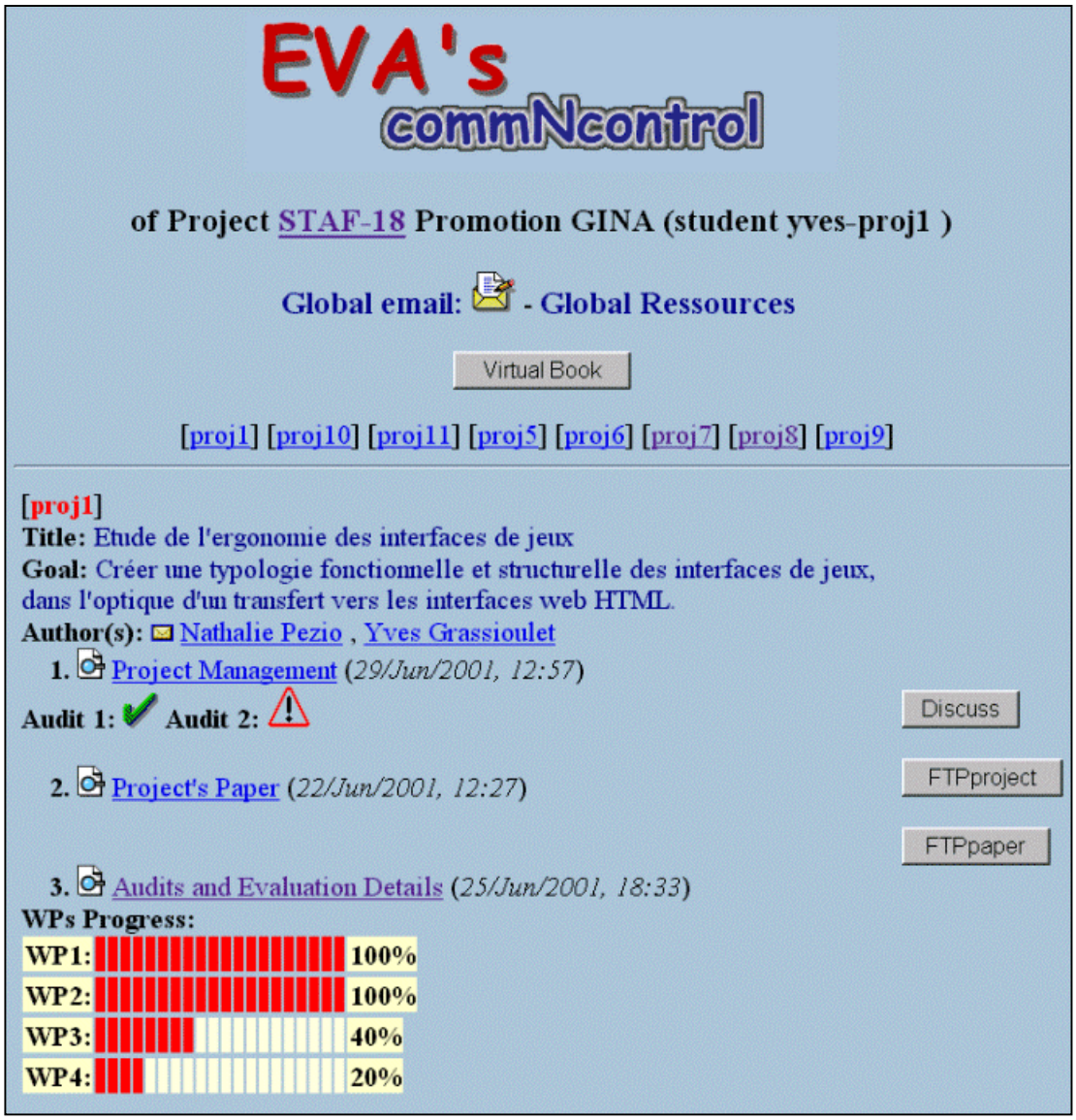


Figure 10. EVA’s commNcontrol for a group of students

5.2.3 Group and global mailing lists

Group and global mailing lists are also created dynamically and are present in this console so that communication at all levels can be guaranteed and facilitated.

So, a mailing list with all students and tutors that take part in the course is automatically created and is available through commNcontrol. Next to each project, there is also a small mailing list for the group members of each project. Both mailing facilities are represented by icons (cf. Figure 11.).

Figure 11. Mailing possibilities of EVA

Apart from the obvious advantages, what we want to emphasize and develop is **community building**.

Creation of online communities is one of the success stories of the Internet. The principle is simple: every actor in an information and communication system should have an active presence in it. That's why web portals or virtual environments are created to allow a community to be created. But that's not enough. There has to be made an effort to animate the site by a key person or group of persons (Schneider, 2001 [54]).

In our case, tutors use the above facilities to "coach" the projects by giving information concerning the projects, making proposals, encouraging the students, and motivating them.

A way to guarantee this kind of community building is, in our case, to split the project management process into phases and encourage the students to follow them. This ends by giving them a regular introduction at the beginning of each phase with information, resources, and advice and a regular feedback at the end of each phase.

5.2.4 Semi-asynchronous persistent in time discussion tool

Students and tutors of each project are also given through EVA the possibility to open a discussion concerning a question posed or a comment made for a specific part of their project.

More explicitly, a student or a group of students that work together on the same project can and should describe their questions or their comments inside their project management file immediately after the node that the question or the comment is about. For example, if the group has a question that concerns the main goal of their project, they only have to put an open question immediately after the node "main goal". An example can be seen in Figure 12.

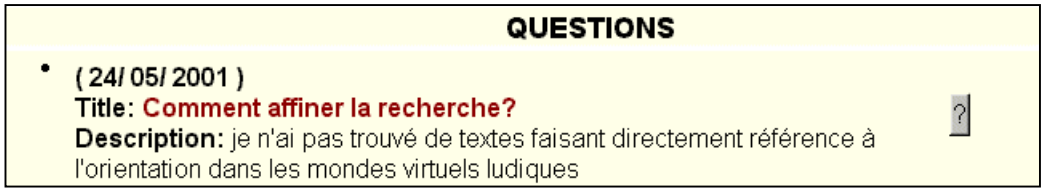


Figure 12. An open question posed inside EVA

A question is added to the project management description and a button is added next to it that allows either the tutors or the members of the group to open a discussion

around this question by giving a solution, an answer, a suggestion to it or by argumenting the answers given to it. For this reason, a web form is given that allows the interested parties to add an entry to the discussion and formulate it in rich text format (XHTML) (cf. Figure 13.).

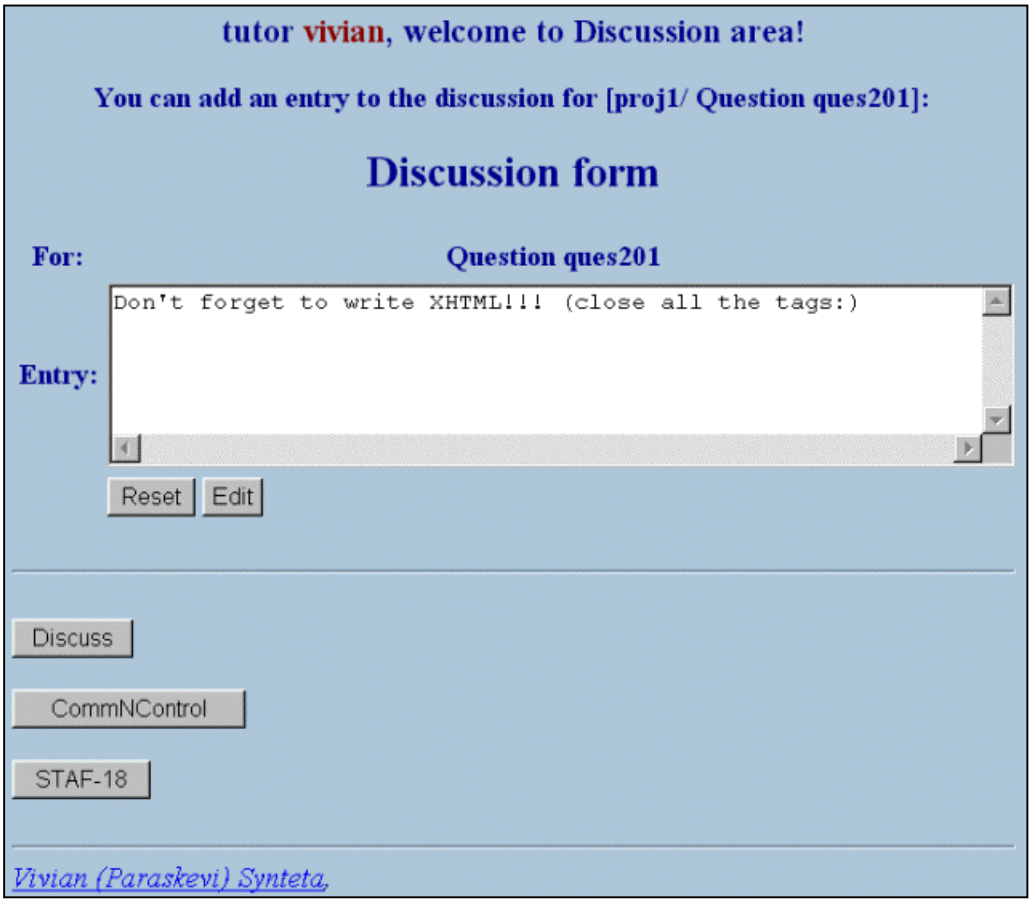


Figure 13. EVA's discussion form

When an EVA's actor (student or tutor) adds a comment to a discussion it will appear among the other information concerning the project's management along with it's metadata (author, date, time), and it will be stored so that everyone will be able to follow this discussion whenever they want to (cf. Figure 14.).

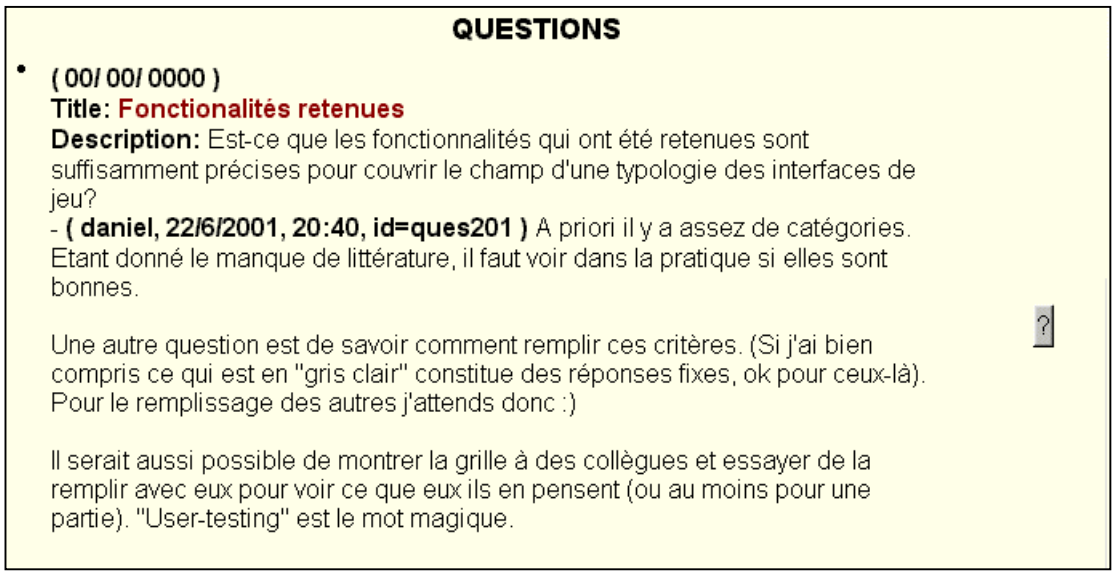


Figure 14. EVA's discussion

5.2.5 File transfer tools and versioning system

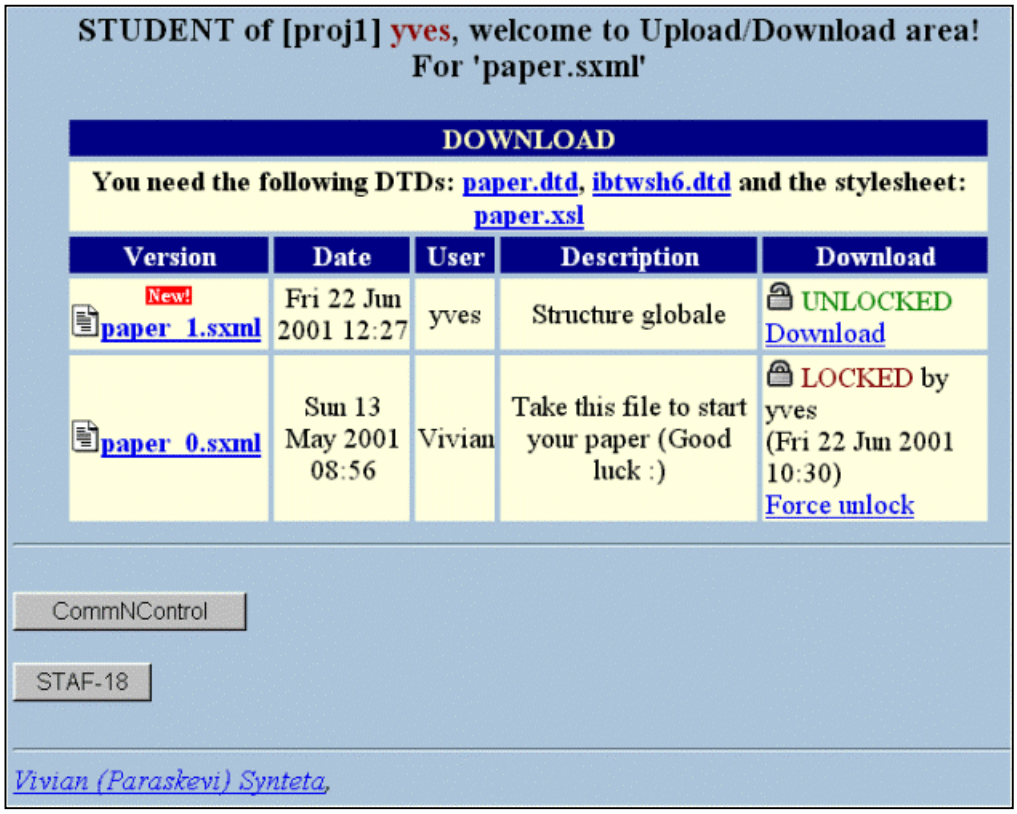


Figure 15. EVA's file download area

EVA also provides through the *commNcontrol* console file transfer tools that are enriched by a simplified versioning system.

As the students have to edit regularly 2 xml files (project.sxml and paper.sxml), one for the project management and one for the final paper that describes their project and

moreover, most of them work in groups, a versioning system followed by file transfer tools seemed absolutely important to be present in EVA.

So, we implemented a file download area where each group can take an initial version of the two files needed to begin with (along with grammars and stylesheets to work locally) (cf. Figure 15.).

And of course, a web form for uploading files, and through which, students can deposit their latest versions inside the versioning system of EVA (cf. Figure 16.).

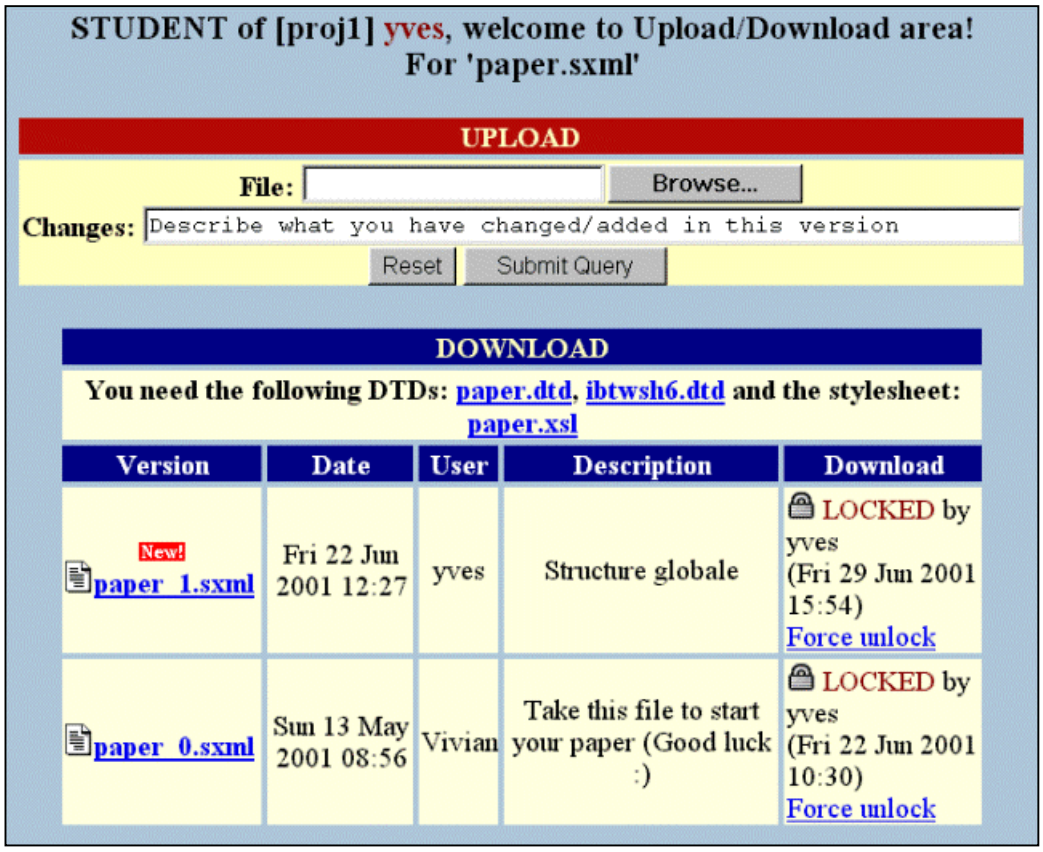
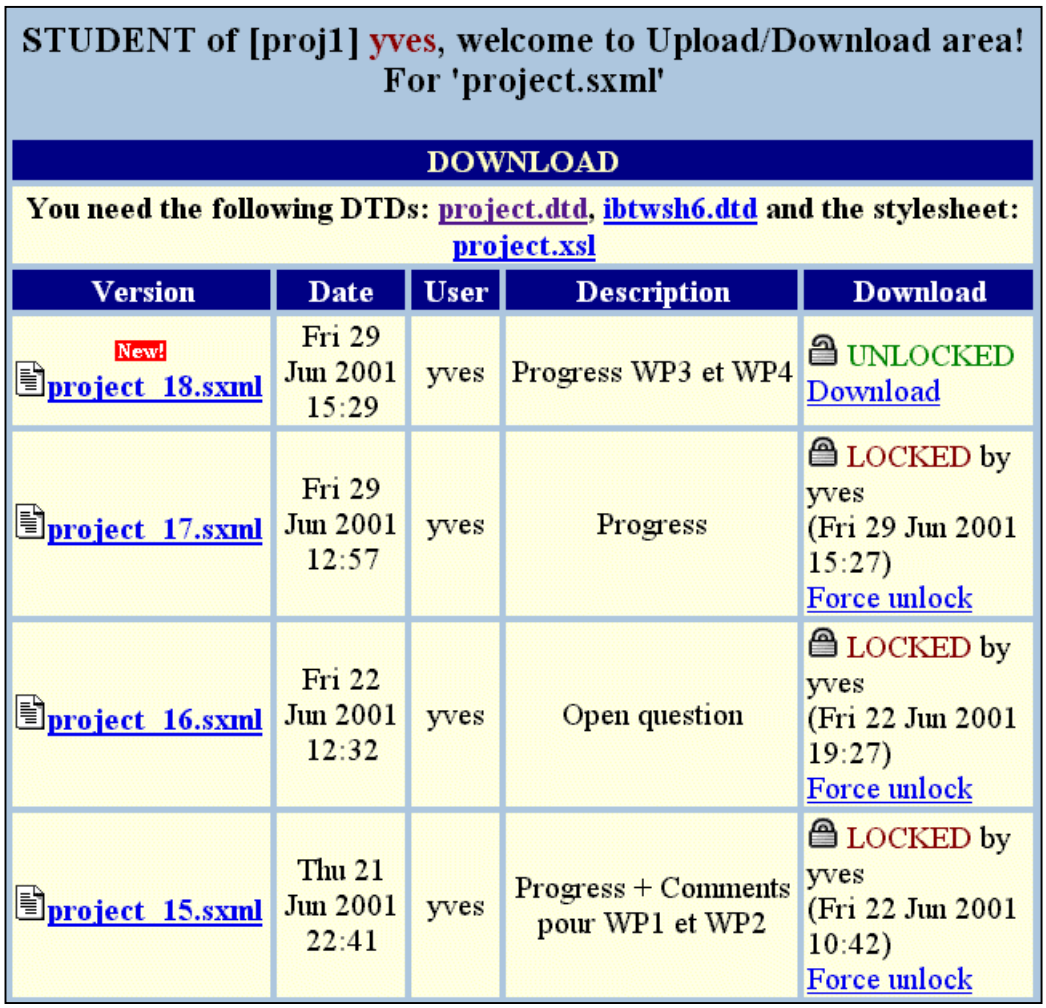


Figure 16. EVA's file upload form

Both file transfer tools (upload and download) are based on a versioning system implemented specifically for the needs of EVA. Every version is "locked" under the name of the student that has downloaded and is working on it. "Unlocked" are the versions that are permitted for download, and there is always the possibility to "force unlock" in extreme cases of abusiveness of the system.

EVA's versioning system automatically numbers every version and keeps track of all it's metadata like author, date, time and short description, plus "lock" information. An example is given in Figure 17.



STUDENT of [proj1] yves, welcome to Upload/Download area!
For 'project.sxml'

DOWNLOAD

You need the following DTDs: project.dtd, ibtwsh6.dtd and the stylesheet: project.xsl

Version	Date	User	Description	Download
New! project_18.sxml	Fri 29 Jun 2001 15:29	yves	Progress WP3 et WP4	UNLOCKED Download
project_17.sxml	Fri 29 Jun 2001 12:57	yves	Progress	LOCKED by yves (Fri 29 Jun 2001 15:27) Force unlock
project_16.sxml	Fri 22 Jun 2001 12:32	yves	Open question	LOCKED by yves (Fri 22 Jun 2001 19:27) Force unlock
project_15.sxml	Thu 21 Jun 2001 22:41	yves	Progress + Comments pour WP1 et WP2	LOCKED by yves (Fri 22 Jun 2001 10:42) Force unlock

Figure 17. EVA's versioning system

5.2.6 Audits and evaluations tools

Another important component of *commNcontrol* are the tools provided to the tutors that allow them to give specific audits and eventually evaluate each project. For this reason, corresponding web forms are provided to annotate them and attribute them to each project. Again, the possibility for the annotation is a rich text format (XHTML) giving the ability for the tutors to suggest URLs, emphasize, etc. (cf. Figure 18.).

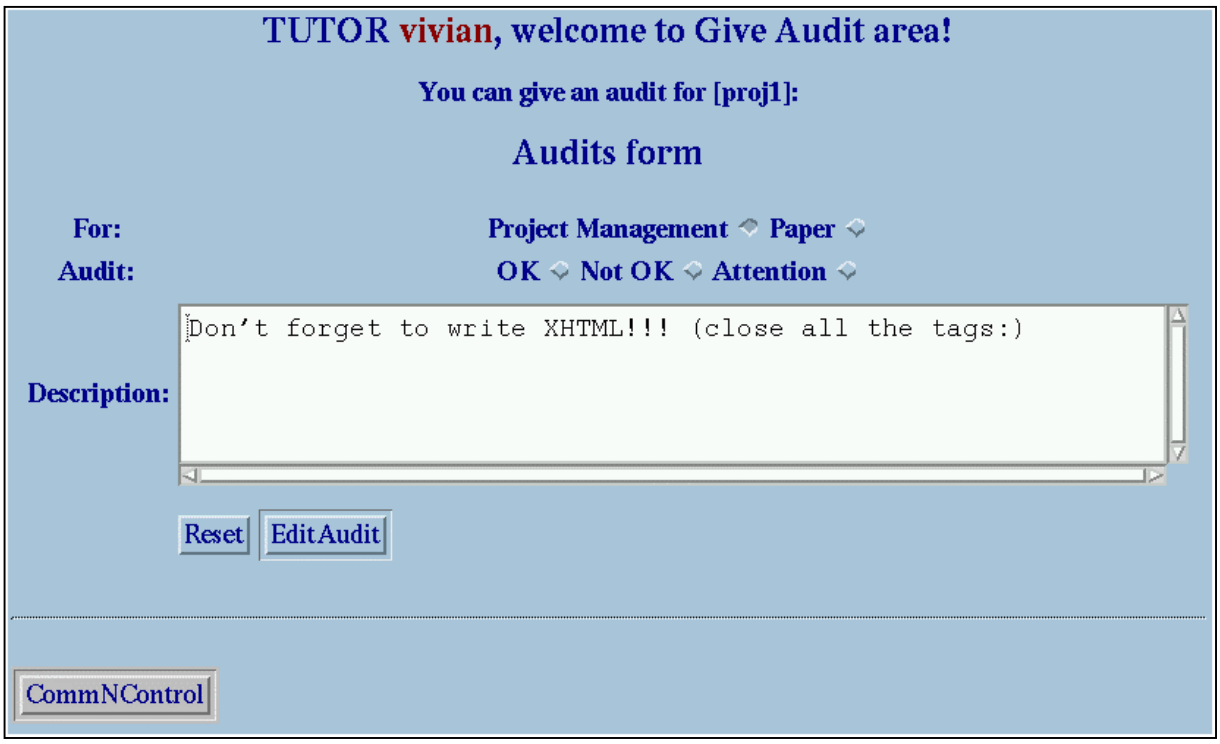


Figure 18. EVA's audit form

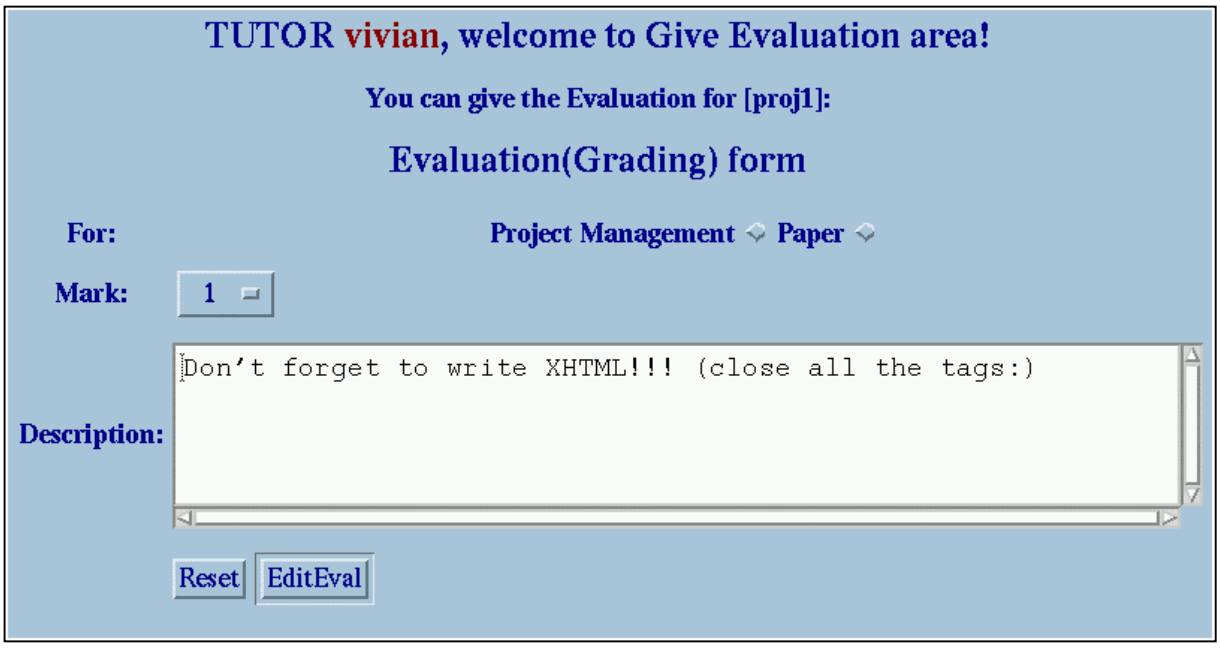


Figure 19. EVA's evaluation form

Both annotations appear inside *commNcontrol* among the other information presented in a form of awareness tools using specific icons. Also in this case, metadata are present and an extended description of all evaluations is given in a separate file (cf. Figure 20.).

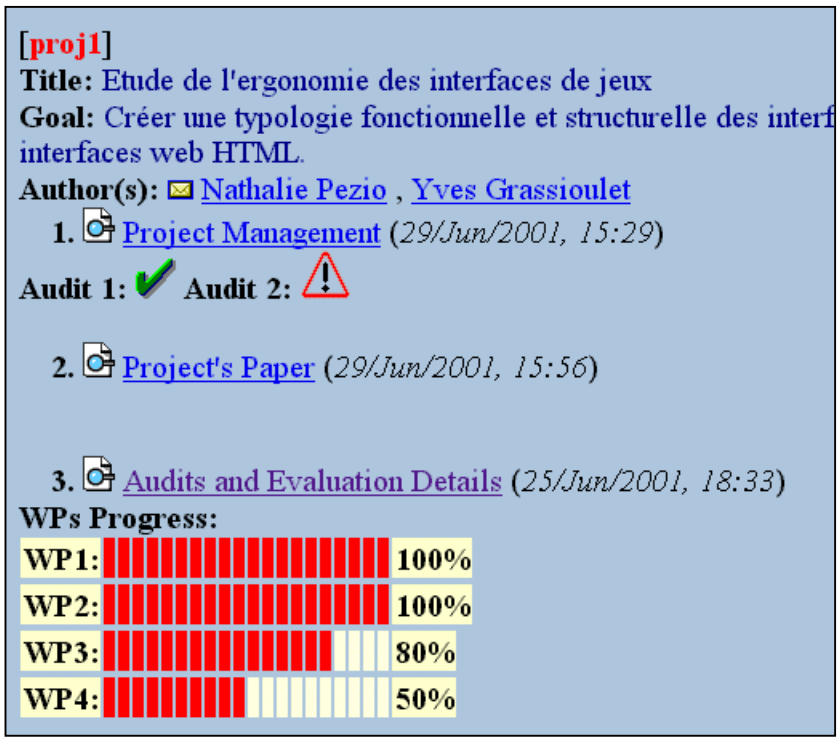


Figure 20. EVA's evaluation and audit results

5.2.7 Project management and Paper stylesheets

Of course, a complete view of the project management for each group is given through a link inside the *commNcontrol* that opens a new browser window. This new window, presents all the information of the latest version rendered with a stylesheet made for it called "project.xsl" leveraging from XSLT.

In the same way, another stylesheet called "paper.xsl" renders the final paper written for each project.

Both help the visualization of the information in a profitable way and one example for each is given in **Erreur! Source du renvoi introuvable.**, 8.3 of Appendix A of this Master thesis.

5.3 EVA's Virtual Book

Finally, EVA is complemented by a web portal (cf. Figure 21.) that hosts all the papers that each group has written (http://tecfa.unige.ch/staf/staf-e/paraskev/servlet/journal). It plays the role of a "Virtual Book" through which one can access the results of each project. It's about a summary list of all the papers written and uploaded to EVA, merged with information and metadata taken from the project management where they already exist, and creating complete scientific articles on the spot.

With this, global dissemination of final project results is guaranteed.

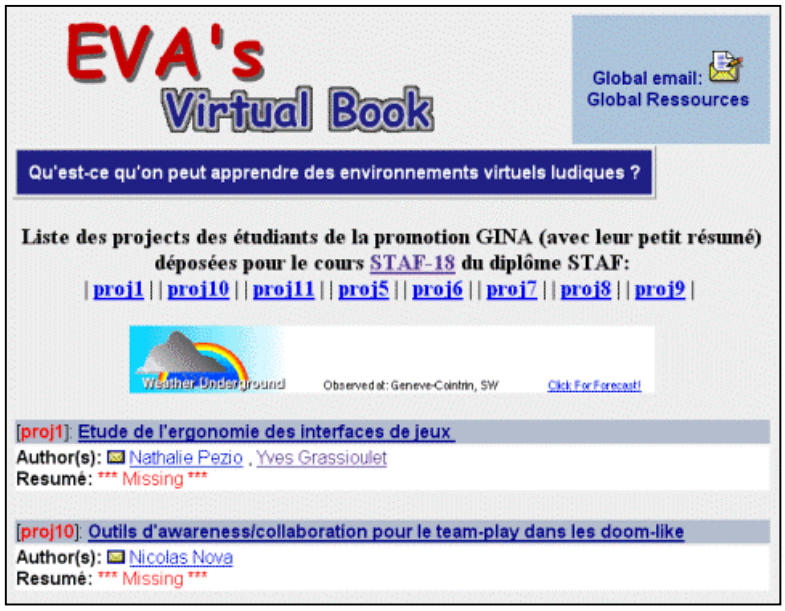


Figure 21. EVA's Virtual Book

5.4 Implementation

5.4.1 Java Servlets

To fulfill the purpose of the current thesis, eleven (8) Java servlets and two (2) java classes have been implemented, a java class is called to write back a DOM tree, a java class is imported and modified from O'Reilly and a piece of code was reused:

1) Authenticate.java; a java servlet that authenticates the users to log to EVA.

2) DOMWriter_vivian.java; a java class by The Apache Software Foundation, which writes back an xml file from a DOM tree. The version that is used for EVA has been slightly modified by the author. An important change is that instead of calling this class from the command line, it can be called through a servlet and the output can also be a file, instead of the terminal, as was the case in the original version.

3) Discuss.java; a java servlet that creates a web form for discussion and writes the data to the *discussion.sxml* file.

4) DownloadFile.java; a java servlet that downloads a file, locks it under the name of the user that has downloaded it and stores the information to *versioninfo.sxml* or *versioninfo2.sxml* file.

5) FTP.java; a java servlet that transfer files (*project.sxml* and *paper.sxml*).

6) GiveAuditEval.java; a java servlet that creates web forms for giving audits or evaluations for the tutors and stores the information to *evaluations.sxml* file.

7) UploadFile.java; a java servlet that uploads a file, renames it after the versioning system, unlocks it so that any member of the project's group will be able to download it and stores the information to *versioninfo.sxml* or *versioninfo2.sxml* file.

8) commNcontrol.java; a java servlet that parses several xml files and creates the web portal **commNcontrol** of the course by combining interesting information from all the projects.

9) cvs.java; a java class that takes care after the versioning system of EVA and shows all the existing versions that have been uploaded so far and their metadata (if locked and by whom and when, if unlocked or if is being forced to unlock and by whom and when).

10) journal.java; a java servlet that parses the two files that the students edit (*project.sxml* and *paper.sxml*) and creates a web portal called **Virtual Book** that gives access to all papers produced and presents a list of all summaries of the papers.

11) pass.java; a java class that encrypts the passwords of all users that have access to EVA.

12) A java class called “com.oreilly.servlet.MultipartRequest” by Jason Hunter and Geoff Soutter; this utility class is imported to handle multipart/form-data requests, the kind of requests that support file uploads. This class emulates the interface of HttpServletRequest, making it familiar to use. It uses a "push" model where any incoming files are read and saved directly to disk in the constructor.

13) and a piece of Java code that is imported that triggers the downloading of a file by simply clicking on a hyperlink (found in http://lists.w3.org/Archives/Public/www-jigsaw/2000NovDec/0064.html).

5.4.2 XML files

The xml files that support EVA_pm method and that are automatically copied inside every student’s workspace the first time that is logged to EVA are:

- project_0.sxml; the initial version of the file for project management that the students take and edit.
- paper_0.sxml; the initial version of the file for the final paper that the students take and edit.
- fullproject.sxml; the xml file that combines *project.sxml* and *discussion.sxml* (using Xinclude of Cocoon).
- text.sxml; the xml file that combines *project.sxml* and *paper.sxml* (using Xinclude of Cocoon).
- authentication.sxml; the xml file that stores the authentication information of all users of EVA.
- evaluations.sxml; the xml file that stores all audits and evaluations for a project.

5.4.3 Technologies used

The technologies that have been used for EVA are:

The following **XML Infoset:**

DTD (Document Type Definition); to create the grammars that define the xml files use.

XML; the format to stock the data (standard of exchanging data for the interconnection of the following tools).

XSLT; to transform the xml content to HTML (in order to visualize the content through the existing HTML-compliant browsers).

Cocoon (Apache XML Project); it's XSL processor (Xalan) to transform in HTML the xml files.

Java servlets; Servlets are the Java platform technology of choice for extending and enhancing web servers. Servlets provide a component-based, platform-independent method for building web-based applications, without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are server- and platform-independent. Servlets have access to the entire Java API and receive all the benefits of it (portability, performance, reusability and crash protection). A servlet can handle also, multiple requests concurrently, and can synchronize requests.

SAX and DOM API; to parse the XML files and update them (SAX for event-driven parsing and DOM for tree-constructing parsing)

5.4.4 Awareness tools

In order to provide the maximum scaffolding, there are several awareness tools that are used inside EVA's environment. The ones that are worth mentioning are the following two:

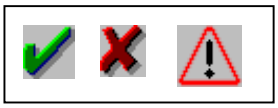


Figure 22. Icons for tutor's audits

The first one is a collection of three icons (cf. Figure 22.) that have been created to correspond to the audits. The green tick appears when everything is ok, the red cross when the audit is negative and the attention sign appears when there is a specific point that should be taken into consideration before continuing the project. The second is a table of progress bars that shows the progress of the work packages of each project. An example is shown below (cf. Figure 23.).

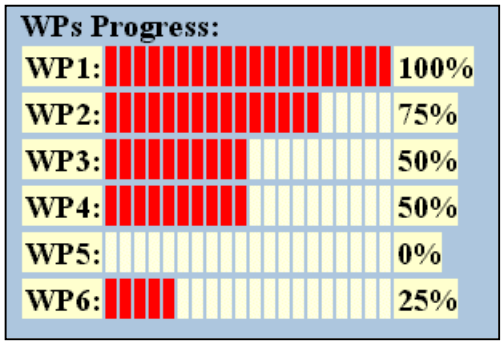


Figure 23. Progress of the work packages

5.4.5 Putting all of them together …

EVA's environment has been designed in the following way:

Authentication. For the authentication to EVA's environment, all possible users' information has been stored in an xml file called "authentication.sxml" that conforms to the corresponding grammar (authentication.dtd). In this file, tutors, groups and guest accounts are created. As xml files are simple ascii files, there is a security problem in storing in them secret information, like passwords in our case. To surmount this problem, a Java servlet has been implemented with the name "pass.java" that uses a one-way encryption algorithm so that the passwords of every user will be transformed (encrypted) into a long chain of characters that is impossible to be transformed back to the original password. A stylesheet (authentication.xsl) with the processing of Cocoon can give an HTML presentation of the whole project, it's tutors, it's working groups and the possible guests allowed to access EVA. The following diagram (cf. Figure 24.) shows this structure:

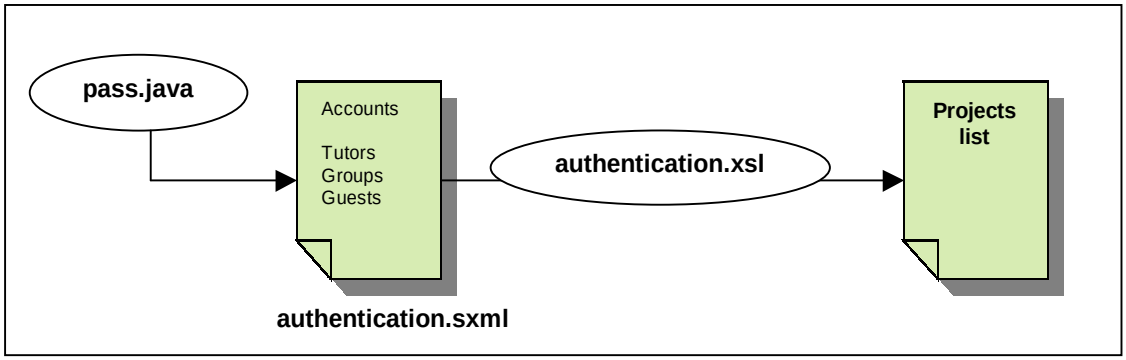


Figure 24. Diagram of EVA's Authentication structure

EVA_pm environment. Access to EVA for STAF-18 students is done through a web page (gestion.html), which is part of the STAF-18 web site. From this page, one can either authenticate and access the whole environment of EVA or go directly to the Virtual Book as guest.

Virtual Book. This is the web portal for the papers produced in the framework of the course. A java servlet called "journal.java" parses both the *project.sxml* and *paper.sxml* files of each group, extracts some information (title, authors and summary) and makes a list of the papers produced. Through this web page, each paper can be accessed by calling a file called *text.sxml.* This is actually an xml file that includes both project.sxml and paper.sxml and is formed using Xinclude technology of Cocoon. In this way, interesting information can be recycled from the project management file and included inside other files, such as authors' information, references, etc. From Virtual Book access to the rest environment can be done, but only as a guest, with limited interaction with the environment.

EVA's commNcontrol. The authentication to EVA starts from a web page called "commNcontrol.html" that is a simple HTML form where the user has to define whether he is a tutor, a student or a guest and log with his personal login and password. The form is treated from a java servlet (authentication.java) that parses the "authentication.sxml" file and searches for a user with the same data. Once, the authentication has been done successfully, the same servlet checks whether it is the first time that the group of the user has been logged to EVA. If it is the first time, the servlet initializes his working space. That means that it creates a new directory for his

project and puts inside it the initial files that are needed to start working in the project. These files are:

1) *project_0.sxml*; the initial version of the project management file

2) *paper_0.sxml*; the initial version of the paper file

3) *versioninfo.sxml*; the file that stores all versions of *project.sxml*

4) *versioninfo2.sxml*; the file that stores all versions of *paper.sxml*

5) *discussion.sxml*; the file that stores all discussions

6) *text.sxml*; the file that includes both project and paper files

7) *fullproject.sxml*; the file that includes project and discussion files

8) *evaluations.sxml*; the files that stores all audits and evaluations

In either case (first time or not), the next step is to pass to the web portal of EVA. It is the commNcontrol.java servlet that creates this portal. It is called **commNcontrol** and stands for **Communication and Control,** as it is the monitoring and communication cockpit of the course's projects. According to the authentication of the user, there are different views of commNcontrol that appear.

1. **Common view.** The things that are common to all types of users (tutor, student or guest) and that come from the parsing of several different xml files are:

- **Global mailing lists;** emails are parsed from *project.sxml* files and a mailing list is created of all the students and the tutors that participate in the course.

- **List of all projects;** in the same way and from the same xml file, group members, group mailing list, project title and project goal are presented.

- **Access to each Project management;** a detailed presentation of the latest version uploaded to EVA of the *project.sxml* using it's corresponding stylesheet "project.xsl" (Cocoon processing) is offered.

- **Access to each final Paper produced;** similarly, a detailed presentation of the latest version of *paper.sxml* using it's corresponding stylesheet "paper.xsl" (Cocoon processing) is offered.

- **Access to each projects Audits and Evaluations;** a detailed presentation of the *evaluations.sxml* using it's corresponding stylesheet "*evaluations.xsl*" (Cocoon processing) is offered.

- **Awareness tools;** progress bars that refer to the workpackages, icons for quick browsing of audits and notes for quick browsing of the evaluations are shown for every project. These are created by parsing the corresponding information from the xml files and by transforming it so that it can be better visualized.

2. **Student extras.** Students share several tools apart from the common view:

- **Discussion tool;** this tool gives access to the "fullproject.sxml" file that, with the homonymous stylesheet "fullproject.xsl" and the processing of Cocoon, creates a web page (actually a web form) where the whole project management is presented

and the “discussion.sxml” is also included. That means, that next to each open-question or comment that is inserted into the project (taking into consideration that it has to have an id unique), there is a button that leads to a web form that will add an entry to the chosen discussion. Every entry to a discussion is stored in the “discussion.sxml” file together with its metadata (author, date). The storing is done with a servlet called “DOMWriter_Vivian.java” that parses the latest version of the discussion file, adds a new entry to it and rewrites the file.

- **File Transfer Tools (2);** EVA also has file transfer tools and a versioning system that supports them. So, for the two files that the students have to edit, *project.sxml* and *paper.sxml*, and given the fact that sometimes a project is being shared by a group, there are two environments to upload and download the different versions produced. The “FTP.java” servlet takes care of the whole procedure but there are several other servlets and classes that complement it. For the versioning system the “cvs.java” class stores in an xml file (versioninfo.xml or versioninfo2.xml according to whether it is about the project file or the paper file) all the information about every version uploaded to EVA, and the storing is being done with “DOMWriter_Vivian.java” class. Two other servlets “UploadFile.java” and “DownloadFile.java” create the web forms to upload a version or to download one. Especially for the file uploading, a class has been imported called “com.oreilly.servlet.MultipartRequest” by Jason Hunter and Geoff Soutter.

3. **Tutor extras.** Tutors also share several tools apart from the common view:

- **Discussion tools;** this is exactly the same tool that is offered to the students

- **Audits tool;** this tool uses the servlet “GiveAuditEval.java” to create a web form to edit an audit. The audit information is stored inside the “evaluations.sxml” file using the “DOMWriter_vivian.java” servlet and appears immediately afterwards in the commNcontrol.

- **Evaluations tool;** in a similar way to the audits tool and with the same servlet, a web form opens to edit an evaluation that will appear in the commNcontrol immediately afterwards.

The following diagram (cf. Figure 25.) gives a global view of the organization of EVA that is described above:

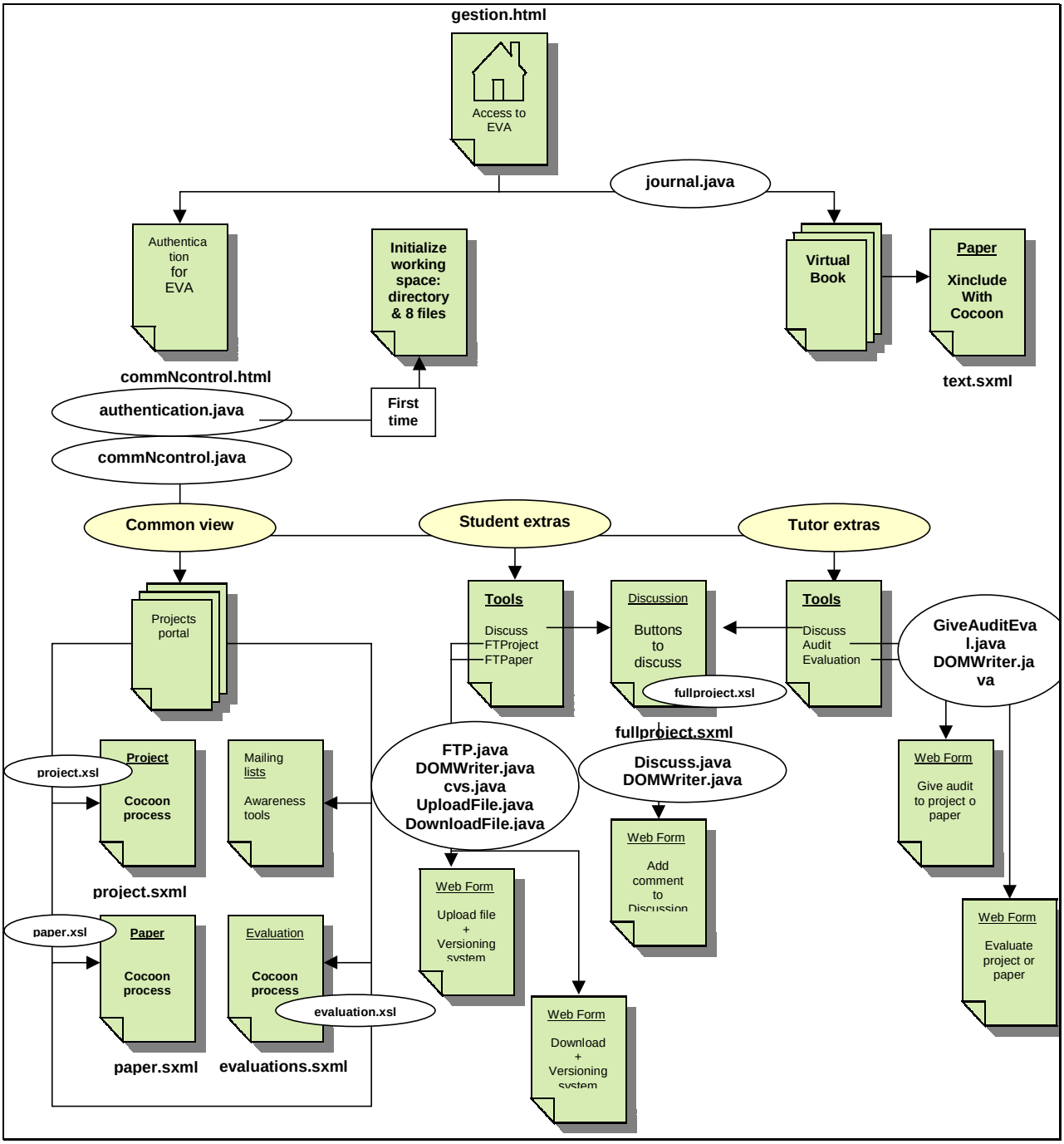


Figure 25. Diagram of EVA's commNcontrol and Virtual Book structure

Chapter

6

6 Testing and Evaluation of EVA

This chapter presents the procedure that has been followed in order to test EVA _pm with students, how a preliminary evaluation has been conducted and what did the results show.

6.1 Testing EVA

The testing group that was chosen for EVA was the 12 students of the STAF Masters diploma in Educational Technologies at TECFA, University of Geneva, Switzerland (promotion “GINA”).

The course is called STAF-18 and the responsible tutors are Dr. Daniel Schneider and Paraskevi (Vivian) Synteta. The context was the study of Multi-user Video Games in order to extract useful lessons for education (Title in french: “Qu'est-ce qu'on peut apprendre des environnements virtuels ludiques ?” cf. For more info see §1.5)

Students in the STAF diploma come from many disciplines, though especially from Psychology and Education. The reason that this group was chosen is that the students have technical skills in web technologies, and more specifically, had already some basic knowledge of the XML Infoset. As the proposed method (EVA_pm) entails for the moment the difficulty of editing an XML file, it would have been difficult to use this method with a different group that had no knowledge of XML. This could have been possible only in the case that there was time within the course dedicated to training on XML editing. In our case, students already knew to some extent, how to read a DTD.

The procedure that was followed in order to present the method to the students is presented below. The results of a preliminary evaluation concerning this testing are given immediately after.

6.1.1 Choose a project, make groups

In the beginning of the course, students have to choose a project from the given list of STAF-18 or to propose a new one with a similar context, and then they were asked with whom they wanted to work with (optionally) and preferably to pair with. The pairings were friendship pairings. Use of pairing by friendship is based on the fact that students are more willing and motivated to cooperate when they respect their peers and really wish to help them. Pairing by cognitive criteria (e.g., ability) can sometimes be frustrating if the pairs are too heterogeneous (Puntambekar, 1997 [40]).

Students send an email to EVA’s administrator with the following information:

Project	ex: proj3
Fullname	ex: John Smith
Unix login	ex: smith
Password	ex: secret

In this way, a student of staf-Gina is able to log to EVA's *commNcontrol* with his unix login and the password that was chosen. All the above information is stored in an xml file called "authentication.sxml" with the slight difference that the password is being encrypted using a one-way encryption method to surmount the security deficiency of xml.

6.1.2 Presenting EVA

Immediately after, a session was dedicated for the presentation of EVA to the students, where the goal of it and a short description of the state of the art that supports the method were given.

This presentation helped students appreciate and better understand why they were going to use this method and what was the work that was expected from them. That immediately created a positive attitude and atmosphere as the first reactions of the students were very enthusiastic. They asked more information about it, saying that they found it very interesting.

A web site, also, supported STAF-18 where all the information about the course was available. Among this information, the site provided documentation about the grammars used for the course. There were three different types of documentation available, the two ones were in HTML form and were online and the third one in Microsoft document format (cf. 4.2.4). Documentation was important for the students to understand the grammar and it's structure.

Along with this presentation, a familiarization of EVA's toolset followed using test accounts and hypothetical groups.

A detailed list of all the possible uses of EVA according to authentication was presented as follows:

A student can:

- Download the latest version of project or paper (if it is unlocked),
- Upload the latest version of project or paper that has just edited (if it was he that had downloaded the previous one),
- Discuss with the other members of the group or with the tutor and follow all discussions that have taken place,
- Verify whether the xml editing was error-free (if it doesn't see any error messages - java exceptions),
- Browse the projects and papers of the other groups through "commNcontrol",
- Communicate with the other members of the group or with all the students of staf-Gina that share the project STAF-18,

A tutor can:

- Give audits and evaluations to each project,
- Discuss with the members of each group or follow all discussions that have taken place,

- Monitor the projects and papers of all groups through “commNcontrol”,
- Communicate with the members of each group or with all the students of staf-Gina that share the project STAF-18,

A guest can:

- Browse the projects and papers of all groups through “commNcontrol”,
- Communicate with the members of each group or with all the students of staf-Gina that share the project STAF-18,
- Find an interesting paper by consulting the abstracts in the “Virtual Book” of STAF-18.

6.1.3 Download initial versions

In consequence, students were prompted to download the initial versions of the two files (project and paper) along with their grammars and their stylesheets to work locally. That served as a first attempt to see whether there was any problem with the interface under the observation and guidance of the tutors. In this first testing there were no specific problems reported.

6.1.4 Follow phases

A web site supported the course with all the information needed (website http://tecfa.unige.ch/tecfa/teaching/staf/staf18/welcome.html). This included, a page with complete documentation for the two grammars of the method, and a page for the course management (“Gestion des projets”). This latter one provided access to EVA and suggested the phases that should be followed (http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/gestion.html).

As coaching for PBL approaches is considered to be crucial for their success, the choice of splitting the course into phases was made. Projects had to be composed of iterative cycles where the intervention of the tutors was needed to pass from one to the next.

The phases were the following:

- **Phase 1:** Choose a project from a list of suggested themes and make groups
- Audit(s) (orally)
- **Phase 2:** Feasibility study and sketch of the objectives
- Audit(s) (orally)
- **Phase 3:** Draft version of the project’s specification
- Audit(s) (through EVA’s commNcontrol)
- **Phase 4:** Final specification (with a detailed description of the work packages)
- Audit(s) (through EVA’s commNcontrol)

- **Phase 5:** Work on the project according to the defined work packages
- Audit(s) (through EVA's commNcontrol)
- **Phase 6:** Results – Redaction of the "paper" – Meta-reflection on the work
- Short audit (for "last minute" corrections) - Final evaluation(s) (through EVA's commNcontrol)

6.1.5 Editing xml files

It was clearly explained to the students that eventually they should edit two (2) xml files (project.sxml and paper.sxml). For administrative reasons, the XML editor preferred for EVA, XMetal, wasn't available, so the editors that proposed to the students were either XEmacs in psgml mode (that students were already familiar with) or a freeware tree editor called Merlot that could be downloaded from the web.

It has to be explained here that the public chosen to test EVA is a specific public with a background in web technologies, so the above editors didn't pose an important problem. Normally, for end-users a professional editor like XMetal or epcEdit should be suggested, as the cognitive load of editing will be diminished.

So, both editors were presented and short instructions were given using examples. The next step was to upload the new version using the file transfer tools of EVA.

6.1.6 Coaching

Apart from the phases that were proposed to the students that guaranteed "coaching", regular email (using the mailing facilities of EVA) was used to prompt students to criticize EVA and it's interface, to give them hints and help, to clarify ambiguities and to give examples.

But the most important part of the "coaching" was regular visits by one of the tutors in the computer room where several of the students were working in order to discuss with them, debate and, in general closely follow the procedure.

6.2 Preliminary findings for EVA

The goal of this thesis eventually was to have an initial product (version1.0 of EVA_pm). That's why a preliminary evaluation was needed in order to keep track of all the possible problems and have an important feedback for later versions.

So, the testing group was the 12 students of the postgraduate diploma in Educational Technologies STAF for the course STAF-18. There were 9 men and 3 women from which 6 with background in Psychology, 2 in Education, 1 engineer, 1 mathematician, 1 lawyer and 1 with a diploma in Cognitive Sciences. The mean age was 29. More precisely, all of them were in mid-twenties apart from two students that were both 49 years old. They all have some technical background but not much (especially in programming).

A little bit less than half of them (5 out of 12) had already a course in methodology during their basic studies. But none of them had any previous knowledge in Research Design. We have to say also, that the target group for testing EVA had already been given an introductory lecture on project management and design in the beginning of the course STAF-18.

Some of students worked in pairs, and some worked alone. So eventually, we had 8 projects of which 4 from pairs and 4 from individuals.

An important also information is that the course STAF-18 is always at the end of the academic year, so the students are pretty tired and overwhelmed from all the projects that have accumulated and rest to be done. Moreover, the course is quite demanding in both aspects: technically and conceptually.

The evaluation of EVA_pm is based on the problems that students and tutors face during PBL and that have been outlined in the introduction of this thesis. So, the goal of this evaluation was to verify whether or not the proposed environment helped them to overcome these problems and to what extend.

There are two things that have been evaluated. On one hand, **the learning environment** and on the other hand, the **students learning** on research design.

6.2.1 Evaluation of the LE

Evaluating the learning environment meant two things: **evaluate how useful it was** and **evaluate how ergonomic it was**. The user groups of the LE were the students and the tutors. For this reason, the following lists, one for each group, have been outlined with the things that had to be measured.

From the students view:

- General satisfaction / appreciation
- Usefulness – long term scaffolding on research design
- Usefulness of the given tools
- Reflection
- Managing time
- Collaboration and feedback
- Awareness
- Resourcefulness
- Motivation from peering or coaching
- Ergonomics, i.e., the usability of the interface, difficulty of using XML

From, the tutor(s) view:

- General satisfaction – appreciation
- Usefulness of the given tools
- Awareness (monitoring)
- Scaffolding / support
- Ergonomics, i.e., the usability of the interface

For the evaluation of the above, different approaches were used. First, recording of **immediate feedback** while the project was running from both students and tutors. So, comments, problems, suggestions, complaints have been gathered. Also, by the end of the project, **an online questionnaire** (cf. 9.1) was given to the students and useful deductions were made. Moreover**, the monitoring of the projects** (through EVA's interface) all the way through to their accomplishment, leads to many interesting conclusions and interpretations. Finally, the **tutor has been interviewed** to give his evaluation for EVA_pm concerning his role in it.

A detailed analysis of these results can be found in the section that follows.

6.2.2 Evaluation of the student learning

Evaluating the students' learning is a quite difficult thing to do. And if we are talking about the learning of research design and project management, it is even more difficult. But still, we had to verify whether students acquired some knowledge on it and to what extent.

The first thing that tells a lot about it were the **results of the students' projects** and the most appropriate person to give such information seemed to be the tutor of the project. There were two strategies that were followed for this.

One strategy was study the tutor's "**Audits and Evaluations**" inside EVA's interface. The other was to **interview the tutor** to give a global evaluation on how coherent was their research design, and of course how concrete were their results (how well they correspond to their goals and objectives).

Apart from the tutor's feedback, **a mini case study** was given to the students to be done, within the questionnaire. The goal was to see whether students are capable organizing a small project in the same way that they have done inside EVA.

Tutor's audits and evaluation:

All the audits that were given to the students were **very positive** and in general the students didn't have any special problems. The most common remark was that some of the students were too optimistic in their objectives. They almost all had enough resources.

More precisely, the tutor had to choose between "OK, Attention and Not-OK" and was describing it by adding comments to his evaluation. So, the results were:

First audit: 6 out of 8 projects got "OK". 1 got "Attention" (because he was too optimistic on his project) and 1 got "Not-OK" because he didn't finish the research plan.

Second audit: 5 out of 8 got "OK". 3 got "Attention" because the tutor judged that they should reschedule their research and focus on specific points (they needed coaching).

Tutor's interview:

How coherent were the projects? According to the tutor, all the students (in their first version of the project) had a fairly well-defined project and a coherent research plan, apart from one group that refused to use the method in the beginning, but eventually did it. In the second version, they were able to adjust it even better. The final research plans were explicit enough and well done. So, the tutor's evaluation is that they all learned how to spell out a research plan.

How well they managed time? As for the time constraints, they all respected the deadlines (maximum delays were of one week late).

Did they have any big problems? Students didn't have any serious problems. Their main problem was dealing with structured information. There was one student that refused accepting it and almost all the rest had problems in using a grammar and understand it but they all eventually used the method and did their projects with it.

How good were the results? The results of the projects also show some information about students' learning. As they approach at the end but haven't all finished yet, it can be said that they are all in track and pretty good.

How soon they finished compare to other promotions? Students finished almost at the same time like the previous year that used the same method but without the implemented support. But there were no major interventions that needed to be done this year in order to put back in track the students. So, we can interpret this as that the well-defined research plan from the beginning, helped them to stay on track with their projects.

The case study:

In the questionnaire that was given to the students to fill in, the last question was about a case study that had to be done. They were asked how they would organize a usability study for the "commNcontrol" of EVA. They had simply to list the basic steps that would describe their study without going into more details.

The goal was to see how easy it was for them to re-apply the steps that they followed inside EVA_pm. The results are:

They were almost all very good in defining the work packages needed.

Not many of them defined their objectives before defining the work packages so that they can see whether they fit well to their objectives.

So, in conclusion we cannot actually say that they have learned how to make a complete research plan but that they have at least learned how to make a first sketch for a project. In order to get into details and learn more, they probably have to use more than once the method, which is totally normal for a subject quite demanding as it is learning Research Design.

6.3 Evaluation of the student's use of the LE

In this section, the evaluation of the students' use of EVA is presented for each of the variables defined before in this chapter. Information is separated according to the sources that is coming from. Immediate feedback from the students, tutor's evaluation by monitoring the projects and finally, the answers to a distributed questionnaire.

In the questionnaire, all the questions required an answer in a range between 1 and 6, by 1 being the minimum and 6 the maximum. They also had the opportunity to comment their rating immediately after in a field provided for it.

6.3.1 General satisfaction – appreciation

Student satisfaction is an important aspect of a learning environment, that's why we start with this preliminary evaluation to see how comfortable students feel with such a method.

Immediate feedback: EVA has been welcomed and people that heard about it or the students that eventually tested it, found that it is a very interesting idea and were excited about it.

Tutors evaluation by monitoring: Students were doing pretty well given the fact that they were quite tired (end of the year, too many demanding projects) but were very frustrated to edit the XML files, so they lost some of their initial excitement of the method.

Questionnaire analysis: After using EVA_pm method, the students found it pretty good (Mean = 4, +/- 1.13)[1]. Apart from one person, that was completely negative to the idea of having a structured grammar instead of being free to choose his own method to work (said that the structure is too constrained), the rest of the students were quite positive. In their comments say that it guided them, that it structured their work, it was easy to monitor the progress and was useful especially for the students that work by distance (a common case with students at STAF that come from different regions of France and Switzerland).

6.3.2 Usefulness – short and long term scaffolding

Immediate feedback: Students had contradictory opinions during their work with EVA. Some found it useful in the beginning and later on, got tired with all the difficulties that XML posed and some others were negative in the beginning (finding several features useless) and later after a discussion and a more detailed explanation they became fond of the method.

Tutors evaluation by monitoring: There was a scaffolding to the students and it can be claimed useful as one could see how the students were adjusting and regulating their work according to their specification inside EVA. Of course, it is difficult for the students to realize immediately that this was happening.

Questionnaire analysis: When students were asked whether EVA_pm helped them or not to be more efficient compared to other projects that there were no support, as it was expected, the results in the questionnaire showed this variation of opinions (Mean = 3.08 +/- 1.44, "Probably no"). In the same way, when they are asked whether the grammar helped them to define better their research they disagree (Mean = 3.25 +/- 1.54, "Probably no") but in this case the positive comments are quite explicit and concrete but the negative ones focus always in the difficulty of the XML and the only interesting point is that the grammar had to be presented in more details in the beginning (so to better profit from it). As for the scaffolding through the tutor's audits, the students found it useful and encouraging to their work (Mean = 4.42 +/- 0.90, "useful") and they all agreed.

[1] From now on in all the results that come from the answers that are given in the questionnaire, and inside a parenthesis, will be given the Mean, the Standard Deviation and the description to the answer that corresponds to the Mean.

6.3.3 Usefulness of the accompanying tools

The usefulness of the tools to support learning is crucial for the success of the learning environment. We need to understand whether the tools that we have designed are helping students in what we intend for them to accomplish. The tools that we refer to are the "Versioning System" with the "File Transfer tools", the "File Locking", the "Discussion tool", and the "Awareness tools".

Immediate feedback: The "locking" system for the files was totally useless for the students that worked alone. Unfortunately, the implementation didn't provide this distinction. Apart from this, the rest of the tools were used and appreciated from the students.

Tutors evaluation by monitoring: The tools inside EVA were implemented in order to solve several problems that have been studied in earlier versions of the method. So, by default, there was some usefulness in the idea of having them. By watching the students, it can be said that the given tools definitely helped them and can be verified by the fact that there was no file losses, no problems between the groups by sharing the files and working together (collaboration was good) and there is track of all the versions of the projects (regular feedback).

Questionnaire analysis: Students found the "**Versioning System**" of EVA useful (Mean = 4.17 +/- 0.72, "useful") and they all agreed (even the ones that were totally negative to EVA). They say in their comments that it's useful and secure to keep all versions so that one can go back and take an older version and keep working on it. It was also useful for the work in groups so that the members could visualize the changes that were made in the project by the others. Students working by distance were the ones that really appreciated it. The "**File Locking**" was even more appreciated (Mean = 5.00 +/- 0.76, "very useful") from the students that worked in pairs. The "**Awareness tools**" like the audits icons and the progress bars were found also useful (Mean = 5.00 +/- 1.21, "useful") and only one disagreed strongly without giving any arguments though.

6.3.4 Reflection

Reflection enhances learning, so it was important to see whether EVA made students reflect on their work or not and what they think about it.

Immediate feedback: As reflection happens only at the end of a task there is no immediate feedback to this point.

Tutors evaluation by monitoring: Students started to reflect on their work from the moment that they are asked to write comments inside the project management file. This auto-critique helped them look their work as a whole.

Questionnaire analysis: The first question posed, was whether or not it helped the students to see inconsistencies or missing elements by watching the structured document (with the stylesheet). They all agreed more or less (Mean = 4.33 +/- 0.98, "yes") and said that it was easy and clear to see what are the weak parts of their work and improve them and of course the missing elements and fill them in. As for the meta-reflection, by writing comments to criticize their work, the majority of them thinks that it helped them to reflect on their work and some not (Mean = 4.00 +/- 1.65, "useful").

6.3.5 Managing time

Managing time is an important factor for the success of a project. One of the most common problems of the students is that they underestimate the time needed and fail to finish eventually.

Immediate feedback: In the beginning students showed this tendency, but as they had to define inside their work packages how they split their work and with the coaching of the tutor were forced to respect their plan.

Tutors evaluation by monitoring: Students respected time constraints. Some needed guidance in the beginning to think in more realistic terms though.

Questionnaire analysis: In a question about whether EVA helped them to organize their research or not they are positive (apart from one case that was always negative) (Mean = 4.08 +/- 1.16, “Probably yes”). It is interesting to note here, that only the students that were positive commented their answer.

6.3.6 Resourcefulness

Resources are important for a project. EVA provided specific tags inside the project management file to store, sort and annotate them.

Immediate feedback: Half of the students welcomed from the beginning this feature and annotated all of their resources.

Tutors evaluation by monitoring: All the students used EVA for their resources. Half of them used them a lot and the other half only because they thought that they were obliged to. It can be seen that the ones that took profit of EVA for this feature have richer results.

Questionnaire analysis: In the questionnaire, students were asked whether they found useful or not to be able to include their resources inside the project management file. The majority found it very useful (Mean = 4.67 +/- 1.44, “useful”) while two students did not. They found it practical, useful also for the other groups (peers), and the possibility to annotate even more interesting. Another advantage, that they found important, is that it makes the resources available from everywhere as they are always on line.

6.3.7 Motivation from peering or coaching

Immediate feedback: Progress bars were the most favorite awareness tools that motivated students.

Tutors evaluation by monitoring: Students have been monitored how they were affected by following the progress of the projects through commNcontrol. Progress bars make them compete and helped to keep on with their project. Moreover, by watching the other projects, they were regulating their work and improving it when they found something interesting in other projects and were copying it.

Questionnaire analysis: There were two questions that were posed to them. One about how often they have looked the work of the others. Apart from the same one person, that was always negative, the majority said frequently (Mean = 4.25 +/- 1.29, “very often”). Some said that that it was the strongest point of the method and someone else that it is important for distant projects (like STAF projects). The second

question was about whether looking at the others progress motivated them or not. The majority again said yes (Mean = 4.00 +/- 1.35, “yes”). It is interesting the comment that some students pointed out, that sometimes if one is late with his work, it could be scary though …

6.3.8 Ergonomics - usability

The integration of the physical layout of the program (design features like interface, links and usability of tools) and the way in which navigation is conceptualized and implemented is an important aspect of learning in a WWW environment (Jones & Faquhar, 1996). As the students of staf-gina were having courses for usability testing and ergonomics of web interfaces, the feedback was rich and the comments very strict.

Immediate feedback: Students that tested EVA’s method (grammars and toolset) made a lot of comments and suggestions and the author is grateful to all of them. A quick summary of this unofficial feedback that was either in the form of direct conversation or as an email during the period that students were using EVA is given below.

For the **grammars** the comments were that it is too detailed, too structured and puts a lot of constraints. Suggestions for corrections to some nodes have been made (many “comment” inside “comments” following the structure of the rest nodes). Many students said that they found themselves repeating the same information several times for different parts inside the project management file. The author had to give an extended explanation and help them to realize eventually that, that characteristic was intended to help them decompose their thoughts and make them reflect on them. The feeling of the author is that she should have explained more about the method in the beginning. Some suggested that dates should be put automatically when they edited a node, as also as ID’s, thing that can be done with a more sophisticated XML editor (but still it entails the configuration or scripting of it). For the paper, a student suggested that the “body” should be splint into “theory” and “implementation”. Requests were made for internal table of contents and generally better navigation of the stylesheets for project and paper as they can get pretty long by the end of the project. Another suggestion that was made is that it would have been interesting to watch the latest changes in the stylesheets. A solution to this would have been to put a “New” icon in front of latest edited nodes given the fact that they have already put dates inside all nodes (which wasn’t the case in this testing).

Students found **XML editing** confusing using XEmacs and not a WYSIWYG editor but apparently this didn’t pose any serious problems if we consider that eventually they all preferred to use XEmacs instead of Merlot.

As for the **interface of the toolset**, suggestions to improve the appearance were made, ergonomics pitfalls were mentioned like changing the order of the buttons “Reset” and “Authenticate” in the authentication web form of EVA. Suggestions to include in *commNcontrol* the possibility to watch the work packages in the form of a Gantt or Pert Chart were made.

A **technical problem** with IExplorer for Mac was mentioned. It apparently refuses to download a file from the versioning system of EVA with the pretext that there is a mistake in parsing the XML file as it tries at the same time to open the file.

Tutors evaluation by monitoring: The toolset was pretty reliable as there were almost no problems during the testing period. The ergonomics though has to be

improved as the environment was complicated and implied different manipulations from the users to be done.

Questionnaire analysis: Students were very strict and rated the ergonomics of the web interface of EVA “fairly good” (Mean = 3.83 +/- 1.11, “fairly good”). They were a little bit troubled because, from their comments, one can see that using it wasn’t difficult (tools were quite well implemented) but the design wasn’t very practical. Some of the things that they suggest to be changed/improved is the navigation, the help on-line (e.g., by giving hints for buttons), take out elements that are useless (like “file locking” for persons that work alone), and give more explicit names to the buttons used. More precisely, they found the ergonomics of the “Versioning System” very good and they all agreed (Mean = 4.25 +/- 0.97, “very good”). The same for the “File Locking” (Mean = 4.50 +/- 0.93, “very good”). As for XML editing, students found it fairly easy (Mean = 3.92 +/- 0.79, “fairly easy”) even though they all preferred to work with XEmacs that is a difficult editor and not with Merlot the WISIYIG editor proposed. And almost all of them (apart from one) used the documentation to understand the grammar. The majority preferred to use as documentation a document that the author of this thesis distributed to the students with the tree structure of the grammar. As they had already given some introductory course on XML, half of them consulted also the grammar (directly the DTD) and only 2 students consulted the official documentation on-line.

6.4 Evaluation of tutor’s use of the LE

To evaluate the tutor’s use of EVA_pm, the tutor has been interviewed and the results of this open discussion, classified according to the variables that needed to be measured, are given below:

6.4.1 General satisfaction – appreciation

It is difficult for the tutor to express his opinion about the method, as he is the one that conceived the idea of using XML for scaffolding as a cognitive tool. But, in general, the complete approach with the accompanied toolset pleased him and he pointed out that it works fairly well, but it can also be improved.

6.4.2 Usefulness of the given tools

Discussion tool: very useful, but he asked for more interaction. The tutor expressed the will to be able to comment all the nodes of the grammar and not only by answering their open-questions (that is the case now) and open a discussion with the students.

Audit tool: very useful, but the interface should keep a pointer to a specific version of the project as information changes with time.

Evaluation tool: very useful and quite practical with the same comments as for the audits tool.

6.4.3 Awareness (Monitoring)

The tutor found that monitoring of both project management and project results was easily made through commNcontrol.

6.4.4 Scaffolding / Support

The tutor believes that the method scaffolds the students' work and makes them reflect on their work, that's why he supports the idea and tries to improve this method during the years.

6.4.5 Ergonomics / Usability of the interface

While the tutor appreciated the implementation that has been done for EVA_pm, and the enormous work that entailed, there are several comments that the tutor made concerning the interface of a future version and some of them are given below.

The basic tutor asked for an improvement in the annotation tools so that he could annotate each node of the project, which wasn't the case with the existing tools.

The discussion tool should show also the question that is referring to.

Names of some buttons have to be changed, as they are not really standard.

Editing (like for audits, evaluations, comments, discussion) has to be supported with more tools like possibility to delete and update.

Navigation has to be improved and a "Nielsen type menu" was suggested (STAF18 -> commNcontrol -> discussion).

In a future version there has to be found a possibility to automatically add ID's to all the nodes. Lack of this feature became an obstacle to the present implementation of extra features.

XML files should be validated against their grammar (and not only for well-formedeness as it is the case now), as many students were uploading their files without validating their syntax through the editor they were using (i.e., having some mistakes that the tutor had to fix later on).

6.5 Evaluation of the author of EVA_pm

If we have to make a conclusion after the above comments for EVA_pm, we would say that in general the approach was very good and encouraging, the grammar is quite mature and the toolset that supported the method is quite powerful.

The ergonomics of the interface though can be improved, even though it is a challenging task (web presentation poses many constraints). The scripting code is quite stable and doesn't present any serious problems. There are characteristics that are missing, but it is normal as it was already a quite ambitious project.

As XML good editing tools lack for the moment, it poses a problem to the method. Implementation also was more difficult because new technologies concerning XML were used (lack of know-how and good documentation in the area). Grammar of XML, though, for the project management had to be better explained and a more detailed documentation should be given to the students. But this problem can be solved quite easily by using in the future an XML editor that displays hints about the grammar.

Chapter

7

7 Conclusions

Finally, this last chapter recalls the initial goal of this thesis, and whether it has been reached or not and to what extent. Perspectives complete the current research giving space to the new attempts that might follow.

7.1 Concluding remarks

The initial goal of this research was to develop a Scaffolding Learning Environment that would improve PBL efficiency and help to overcome the most common pitfalls. To reach this goal, Markup Languages (XML grammars) were conceived to describe a student's work on a project and his final result. In order to support these grammars, an online environment was implemented with several tools that would hopefully facilitate the tasks. Now that this research is reaching it's end, it is interesting to go back and check whether or not this goal has been reached and to what extent.

We have presented EVA_pm, it's conceptual framework, and how it's been implemented. It is time to see if it kept its promises.

In this thesis, EVA_pm promised to help students **develop scientific inquiry** and knowledge integration skills, to focus on important and investigate key issues. This objective has greatly succeeded.

It promised also to support students in **directing investigations**. The most important is that the students were encouraged by the grammar to explicitly state out their goals and split their work into work packages that should specify how their sub-goals were going to be achieved. Finally, the possible resources needed for the project were included inside the project management file. In that way, the tutor could verify whether or not there were sufficient resources for the project and which were the important ones, by using the audit tool or the discussion tool of EVA. It cannot be claimed that it was a very sophisticated interaction as there was no possibility to annotate directly specific nodes but still it was a quite good approach.

A thing that has been promised and definitely has been respected was to make students **better manage the time** and respect the time constraints. Given the fact that the students that used EVA_pm all finished on time and respected the deadlines of all the phases, it can be said that it was successfully achieved. It can be claimed that defining in detail the work packages (by redefining objectives inside them that should correspond to the initial ones) helped a great deal to the good results. Regular feedback of the tutor empowered this.

EVA_pm promised to **overcome** possible **writer's block** or even better avoid it. And it definitely did it. The detailed grammar that prompted students to fill in the description of their work by defining the given tags, together with the phases that should be followed made it happen. It also made it easier to work in parallel on different tasks, i.e., get something written which is very encouraging for the students to see.

A very challenging objective was that EVA_pm could help students acquire **knowledge on project design** and research skills. Students by using EVA's grammar for project

management learnt at least the basic steps that should be followed when working on a project (define goal and objectives, split work into work packages and gather resources). It is uncertain though whether they could conduct a research successfully in the future, but it is of course a very encouraging approach. Probably, in order to achieve this, EVA_pm should be used several times and in different contexts by the same students to be able to deduct conclusions.

Another objective was to improve **team management** and collaboration. One of the strong points of EVA_pm's environment is its versioning system combined with its file transfer tools. It was welcomed by the students and has worked amazingly well. Especially in our case where groups of students had to share editing of files, it seems that the tool served this goal to the maximum. Another goal of the versioning system was to give the possibility to the students to go back and start from an older version in case it is needed. None of the students used this possibility. There are many assumptions that can be made but as there is no way to verify them, it is left to be explored in the future.

It has been promised that students would **reflect on their work**. By studying the students' auto-critique, it can be said in some way it has been succeeded, but not enough.

EVA_pm should **support the tutor's role** in a PBL approach. It can be said that it did, as the whole organization proposed by EVA's grammars guided also the tutor and gave him a special place and pace inside it.

A task that was greatly facilitated was **monitoring and evaluation** for the tutor. The tools that EVA provides make these tasks much easier than with any other traditional way. Of course, the progress of the work packages was left to the judgment of the student (i.e., too objective) and that might lead to misinterpretations, but still a tutor can compare with the rest of the information provided. The only thing that is missing for the moment and has been asked for from the tutor that used EVA, was the possibility to annotate specific parts of the project management. For the moment, the tutor can annotate generally the project and the paper and answer to open-questions and comments. Unfortunately, this hasn't been implemented because of several practical problems.

Another optimistic objective was to help the tutor **verify whether knowledge is being acquired**. This is always a problem with PBL strategies. Even if the student manages to bring successfully the project to the end and culminate with interesting results, there is no direct proof what and to what extent the student has learnt. Being able to watch closely during all their steps until the end of their project with EVA_pm made this a bit easier. By asking the students to write the final paper, someone can also deduct interesting information about it. It still remains a challenge, though.

EVA promised to **motivate the peers**, and definitely succeeded. The author has very often observed how students were motivated from peering and improving their work after watching the others. Progress bars were competing with each other and students were imitating the good work of others. Moreover, the "Virtual Book" also used to motivate the peers and make them reflect on their work, by watching a concrete product by the end of the project, from which a part was their work.

Finally, **distribution** of the projects and of results to **bigger audiences** is taken for granted. The portal commNcontrol is open to guests and Virtual Book joins all the results of students' projects making students take some distance from their work and see it as a part of the whole course.

Of course, we have to say here, that no matter how sophisticated a tool or a method might be for a PBL course, the role of the tutor is crucial. It is the key person that with his coaching will take the most out of the tools.

A summary of the special features of EVA's environment is the following:

- an XML grammar (DTD) for project management that was used as a cognitive tool to scaffold students during a project,
- a web portal (commNcontrol) that provided tools to support the method like awareness tools, versioning system and file transfer tools, discussion tools and tutor evaluation tools,
- a web portal (Virtual Book) that joined the results of all the projects and played the role of a constructivist element for the students by giving them the chance to see a concrete result of the whole course,
- An XML grammar (DTD) for the final paper to make students concretize their results and used also as a cognitive tool.

EVA has been welcomed and people that heard about it or the students that tested it found that it is a very interesting idea and were excited about it. Of course, later on, when students had to use the method themselves, they realized that it wasn't so easy as they imagined it should be.

Editing XML, a big problem and disadvantage of the method so far, is difficult and needs training. We believe that very soon we will have better editors with friendlier interface. But the most important and difficult thing is to get trained for a specific grammar. As for using correctly a word processor (like Ms-Word) it needs training, in the same way XML editing needs training and takes some time before getting used to it. Apart from that, training is needed also for using the interface that supports the method and the tools in it.

7.2 Outlook

Based on the above findings, EVA_pm has a lot of potential in it. On the other hand, there are several things that could be improved concerning the grammar but most of all concerning the usability of the interface. For this, important feedback from the users (students and tutor) that tested it has been written down and can be used.

It would be interesting to test this method with bigger classes and with more friendly XML editors, so that we can really evaluate the method. On the top of the wishing list for a later version of EVA_pm, are easier editing, better interface and more features for interaction between tutors and students.

Our vision for a future version of EVA_pm would be that the teachers would be able to adapt the grammar to their own specific needs (different types of projects, academic discipline, age groups, size of projects, methodological approaches, teaching philosophy, etc.).

Significant issues and challenges remain, but we believe that EVA_pm demonstrates a successful approach and we are glad that we gave it a chance.

8 Appendix A: EVA_pm's grammars

8.1 EVA_pm ML ("project.dtd")

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- ____________________________________________ _________________        -->
<!-- EVA_pm DTD for student project management & specification          -->
<!--   Copyright: (2000 & 2001) Paraskevi.Synteta@tecfa.unige.ch        -->
<!--              http://tecfa.unige.ch/~paraskev/
<!--                            Daniel K. Schneider                      -->
<!--              http://tecfa.unige.ch/tecfa-people/schneider.html     -->
<!--   Created: 23/08/2000                                              -->
<!--   Updated: 28/06/2001                                              -->
<!-- VERSIONS                                                           -->
<!-- v1.0 First stable version                                          -->
<!-- ____________________________________________________________________  -->

<!-- ___________________ ENTITY DECLARATIONS ____________________        -->
<!ENTITY % foreign-dtd SYSTEM "ibtwsh6.dtd">
<!ENTITY % stamp "
  creation-day (
00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|
25|26|27|28|29|30|31 ) '00'
  creation-month ( 00|01|02|03|04|05|06|07|08|09|10|11|12 ) '00'
  creation-year  ( 0000|2001|2002|2003|2004|2005|2006|2007|2008|2009|2010 )
'0000'
">

%foreign-dtd;

<!--  ___________________________ MAIN ELEMENT  ____________________________ -->
<!ELEMENT project  ( (hint )? , project-info ,  project-specification ,
project-problems ,  project-meetings ,  project-results ,  resources-for-
project ,  comments ,  open-questions ) >

<!--  ___________________________ GENERAL INFO  ____________________________ -->
<!ELEMENT project-info  ( (hint )? , (project-acronym )? ,  title ,
description , project-type ,  project-institution ,  project-workgroup ,
(project-coordinator )? ,  (project-supervisor )? ,  project-keywords )>
<!ATTLIST project-info  %stamp;
                        info-id          ID     #IMPLIED
                        projecturl       CDATA  #REQUIRED
                        projectlinkedto  IDREF  #IMPLIED
>
<!ELEMENT project-acronym  (#PCDATA )>
<!ELEMENT project-type  (#PCDATA )>
<!ELEMENT project-institution  (#PCDATA )>
<!-- ____________________________________________________________________ -->

<!ELEMENT project-workgroup  ( (workgroup-member )* )>
<!ELEMENT workgroup-member  ( (member-campuslogin )? ,  (member-contactinfo
) ,  (member-role )?  )>
<!ELEMENT member-campuslogin EMPTY>
<!ATTLIST member-campuslogin  campusid ID  #REQUIRED >
<!ELEMENT member-contactinfo EMPTY>
<!ATTLIST member-contactinfo  firstname     CDATA  #REQUIRED
                              familyname    CDATA  #REQUIRED
```

```
                              homepageurl   CDATA  #REQUIRED
                              workurl       CDATA  #IMPLIED
                              email         CDATA  #REQUIRED >
<!ELEMENT member-role  (#PCDATA )>
<!-- ____________________________________________________________ -->

<!ELEMENT project-coordinator  (#PCDATA )>
<!ELEMENT project-supervisor  (#PCDATA )>
<!ELEMENT project-keywords  (keyword )+>
<!ELEMENT keyword  (#PCDATA )>
<!-- ____________________________________________________________ -->
<!-- ____________________________ END OF PROJECT-INFO  _____________ -->
<!-- ____________________________________________________________ -->


<!-- ____________________ Project Specification _____________________ -->
<!ELEMENT project-specification  ( (hint )? ,  project-maingoal , project-
objectives ,  (project-approachused )? ,  (project-scenarios )? ,  project-
deliverables ,  project-work )>
<!-- ____________________________________________________________ -->

<!ELEMENT project-maingoal  ( title , description , (targetpopulation )? ,
(comments )? ,  (open-questions )? )>
<!ATTLIST project-maingoal  %stamp;
                            goal-id ID #IMPLIED >
<!ELEMENT targetpopulation  (#PCDATA )>
<!-- ____________________________________________________________ -->

<!ELEMENT project-objectives  ( (project-objective )* ,  (comments )? ,
(open-questions )? )>
<!ELEMENT project-objective  ( title , description ,  (comments )? ,
(open-questions )? )>
<!ATTLIST project-objective  %stamp;
                             obj-id ID #IMPLIED >
<!-- ____________________________________________________________ -->

<!ELEMENT project-approachused  ( title ,  (description )? ,  (comments )?,
(open-questions )? )>
<!ATTLIST project-approachused  %stamp;
                                approach-id ID #IMPLIED >
<!-- ____________________________________________________________ -->

<!ELEMENT project-scenarios  ( (project-scenario )+ ,  (comments )? ,
(open-questions )? )>
<!ELEMENT project-scenario  ( title , description ,  (comments )? ,  (open-
questions )? )>
<!ATTLIST project-scenario  %stamp;
                            scenario-id ID #IMPLIED  >
<!-- ____________________________________________________________ -->

<!ELEMENT project-deliverables  ( (project-deliverable )* ,  (comments )? ,
(open-questions )? )>
<!ELEMENT project-deliverable  ( title , description , delivery-date ,
deliverable-nature ,  (comments )? ,  (open-questions )? )>
<!ATTLIST project-deliverable  %stamp;
                               deliv-id ID #IMPLIED  >
<!ELEMENT delivery-date EMPTY>
<!ATTLIST delivery-date  day   (
00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|
25|26|27|28|29|30|31 ) '00'
```

```
                        month ( 00|01|02|03|04|05|06|07|08|09|10|11|12 )
'00'
                        year  (
2001|2002|2003|2004|2005|2006|2007|2008|2009|2010 ) '2001'
>
<!ELEMENT deliverable-nature  (#PCDATA )>
<!-- ____________________________________________________________ -->

<!ELEMENT project-work  (workpackages ,  (comments )? ,  (open-questions )?
)>
<!ELEMENT workpackages  (workpackage* )>
<!ELEMENT workpackage  ( title , description , wp-objectives , wp-planning
, wp-expectedresults ,  (comments )? ,  (open-questions )? )>
<!ATTLIST workpackage  %stamp;
                       wp-id                 ID      #IMPLIED
                       depends-on-wp         IDREFS  #IMPLIED
                       referstodeliverableid IDREFS  #REQUIRED >
<!-- ____________________________________________________________ -->

<!ELEMENT wp-objectives  ( (wp-objective )+ )>
<!ELEMENT wp-objective  ( title , (description )? )>
<!ATTLIST wp-objective  %stamp;
                        wpobj-id      ID     #IMPLIED
                        referstoobjid IDREF  #REQUIRED >
<!-- ____________________________________________________________ -->

<!ELEMENT wp-planning EMPTY>
<!ATTLIST wp-planning  start-day   (
00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|
25|26|27|28|29|30|31 ) '00'
                       start-month ( 00|01|02|03|04|05|06|07|08|09|10|11|12
) '00'
                       start-year  ( 0000|2001|2002|2003|2004|2005 )
'0000'
                       end-day     (
00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|
25|26|27|28|29|30|31 ) '00'
                       end-month   ( 00|01|02|03|04|05|06|07|08|09|10|11|12
) '00'
                       end-year    ( 0000|2001|2002|2003|2004|2005 )
'0000'
                       man-days    NMTOKEN  #REQUIRED
                       completion  NMTOKEN  #REQUIRED >
<!-- ____________________________________________________________ -->

<!ELEMENT wp-expectedresults  ( (wp-expectedresult )+ )>
<!ELEMENT wp-expectedresult  ( title ,  (description )? )>
<!ATTLIST wp-expectedresult  %stamp;
                             expres-id ID #IMPLIED  >
<!-- ____________________________________________________________ -->
<!-- _____________________ END OF PROJECT SPECIFICATION ___________ -->
<!-- ____________________________________________________________ -->


<!-- ___________________________ PROBLEMS _____________________________ -->
<!ELEMENT project-problems  ( (hint )? , (problem )* )>
<!ELEMENT problem  ( title ,  (description )? ,  (solution-to-problem )? ,
(comments )? ,  (open-questions )? )>
<!ATTLIST problem  %stamp;
                   problem-id ID #IMPLIED
                   problem-status  (open | closed )  'open' >
```

```
<!ELEMENT solution-to-problem  (#PCDATA )>
<!-- _____________________________________________________________ -->
<!-- _____________________________ END OF PROBLEMS _____________________________ -->
<!-- _____________________________________________________________ -->


<!-- _____________________________ MEETINGS _____________________________ -->
<!ELEMENT project-meetings  ( (hint )? , (meeting )* )>
<!ELEMENT meeting  ( title ,  (description )? ,  (comments )? ,  (open-
questions )? )>
<!ATTLIST meeting  %stamp;
                   meeting-id   ID     #IMPLIED
                   participants CDATA  #IMPLIED >
<!-- _____________________________________________________________ -->
<!-- _____________________________ END OF MEETINGS _____________________________ -->
<!-- _____________________________________________________________ -->


<!-- _____________________________ RESULTS _____________________________ -->
<!ELEMENT project-results  ( (hint )? , (project-result )*  ,  (paper-for-
results )? )>
<!ELEMENT project-result  ( title ,  (description )? , (result-url )? ,
(comments )? , (open-questions )? )>
<!ATTLIST project-result  %stamp;
                          result-id ID #IMPLIED  >
<!ELEMENT result-url  (#PCDATA )>
<!ELEMENT paper-for-results  EMPTY>
<!ATTLIST paper-for-results  url  CDATA  #REQUIRED  >
<!-- _____________________________________________________________ -->
<!-- _____________________________ END OF RESULTS _____________________________ -->
<!-- _____________________________________________________________ -->


<!-- _____________________________ RESOURCES _____________________________ -->
<!ELEMENT resources-for-project  ( (hint )? ,  (resource )* , citations ,
(comments )? ,  (open-questions )? )>
<!ELEMENT resource  ( (person-as-resource | book-as-resource | url-as-
resource | paper-as-resource | other-resource )  )>
<!ATTLIST resource  %stamp;
                    res-id  ID              #REQUIRED
                    cite-id IDREF           #IMPLIED
                    language   (en|fr|other) #REQUIRED  >
<!ELEMENT person-as-resource EMPTY>
<!ATTLIST person-as-resource  name     CDATA  #REQUIRED
                              home-url CDATA  #IMPLIED
                              email    CDATA  #REQUIRED >
<!ELEMENT book-as-resource EMPTY>
<!ATTLIST book-as-resource  title        CDATA  #REQUIRED
                            authors      CDATA  #REQUIRED
                            publish-date CDATA  #IMPLIED
                            publisher    CDATA  #IMPLIED
                            url          CDATA  #IMPLIED
                            isbn         CDATA  #REQUIRED >
<!ELEMENT paper-as-resource EMPTY>
<!ATTLIST paper-as-resource  title        CDATA  #REQUIRED
                             authors      CDATA  #REQUIRED
                             publish-date CDATA  #IMPLIED
                             journal      CDATA  #IMPLIED
                             url          CDATA  #IMPLIED >
<!ELEMENT url-as-resource EMPTY>
<!ATTLIST url-as-resource  address CDATA  #REQUIRED
```

```
                          title   CDATA  #IMPLIED >
<!ELEMENT other-resource EMPTY>
<!ATTLIST other-resource  description CDATA  #REQUIRED >
<!ELEMENT citations  ( (citation )* )>
<!ELEMENT citation %vert.model;>
<!ATTLIST citation %stamp;
                  for-resource IDREF #REQUIRED
                  cite-id      ID    #IMPLIED  >
<!-- ____________________________________________________________ -->
<!-- _____________________ END OF RESOURCES _____________________ -->
<!-- ____________________________________________________________ -->

<!-- ____________________________________________________________ -->
<!-- _____________________ END OF PROJECT _____________________ -->
<!-- ____________________________________________________________ -->

<
!-- ___________________ COMMON ELEMENTS ____________________ -->
<!ELEMENT comments %vert.model;>
<!ATTLIST comments  %stamp;
                   comm-id ID #IMPLIED  >
<!-- ____________________________________________________________ -->

<!ELEMENT open-questions  (question )*>
<!ELEMENT question  (title ,  (description )? )>
<!ATTLIST question  %stamp;
                   quest-id ID #IMPLIED  >
<!-- ____________________________________________________________ -->

<!ELEMENT title  (#PCDATA )>
<!-- ____________________________________________________________ -->

<!ELEMENT description %vert.model;>
<!-- ____________________________________________________________ -->

<!ELEMENT hint  %vert.model;>
```

8.2 EVA_paper ML (“paper.dtd”)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- ____________________________________________________________________  -->
<!-- EVA_paper DTD for the paper that will describe the final product(s)-->
<!--          of the student's project                                     -->
<!--   It includes the DTD "ibtwsh6.dtd" by John Cowan (cowan@ccil.org)    -->
<!--   Copyright: (2000 & 2001) Paraskevi.Synteta@tecfa.unige.ch           -->
<!--              http://tecfa.unige.ch/~paraskev/                         -->
<!--                            Daniel K. Schneider                        -->
<!--              http://tecfa.unige.ch/tecfa-people/schneider.html        -->
<!--   Created: 23/08/2000                                                 -->
<!--   Updated: 28/06/2001                                                 -->
<!-- VERSIONS                                                              -->
<!-- v1.0 First stable version                                             -->
<!-- ____________________________________________________________________  -->

<!ENTITY % foreign-dtd SYSTEM "ibtwsh6.dtd">

<!-- ______________________________ content ___________________________________ -->

%foreign-dtd;

<!ELEMENT paper ( abstract, preface, introduction, main, conclusion,
(annexe)?) >
<!ATTLIST paper
  creation-day
(00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24
|25|26|27|28|29|30|31) '00'
  creation-month (00|01|02|03|04|05|06|07|08|09|10|11|12) '00'
  creation-year  (2001|2001|2002|2003|2004|2005|2006|2007|2008|2009|2010)
'2001'
>

<!ELEMENT abstract %vert.model;>
<!ATTLIST abstract %all;>

<!ELEMENT preface %vert.model;>
<!ATTLIST preface %all;>

<!ELEMENT introduction %vert.model;>
<!ATTLIST introduction %all;>

<!ELEMENT main %vert.model;>
<!ATTLIST main %all;>

<!ELEMENT conclusion %vert.model;>
<!ATTLIST conclusion %all;>

<!ELEMENT annexe %vert.model;>
<!ATTLIST annexe %all;>
```

8.3 Example file (xsl view) for project management DTD

An example can be found in the following address: http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/projects/proj11/project.sxml

8.4 Example file (xsl view) for paper DTD

An example of the grammar can be found here:

http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/projects/proj11/text.sxml

and all the papers written through the portal “Virtual Book”:

http://tecfa.unige.ch/staf/staf-e/paraskev/servlet/journal.

8.5 EVA’s URL’s

Access: http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/commNcontrol.html

Virtual Book: http://tecfa.unige.ch/staf/staf-e/paraskev/servlet/journal

Documentation: http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/pm-tech.html

Java servlets: http://tecfa.unige.ch/staf/staf-e/paraskev/memoire/important/

STAF-18 course web site:

http://tecfa.unige.ch/tecfa/teaching/staf18/staf18-overview.html

and inside it the phases followed:

http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/gestion.html

9 Appendix B: Questionnaire and Statistics

9.1 Online Questionnaire for EVA_pm (to STAF-Gina students)

Access online from:

http://tecfa.unige.ch/staf/staf-e/paraskev/memoire/important/evaluation/welcome.html

A copy is given in the following pages (numbering is missing) and a table with the descriptive statistics of the answers follows.

9.1.1 Statistics

Descriptive

Questions	N	Minimum	Maximum	Mean	Std. Deviation
A1	12	23.0	49.0	29.08	9.395
A4.1	12	1	4	2.42	.996
A4.2	12	2	4	3.67	.651
A4.3	12	2	5	3.75	.965
A4.3	12	1	4	2.42	.793
B1	12	1	5	4.00	1.128
C1	12	2	5	3.83	1.115
D1	12	3	5	3.92	.793
E1	12	3	6	4.25	.965
E2	12	3	6	4.17	.718
E4	8	3	6	4.50	.926
E5	8	4	6	5.00	.756
F1	12	2	6	5.00	1.206
F2	12	2	5	4.42	.900
G1	12	1	5	3.08	1.443
G2	12	1	6	3.25	1.545
G3	12	1	5	4.08	1.165
H1	12	1	6	4.25	1.288
H2	12	1	6	4.00	1.348
I1	12	3	6	4.33	.985
I2	12	1	6	4.00	1.651
J1	12	2	6	4.67	1.435
Valid N	8				

10 Appendix C: EVA_pm manual

Here can be found all the information about EVA_pm method, in what framework can be applied, to which public, technical requirements, etc.

10.1 User Guide for EVA

10.1.1 Framework

EVA can be used as a supporting tool to a Project-Based Learning approach and can be used for projects on almost any disciplinary. It can be combined with other tools or software that is needed to perform the project. It is very important that in the beginning the tutors provide a list with projects from which the students can choose one or at least can be inspired from.

It is suggested to split the project into phases where the students are asked little by little to fill in information in the xml file and tutors give them audits in order to continue (see §6.1.4).

10.1.2 Target public

Eva_pm can be used mostly for high school or university students for the simple reason that editing the xml files is quite demanding as the grammars are asking for a lot of details. But we could imagine a much more simplified grammar based in the same principles of EVA_pm that would be easily applied to late primary school students.

10.1.3 Collaboration

It is strongly advised to work in pairs, thing that makes the experiences gained richer. Global and group mailing lists are also very important to be used for students to communicate between them. Like that, students can help each other and in that way, they will also understand better. These mailing lists can be used even for reasons out of the context of a project (e.g. for the organization of a party). That helps the community building and strengthens the team spirit.

10.1.4 Coaching / Support

It is extremely important to coach PBL approaches, so the tutors that will use EVA_pm should be available to coach regularly the project (probably, that's why it is much better to be more than one). Regular feedback (using always EVA's tools) in the form of audits or evaluations help students regulate their work, encourages them and motivates them by peering the other students' evaluations.

10.1.5 Technical requirements

The technical requirements to install EVA's environment are:

- is a simple web server (e.g. Apache web server),
- a Java server (e.g. Tomcat server),

- Xalan, the XSLT processor or Cocoon XML Publication Framework (it's XSLT processor, Xalan and configuration of web server to understand the extension (.sxml)

- Classes needed: xerces.jar (especially DOMParser), Cocoon servlet.jar, and cos.jar (com.oreilly.servlet.MultipartRequest)

To edit the XML files an XML editor is needed (see §5.1). Our favorites are XEmacs in *psgml mode*, XMetal and epcEdit.

For the installation, there are three things that need to be defined. One is the path where the projects are going to be stored. The second is the path where EVA's components are going to be stored. And the third is, the path from where the access to EVA will be made. In our case these are:

http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/projects/

http://tecfa.unige.ch/staf/staf-e/paraskev/memoire/important/ and inside four subdirectories: **servlets** (for code java), **dtd** (for pm dtd's, stylesheets, initial xml files, etc.), **paper** (for paper's dtd's, stylesheets, initial files, etc.) and **images** (for the images used inside the servlets).

http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/commNcontrol.html

For more, contact Paraskevi (Vivian) Synteta (Paraskevi.Synteta@tecfa.unige.ch).

10.1.6 Documentation for the Grammar

Documentation for both grammars (project.dtd and paper.dtd) can be found here:

http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/projectDTD.html (for project),

http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/dtdparseproject/ (for project),

http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/dtdparsepaper/ (for paper).

11 References

[1] Barron B. J. S., Schwartz D. L., Vye N. J., Moore A., Petrosino A., Zech L., Bransford J. D. and The Cognition and Technology Group at Vanderbilt. (1998). *Doing with the understanding: Lessons from research on problem and project-based learning.* The Journal of the Learning Sciences. 7. 271-311.

[2] Blumenfeld P. C., Soloway E., Marx R. W., Krajcik J. S., Guzdial M., and Palincsar A. (1991). *Motivating project-based learning: Sustaining the doing, supporting the learning.* Educational Psychologist. 26. 369-398.

[3] Brown Edward (brown@cs.mun.ca) and Byrne Rodrigue (rod@cs.mun.ca). (2000). Department of Computer Science Memorial University of Newfoundland Canada. *Lightweight Tools for On-line Course Development*.

[4] Brown J. S., Collins A., and Duguid P. (1989). *Situated cognition and the culture of learning.* Educational Researcher. 18(1). 32-42.

[5] Cognitive and Technology Group, Learning Technology Center, Peabody College of Vanderbilt University. (1991). *Technology and the design of generating learning environments.* Educational Technology. 31(5). 34-40.

[6] Collins A., and Brown J. S. (1988). *The computer as a tool for learning through reflection.* Learning Issues for Intelligent Tutoring Systems. H. Mandl and A. Lesgold (Eds) New York: Springer. 1-18.

[7] Collins A., Brown J. S., and Newman S. (1991). *Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics.* In L. B. Resnick (Ed.) Motivation, learning and instruction: Essays in honor of Robert Glaser. 453-494. Hillsdale: Lawrence Erlbaum Associates.

[8] Collis Betty. (1998). Faculty of Educational Science and Technology, University of Twente, The Netherlands. *New didactics for university instruction: why and how?*. Computers & Education. 31. 373-393.

[9] *Contextual Academics Support: Project-Based Learning*. Retrieved October 16, 2000 from the web: http://www.cord.org/lev2.cfm/65 (Title: Project-Based Learning).

[10] *CSILE. Resource List: In print and Online*. Retrieved February 5, 2001 from the web: http://csile.oise.utoronto.ca/csile_biblio.html (Title: Resource List).

[11] *CSILE: An Overview*. Retrieved February 5, 2001 from the web: http://csile.oise.utoronto.ca/intro.html (Title: Introduction to CSILE).

[12] Cuthbert Alex. (April 28, 2000). *WISE Perspective: Scaffolds For Knowledge Integration*. Design Issues for Scaffolded Learning Environments (SLEs), AREA 2000 Symposium. Retrieved October 24, 2000 from the web: http://www.kie.berkeley.edu/people/alex/aera2000/cuthbert_position.html .

[13] Derry S., Tookey K., and Chiffy A. (1994). *A microanalysis of pair problem solving with and without a computer tool.* Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans, LA.

[14] Ferrandez Teresa. (1998). B.S.C.E., University of Dalarna. *Development and Testing of a standardized format for distributed learning assessment and evaluation using XML*.

[15] Fragnière Jean-Pierre. (1996). *Comment réussir un mémoire.* DUNOD, Paris. ISBN 2-100-02775-1.

[16] Greening T. (1998). *Scaffolding for success in PBL.* Med. Educ Online. Available from URL http://www.Med-Ed-Online.org.

[17] Grønbæk, K. (1991*). Prototyping and active user involvement in system development: Towards a cooperative prototyping approach*. Ph.D. dissertation, Computer Science Department, Aarhus University, Denmark.

[18] Guzdial M. (1998). *Technological Support for Project-Based Learning.* In C. Dede (Ed.) Learning with Technology, Alexandria, VA:ASCD. (pp. 47-72).

[19] Guzdial M., Hmelo C., Hubschel R., Nagel K., Newstetter W., Puntambekar S., Shabo A., Turns J., and Kolodner J. L. *Integrating and Guiding Collaboration: Lessons Learned in Computer-Supported Collaborative Learning Research at Georgia Tech.* EduTech Institute, College of Computing Georgia Institute, Learning Research and Development Center University of Pittsburgh, EduTech Institute Center for Human-MachineSystems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology.

[20] Hmelo C. E, Guzdial M. and Turns J. (1998). *Computer Support for collaborative learning: Learning to support student engagement.* Journal of Interactive Learning research. 9(2). 107-129.

[21] Hoadley M. Christopher. University of California at Berkeley, EDUTECH. *Design activities for learning science: experiences and strategies.* Retrieved October 24, 2000 from the web: http://www.cc.gatech.edu/edutech/dew/papers/hoadley.html .

[22] Isenhour L. Philip, Caroll M. John, Neale C. Dennis, Rosson Mary Beth, and Dunlap R. Daniel. (2000). Center for Human-Computer Interaction, Department of Computer Science, Virginia Tech. *The Virtual School: An integrated collaborative environment for the classroom*. Educational Technology & Society ET&S. 3(3). ISSN 1436-4522.

[23] Jackson M. (April 25, 1997). IMPEL2 Project, University of Northumbria at Newcastle, UK. *IMPEL2 – Survey on Resource Based Learning.* DeLiberations. Retrieved October 17, 2000 from the web: http://www.lgu.ac.uk/deliberations/rbl/impel.html .

[24] Jacobson J. Michael and Levin A. James. (1995). *Conceptual Frameworks for Network Learning Environments: Constructing Personal and Shared Knowledge Spaces*.

[25] Jonassen D. H., and Wang S. (1993). *Acquiring structural knowledge from semantically structured hypertext.* The Journal of Computer-based Instruction. 20(1). 1-8.

[26] Jonassen David (University of Colorado), Terry Mayes, and Ray McAleese (Heriot-Watt University). *A Manifesto for a Constructivist Approach to Technology in Higher Education.*In T. Duffy, D. Jonassen, & J. Lowyck (Eds), Designing contsructivist learning environments. Heidelberg, FRG: Sprinter-Verlag. Retrieved October 19, 2000 from the web: http://led.gcal.ac.uk/clti/papers/TMPaper11.html .

[27] Jones B. F., Rasmussen C. M., and Moffitt M. C. (1997). *Real-life problem solving: A collaborative approach to interdisciplinary learning.* Washington, DC: American Psychological Association.

[28] Kearsley Greg & Schneiderman Ben. (Version 4/5/1999). *Engagement Theory: A framework for technology-based teachning and learning*. Retrieved June 6, 2001 from the web: http://home.sprynet.com/~gkearsley/engage.htm (Title: Engagement Theory).

[29] Kehoe Colleen, Guzdial Mark, and Turns Jennifer. *What we know about technological support for project-based learning*.

[30] Klein D. C. D., O'Neil H. F., Dennis R. A., and Baker E. L.. (1997). *The five families of cognitive learning: A context in which to conduct cognitive demands analysis of technologies.* Paper presented at the Annual Meeting of the American Educational Research Association, Chicago.

[31] Laffey J. (1998). *A computer-mediated support system for project-based learning.* ETR&D. 46(1). 73-. Available online: http://140.122.109.65/Minutes/001109flyfox.html .

[32] Laffey James. May 1996. *Teachers To Attend Summer Workshop, Will Learn Stella: Improving technology skills to improve student projects.* Project MOST News. 1(2).

[33] Mayes Terry. Institute for Computer-Based Learning, Heriot-Watt University, Edinburgh. *MINDTOOLS: A suitable case for learning.* Retrieved October 19, 2000 from the web: http://led.gcal.ac.uk/clti/papers/TMPaper5.html .

[34] *MSDN Online - Web Workshop - XML TOC TOC*. Retrieved June 15, 2001 from the web: http://msdn.microsoft.com/workshop/xml/toc.htm .

[35] Nolan Jason (jason.nolan@utoronto.ca). (last updated: March 28, 2000). Knowledge Media Design Institute, University of Toronto. VASE: Virtual Assignment Server Environment. Website: http://achieve.forestry.utoronto.ca/papers/tcc2000/ .

[36] *PBLSS (Project Based Learning Support System)*. Available online: http://babylon.coe.missouri.edu/cc/aboput.html#features .

[37] *PIViT: Project Integration and Visualization Tool*. Retrieved February 5, 2001 from the web: http://www.umich.edu/~pbsgroup/PIViT.html .

[38] Project MOST: Supporting Exciting Opportunities through Project-Based Learning and Technology. (February 1996). Project MOST News. 1(1).

[39] *Project-Based Science (PBS)*. Retrieved February 5, 2001 from the web: http://www.umich.edu/~pbsgroup/ .

[40] Puntambekar S. and du Boulay B. (1997). *Design and Development of MIST: A system to help students develop metacognition*. Journal of Educational Computing Research. 16(1). 1-35.

[41] Puntambekar S., and Kolodner J. L. *Distributed scaffolding: Helping students learn in a learning by design environment.* In A.S. Bruckman, M. Guzdial, J. L. Kolodner, and A. Ram (Eds), ICLS 1998, Proceedings of the International Conference of the Learning Sciences. 35-41.

[42] Puntambekar Sadhana and Kolodner L. Janet. (1998). *The Design diary: A Tool to Support Students in Learning Science by Design*. Retrieved November 22, 2000 from the web http://www.cc.gatech.edu/edutech/projects/dd_icls98/icls_design_diaries.html

[43] Puntambekar Sadhana. (1999). *An integrated approach to individual and collaborative learning in a web-based learning environment*. University of Connecticut. Retrieved October 19, 2000 from the web: http://kn.cilt.org/cscl99/A57/A57.HTM .

[44] Quivy R. & Van Campenhoudt L. (1995). *Manuel de recherche en sciences sociales.* DUNOD, Paris.

[45] Reginald Gregoire inc. and Laferriere Therese. (March 31, 1999). Canada's Schoolnet. *Project-Based Collaborative learning with network computers*. Retrieved November 6, 2000 from the web: http://www.tact.fse.ulaval.ca/ang/html/projectg.html (Title: Learning together in the networked classroom).

[46] Riley Kevin. *Gestalt – On-line Educational and Training – State of the Art*. [Retrieved from the web]

[47] Sage S. M. (1996). *A qualitative examination of problem-based learning at the K-8 level: Preliminary findings.* Paper presented at the Annual Meeting of the American Educational Research Association, New york.(ED 398 263).

[48] Salomon Gavriel. *Distributed Cognitions: Psychological and educational considerations*. Learning in doing: Social, Cognitive, and Computational Perspectives. CAMBRIDGE University Press. ISBN 0-521-41406-7.

[49] Scardamalia M., and Bereiter C. (1991). *Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media.* Journal of the Learning Sciences. 1. 37-48.

[50] Scardamalia M., Bereiter C. (1994). *Computer support for knowledge building communities.* Journal of the Learning Sciences. 3(3). 265-283.

[51] Scardamalia M., Bereiter C., and Lamon M. (1994). *The CSILE Project: Trying to bring the classroom into World 3.* Classroom Lessons: Integrating Cognitive Theory & Classroom Practice. K. McGilly, Ed. Cambridge, Mass.: MIT Press. 201-228.

[52] Scardamalia M., Bereiter C., McLearn R.S., Swallow J. and Woodruff E. (1989). *Computer supported intentional learning environments.* Journal of Educational Research. 5. 51-68.

[53] Schneider K. Daniel. (2000). *Balises de methodologie pour la recherche en sciences sociales*. (Matériaux de cours)

[54] Schneider K. Daniel. (2001). *Le rôle de Internet dans la formation supérieure: scénarii et technologies,* Introduction et essai prepare pour le colloque International "Enseignement des Langues et Multimedia (28-29 a Alger)" et la journee NTICE@INSTAT (27 avril 2001).

[55] Schneiderman B. (1988). *Relate-Create-Donate: an educational philosophy for the cyber-generation.* Computers and Education.

[56] Schneiderman Ben, Borkowski Ellen Yu, Alavi Maryam, and Norman Kent. (1998). *Emergent Patterns of Teaching /Learning in Electronic Classrooms*. ETR&D. 46(4). 23-42. ISSN 1042-1629.

[57] Schneiderman Ben. (1998). *Codex, memex, genex: The pursuit of transformational technologies*. International Journal of Human-Computer Interaction. 10(2). 87-106. Available online: ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/3862HTML/3862.html .

[58] Singer Jon. (May 1996). *MOST Student-Teachers reflect on Facilitating Project-Based Science.* Project MOST News. 1(2).

[59] Stites Regie. (January, 1998). SRI International. *The Challenge 2000 Multimedia Project: Project-Based Learning with Multimedia*. Retrieved October 16, 2000 from the web: http://pblmm.k12.ca.us/PBLGuide/pblresch.html (Title: Evaluation of Project-Based Learning. What does research say about outcomes from project-based learning?).

[60] *The Challenge 2000 Multimedia Project: Project-Based Learning with Multimedia*. Retrieved January 29, 2001 from the web: http://pblmm.k12.ca.us/PBLGuide/WhyPBL.html (Title: Why do project-based learning?).

[61] *The Challenge 2000 Multimedia Project: Project-Based Learning with Multimedia*. Retrieved January 29, 2001 from the web: http://pblmm.k12.ca.us/PBLGuide/PBL&PBL.html (Title: Project-Based and Problem-Based: The same or different?).

[62] *The Guide on the Side: Project-Based Learning Resources – compiled by William Belsey*. Retrieved October 16, 2000 from the web: http://www.iearn_canada.org/guideontheside.html .

[63] Thomas J. W., Mergendoller J. R., and Michaelson A. (1999). *Project-based learning: A handbook for middle and high school teachers.* Novato, CA: The Buck Institute for Education.

[64] Thomas W. John. (March 2000). *A review of research on project-based learning*. Retrieved from the web: http://www.autodesk.com/foundation .

[65] Torp L., and Sage S. (1998). *Problems as possibilities: Problem-Based learning for K-12 education.* Association for Supervision and curriculum development.

[66] *What is Resource-Based Learning?*. Retrieved September 23, 2000 from the web: http://stauffer.queensu.ca/inforef/tutorials/rbl/rblintro.html (Title: Resource-Based Learning).

[67] *XML Editors: Allegations of Functionality in search of reality.* Retrieved June 15, 2001 from the web: http://www.ivritype.com/xml/ (Title: XML: In search of a good editor) .

[68] *XML Tutorial – XML Usage*. W3Schools. Retrieved January 31, 2001 from the web: http://www.w3schools.com/xml/xml_usedfor.asp .

[69] *XML.com: What is XML?*. Retrieved March 13, 2001 from the web: http://www.xml.com/pub/a/98/10/guide1.html

12 Table of Figures

Figure 1. *This image shows how the XML file that a student writes can be transformed through EVA.* 11

Figure 2. *Genex framework: Foundational beliefs are tied to creative phases and sets of existing tools that can be refined and integrated into a genex (Schneidermann B., 1998).* 20

Figure 3. *The prototyping model by Naumann & Jenkins (1982)* 25

Figure 4. *Typical process according to traditional Information System prototyping approaches. Few activities where users participate (taken from the Ph.D thesis of Grønbæk, 1991).* 26

Figure 5. *A typical prototyping process as suggested by the cooperative prototyping approach. More activities where users participate, and more interplay between activities (taken from the Ph.D thesis of Grønbæk, 1991).* 26

Figure 6. *EVA's evolution of versions for commNcontrol (project's portal) for staf-eva, staf-fanny and staf-gina (STAF-18 course, http://tecfa2.unige.ch/staf/staf-e/staf18/proj-liste1.php, http://tecfa2.unige.ch/servlets/staf18FList2, http://tecfa.unige.ch/tecfa/teaching/staf18/files-g/commNcontrol.html?action=commNcontrol).* 27

Figure 7. *EVA's evolution of versions for the final papers' portal for staf-fanny, staf-gina (STAF-18 course, http://tecfa2.unige.ch/servlets/staf18FList3, http://tecfa.unige.ch/staf/staf-e/paraskev/servlet/journal).* 27

Figure 8. *EVA's authentication form* 46

Figure 9. *EVA's commNcontrol console for a tutor* 47

Figure 10. *EVA's commNcontrol for a group of students* 48

Figure 11. *Mailing possibilities of EVA* 49

Figure 12. *An open question posed inside EVA* 49

Figure 13. *EVA's discussion form* 50

Figure 14. *EVA's discussion* 51

Figure 15. *EVA's file download area* 51

Figure 16. *EVA's file upload form* 52

Figure 17. *EVA's versioning system* 53

Figure 18. *EVA's audit form* 54

Figure 19. *EVA's evaluation form* 54

Figure 20. *EVA's evaluation and audit results* 55

Figure 21. *EVA's Virtual Book* 56

Figure 22. *Icons for tutor's audits* 58

Figure 24. *Diagram of EVA's Authentication structure* 59

Figure 25. *Diagram of EVA's commNcontrol and Virtual Book structure* 62

13 Glossary

Cascading stylesheets (CSS)
Formatting descriptions that provide augmented control over presentation and layout of HTML and XML elements. CSS can be used for describing the formatting behavior of simply structured XML documents, but does not provide a display structure that deviates from the structure of the source data.

Constructivism
Constructivism is a theory of learning based on the idea that knowledge is constructed by the knower based on mental activity. Learners are considered to be active organisms seeking meaning. Constructions of meaning may initially bear little relationship to reality (as in the naive theories of children), but will become increasing more complex, differentiated and realistic as time goes on.

Cognition
The mental activity by which an individual is aware of and knows about his or her environment, including such processes as perceiving, remembering, reasoning, judging, and problem solving.

DTD
A set of declarations, written in a formal notation defined in the XML and SGML standards, that define the structure of a set of documents. Among other things, a DTD declares all of the element names that can appear in a document, the hierarchy in which they can be arranged, the type of content they can have, and which attributes they can have.

DOM
(Document Object Model) is a platform - and language - neutral standard **API** (Applications Programming Interface) for manipulating XML (and HTML) **document trees**.

SAX
Simple API for XML **event-based** parsing (parsing events reported to application through callback).

Scaffolding
Traditionally, scaffolding occured through personal interaction between students and instructors. With the popularity of electronic telelearning environments however, scaffolding should be integrated in the course environment. Scaffolding is studied by many researchers. Especially in the model of cognitive apprenticeship, developed by Collins-Brown-Newman (1989). In this model scaffolding is mentioned as a means of coaching students until students can perform intellectual tasks on their own. In a study by Jonassen, scaffolding is used in the field of diagnostic reasoning. A causal model is constructed of the diagnostic reasoning process that is used by experts. Students are scaffolded by comparing their reasoning process to that of experts.

SLE
Scaffolding Learning Environment

Text Markup
Inserting tags into the middle of an element's text flow, to mark certain parts of the element with additional meta-information.

Valid XML document
Document that is well-formed and conforms to the vocabulary specified in a DTD or schema.

Well-formed XML document
Document that meets the requirements listed in the W3C Recommendation for XML 1.0: It contains one or more elements; it has a single document element, with any other elements properly nested under it; each of the parsed entities referenced directly or indirectly within the document is well-formed. A well-formed XML document does not necessarily include a DTD.

XHTML
XHTML is a family of current and future document types and modules that reproduce, subset, and extend HTML 4 [HTML]. XHTML family document types are XML based, and ultimately are designed to work in conjunction with XML-based user agents.

XML
eXtensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them. XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language [ISO 8879].

XML Parser
A generalized XML parser reads XML files and generates a hierarchically structured tree, then hands off data to viewers and other applications for processing. A validating XML parser also checks the XML syntax and reports errors.

XSchema
Defines structure, content and semantics of XML documents. More sophisticated than DTD in how they describe, create, manage and validate the info. XSchema is functionally equivalent to a DTD, but is written in XML; a schema also provides for extended functionality such as data typing, inheritance, and presentation rules.

XSL
XSL is a language for expressing stylesheets. It consists of two parts: a language for transforming XML documents (**XSLT**), and an XML vocabulary for specifying formatting semantics (**XSLFO**). An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary.

14 Index

C

CaMILE,**3, 16**
coaching,3, 4, 6, 18, 45, 65, 66
commNcontrol,11, 28, 44, 45, 46, 47, 50, 52, 53, 54, 64, 73
constructivism,vii, 4
CSILE,**2, 89**

D

DTD,vii, 21, 28, 29, 31, 34, 35, 36, 37, 38, 40, 42, 43, 79, 84, 95

E

EVA,**21, 22, 37, 38, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 63, 64, 65, 66, 70, 73, 78, 79, 87**

G

Genex,19, 20, 46
grammar,**6, 7, 12, 20, 21, 24, 28, 31, 32, 35, 37, 38, 40, 41, 42, 43, 44, 79, 84**

I

IMS,32, 33
inquiry model,15

P

PBL,vii, 1, 2, 4, 5, 7, 8, 13, 14, 15, 16, 17, 18, 19, 21, 32, 34, 38, 65, 92
PBLSS,**4, 91**
PIViT,4, 91
procedural-facilitation,2
production model,15

R

reflection,vii, 4, 5, 6, 15, 18, 21, 22, 32, 37, 38
Reflective Learner,3, 5
Resource-Based Learning,19, 92

S

scaffolding,**vii, 2, 4, 5, 6, 15, 18, 19, 21, 22, 31**
staf18,7, 23, 63, 64, 65, 66

T

Team Facilitator,3
thinking tools,16

W

WBILE,4

X

XMetal,**44, 66**
XML,vii, 5, 7, 8, 11, 21, 29, 30, 31, 32, 34, 36, 40, 41, 42, 43, 44, 45, 46, 66, 89, 91, 92, 93, 95