Tempus Tacis Project UM_CP-20560-1999

‘Development of New Framework for the M.Sc. Thesis Work’

Additional Outcome: Web-based consulting system for credit transfer and course recognition

Borys Omelayenko

Vrije Universiteit Amsterdam

Email: borys@cs.vu.nl

URL: www.cs.vu.nl/~borys

Date: 19 August 2000

Introduction

The second outcome of the Project is targeted to ‘Developing the set of Manuals for accounting credits for courses and projects’, that must contain the new model of diplomas grade list, instruction for assigning credits for internal courses, external courses and distance courses.

During the project work we identified that the manuals for paperwork are not sufficient to improve the situation with credit transfer and credit accounting in KhTURE. The main obstacles to this are:

- administrative staff in the Dean’s office has low knowledge of English;
- the staff does not understand European credit transfer system very well because of big differences to Ukrainian system and the above mentioned luck of English;
- different European universities exploit different credit accounting schemes; after the documents of a student are translated into ECTS an additional effort of the administrative staff is still needed to match the courses;

We found that students in Ukraine want to participate in EU exchange programs but they do not know where to find information about credit transfer. The main problem here is that the students need some automated and ultimately available source of credit transfer information to evaluate how their courses taken in Ukraine will be recognized by some EU university.

European students also need such a resource to quickly find the list of courses that they can take in some certain university, and to evaluate their ability to get a degree in it.

Finally, international offices of the universities in Europe need such a tool to ..

Finally, ordinary students need this information source to plan their studies and courses in the universities

The previous project idea was to develop guidelines for translating of Ukrainian academic documents into some European documentation system to allow fast and easy credit transfer and course recognition. The retrainings held during the project cleared out that this is insufficient, and we need to build some document transfer system to facilitate this. This shift is illustrated in Figure 1.

This forced us to start developing of a web-based tool for credit transfer. The tool must be open for a wide community and provide all major knowledge about credit transfer and course recognition in European universities. It must combine the features of an inexpensive consulting system and contain enough normative information about credit transfer and course recognition in the universities.

The rest of the document describes the architecture of the system, basic models and data structures, usage issues and further development directions of the system.

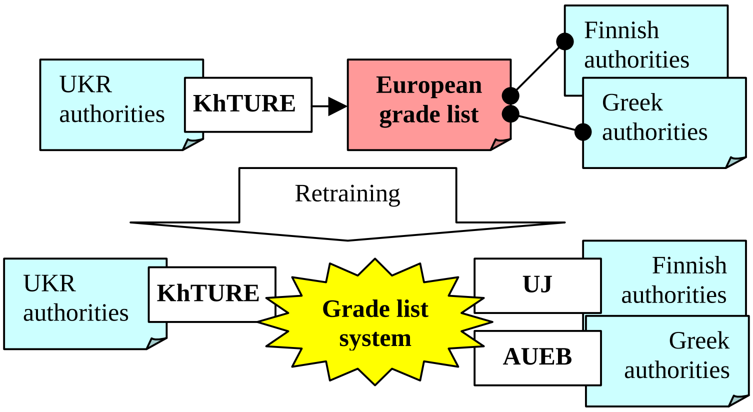

Figure 1. The new look on credit transfer and course recognition process

The technology for the system

The wide target user group of the system and limited development resources available for the team highlighted several requirements for the technology to be used. The system might be:

- Web-based do guarantee easy access of a wide community of users;
- Open and have easy understandable architecture and principles because it is posed as a kind of standard consulting system;
- Scalable to include new members easily;
- Easy and clearly documentable to be acquired by the community;

The most suitable technology for such a system is XML toolkit for Internet-based data storage, representation and manipulation. We selected XML technology because

- It provides a set of standards, recommended by standard organization (WWW consortium www.w3c.org);
- XML is already a standard technology for information interchange;
- All components of the system might be represented in plain text files, easy to read and understand;
- It is already accepted by software vendors and packages;
- It is principally web-based and open;
- And it comes with a complete architecture.

The basic XML toolkit includes four main components:

- XML files with actual data;

- DTD or XSL with structure for the XML-data. All users are forced to keep this structure;
- XSL style sheets define the way to browse the XML-data;
- Standard means for web interface (CGI and JavaScript) for the system.

The way how the system must work is presented in Figure 2.

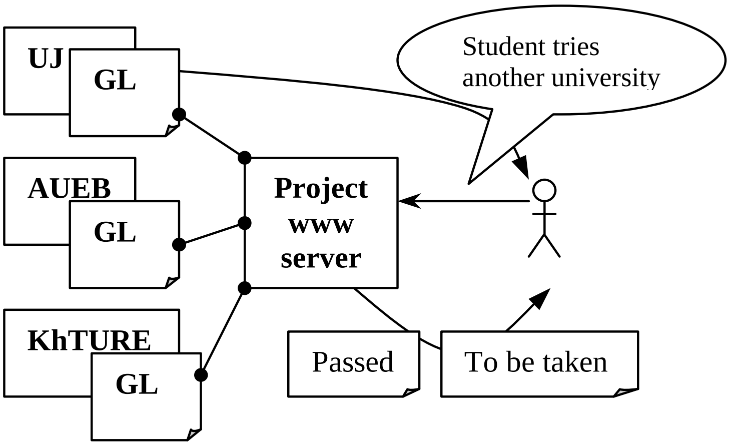

Figure 2. The way of system functioning

The consortium of participating universities uploads their grading requirements and course offering to the system web site. The data is presented in a special format and processed by the web server. We expect a user (a student, or an administrative staff officer) to browse the web site and to communicate with the system through web user interface. The student receives his list of passed courses from his department in a special XML-based format and uploads it to the server. Then he selects the university he wants to try to study (or get a degree), and receives the two reports: the list of courses that are recognized by the university he tried, and the list of courses in might pass to get a degree in the latter university. The system will contain information about course correspondence, mutual course recognition by the universities, and grade requirements of the universities.

In a similar way the system will consult administrative staff and provide them with the report on curriculum similarity.

The Architecture

The schema of file interrelation is presented in Figure 3.

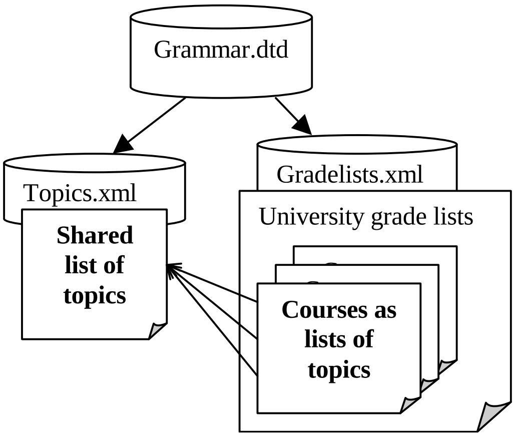

Figure 3. The files

The architecture of the system is presented in Figure 4.

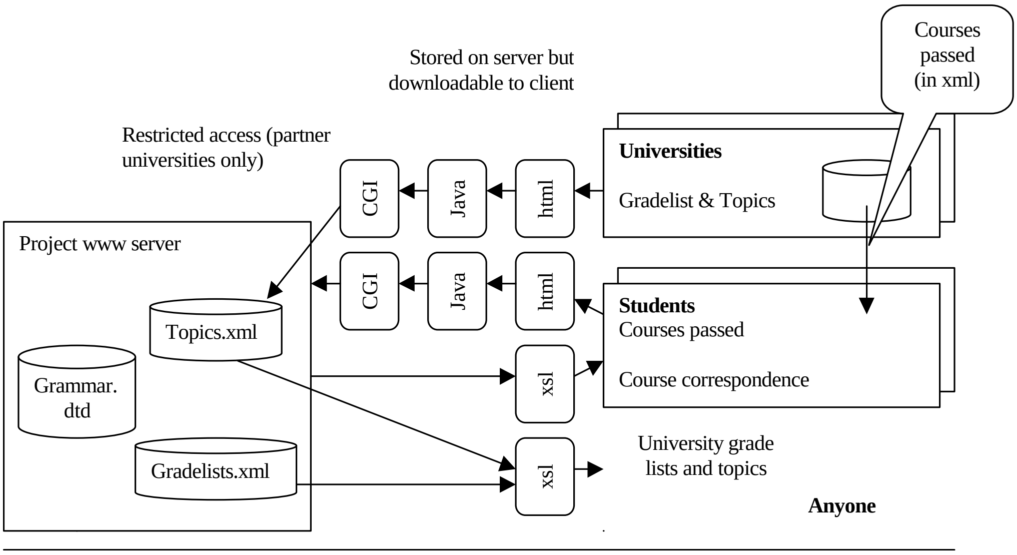

Figure 4. The architecture

The Data Structures

The structure of the files used in the system are defined by the DTD (Document type definition), that is presented in Appendix A. Basic ideas of the data structure used are presented in Figure 5. The data include both normative information (Grade requirements) and less informative information (course offering and course recognition and correspondence).

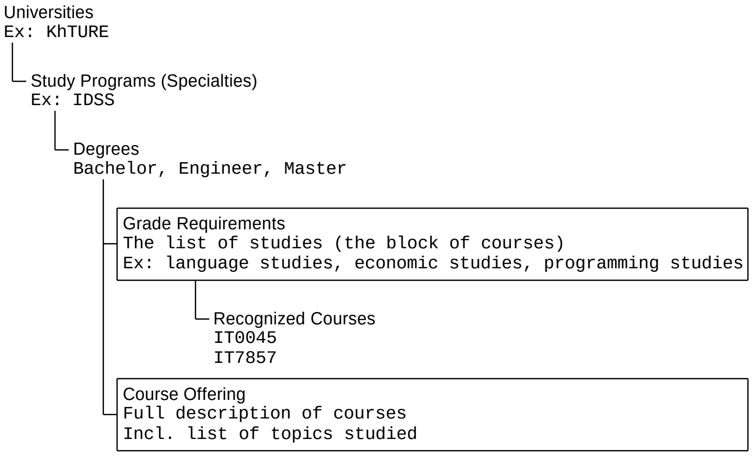
Universities
Ex: KhTURE
Study Programs (Specialties)
Ex: IDSS
Degrees
Bachelor, Engineer, Master
Grade Requirements
The list of studies (the block of courses)
Ex: language studies, economic studies, programming studies
Recognized Courses
IT0045
IT7857
Course Offering
Full description of courses
Incl. list of topics studied

Figure 5. The basic data schema

The examples of basic files are given in Figure 6 – Figure 8.

```
<Institution>
<Nickname>KhTURE</Nickname>
<GradeScheme>
        <Specialty>ISOI</Specialty>       …
        <Degree>
            <Title>Bachelor</Title>       …
                <Studies>
                    <Title>Software Engineering</Title>
                    <Credits>61</Credits>
                    <CompulsoryCourses>
                        <CourseID>
                            <InternalID>SE25</InternalID>
                            <UniversalID>DB0001</UniversalID>
                        </CourseID>
                    </CompulsoryCourses>
                    <OptionalCourses/>
                </Studies>
…
```

Figure 6. Degree requirements of KhTURE

```
<TheCourse>
  <Container>
    <ID>SE 25</ID>
    <Title>Operational Systems and Data Communications</Title>
    <Author/>
    <Organisation>KhTURE</Organisation>
    <Description>The course presents...</Description>
  </Container>
  <UniversalID>DB001</UniversalID>
  <Level>undergraduate</Level>
  <Credits>4</Credits>
  <GradeType>
    <Exam/>
  </GradeType>
</TheCourse>
```

Figure 7. Course Offering of KhTURE

```
<Institution>
<Nickname>UJ</Nickname>
<GradeScheme>
  <Specialty>IT in Economics </Specialty>  …
    <Degree>
      <Title>Bachelor</Title>
      <Credits>180</Credits>
      <Studies>      …
          <Title>Major subject</Title>
          <CompulsoryCourses>
            <CourseID>
              <InternalID>TJTC10</InternalID>
              <UniversalID> DB0001</UniversalID>
            </CourseID>
```

Figure 8. Degree requirements of UJ

Course Recognition

The system aids the human user to define course correspondence between the universities on four levels. The Four Levels of course recognition are:

1. Degree recognition. *For example, University 1 can accept the Bachelor degree assigned by University 2. On this level the system provides only consulting features for human-assisted course recognition.*

2. Studies recognition. *For example, University 1 can accept can accept Programming studies or Language studies that are offered by University 2. On this level the system also provides only consulting features for human-assisted course recognition.*

3. Course recognition. *The universities define the mapping 'local course'   'standard course' and if two courses are defined by the universities as equivalent to the same standard course then they are recognized. On this level the system analyses course correspondence automatically (assuming that humans provided initial correspondence of the courses held in the universities to standard courses).*

4. Topic recognition. *This level contains automatic generation of course correspondence report based on the topics studied in the courses. The two courses that consist of the equivalent list of topics studied are considered as equivalent. On this level the system also assumes that humans defined the correspondence between the particular topics and the standard topics studied in computer science.*

For example course recognition on Level 3, given the information from Figure 6 – Figure 8 will be as follows. The system will understand that the studies 'Software engineering' and 'Major Studies' differ in name and amount of credits, but the courses **SE25** can be recognized as **TJTC10** because both are equivalent to the same **DB0001** 'standard' course.

Conclusions

The developed data structures allow efficient representation of grade requirements and course offering. The next steps to be taken are:

- implementation of the test web-site with 2-3 study programs;
- verification;
- practical usage on a small group of students;
- finalizing the web-site;

promotion of the course recognition system.

Appendix A

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.0 (http://www.xmlspy.com) by Boris Omelaenko (Home) -->
<!-- The web system for credit transfer and course recognition, for Tempus Tacis Project UM-CP-20560 -->
<!-- The structure of the documents used in the system -->
<!-- Developed by  Borys Omelayenko (Vrije Universiteit); www.cs.vu.nl/~borys; borys@cs.vu.nl -->
<!-- This file is a structure for supplements.xml (the list of grading rules and course offering uploaded by the universiites); topics.xml (the list of topics studied in Computer Science); courses.xml (the index of course offerings and descriptions); and for the files with the coursed passed by a student, he receives these files from the administrative staff of participating institutions -->
<!--The file has a Container and might be either a Topic list or a Supplement (course and grade list)-->
<!ELEMENT Root (Container, (TopicList | Supplement*))>
<!--The file can contain only the supplements of one type: either Person or Institution. It is prohibited to create files that mix two types of supplements-->
<!--The Topic list might contain one or more topics-->
<!ELEMENT TopicList (Topic)*>
<!ELEMENT Topic (Container, Hours, References*, RequiredTopicID*)>
<!--Container includes slightly extended Dublin Core Metadata set of attributes-->
<!ELEMENT Container (ID, Title, Author, Organisation, Address?, Description, Date?, URL?, Language?, Format?, Area?, Keyword*)>
<!--Literature references may be used within topic or course descritions-->
<!ELEMENT References (Authors, Title, Year, URL)>
<!--Diploma supplement, for the university contains grading scheme and course offering for a particular program; for a student contains a list of courses passed (to be issued to the student in XML by the faculty)-->
<!--Most informative are the Holder attribute (contains grading scheme) and the CourseList attribute (contains course offfering)-->
<!ELEMENT Supplement (Container, Holder, CourseList, Certification)>
<!ELEMENT Holder (Institution | Person)>
<!ELEMENT Person (FamilyName, GivenNames, Birth, StudentId, Institution, LanguageOfInstruction)>
<!ELEMENT Institution (Nickname, Description, GradeScheme, URL)>
<!--The grading requirements of the university; contains a list of grades assigned and the grade structures-->
<!ELEMENT GradeScheme (Specialty, Title, Description, Degrees, ModeOfStudy, NormalProgramLength, ProgramRequirements, FinalDegree)>
<!ELEMENT Degrees (Degree)*>
<!ELEMENT Degree (Title, Description, Credits, Studies+)>
<!-- Fields/studies, number of credits requred, and the list of compulsory courses and recognised optional courses-->
<!ELEMENT Studies (Title, Description, Credits, CompulsoryCourses?, OptionalCourses?)>
<!ELEMENT CompulsoryCourses (CourseID*)>
<!ELEMENT OptionalCourses (CourseID*)>
<!ELEMENT Certification (Date, Signature+, Stamp)>
<!--Course offering by a program (department)-->
<!ELEMENT CourseList (Container?, Course)*>
<!ELEMENT Course (TheCourse, Credits, Grade)>
<!--Course and Topic IDs here stand for Universal IDs-->
<!ELEMENT TheCourse (Container, UniversalID, Level, Credits, GradeType, Hours?, Specialty?, TopicsID*, RequiredCourseID*, ReplacementCourseID*)>
<!ELEMENT CourseID (InternalID, UniversalID)>
<!---->
<!ELEMENT Birth (Date, BirthPlace, BirthCountry)>
<!ELEMENT Date (Day, Month, Year)>
<!--The Holder-->
<!ELEMENT FamilyName (#PCDATA)>
<!ELEMENT GivenNames (#PCDATA)>
<!ELEMENT StudentId (#PCDATA)>
<!--Birth-->
<!ELEMENT BirthPlace (#PCDATA)>
<!ELEMENT BirthCountry (#PCDATA)>
<!ELEMENT Day (#PCDATA)>
<!ELEMENT Month (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
```

```
<!ELEMENT LanguageOfInstruction (#PCDATA)>
<!--Contents and Results Gained-->
<!ELEMENT ModeOfStudy (#PCDATA)>
<!ELEMENT NormalProgramLength (Years, HoursPerYearMin, HoursPerYearMax)>
<!ELEMENT Years (#PCDATA)>
<!ELEMENT HoursPerYearMin (#PCDATA)>
<!ELEMENT HoursPerYearMax (#PCDATA)>
<!ELEMENT ProgramRequirements (#PCDATA)>
<!---->
<!ELEMENT Signature (Name, Position)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Position (#PCDATA)>
<!ELEMENT Stamp (#PCDATA)>
<!--Consortium members: do not create your own topic lists, add your topics into shared project database!-->
<!--Course, includes list of topics. Each university has its own list of courses-->
<!ELEMENT TopicsID (#PCDATA)>
<!ELEMENT RequiredCourseID (#PCDATA)>
<!ELEMENT Specialty (#PCDATA)>
<!--Container-->
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Organisation (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!--Course and Topic descriptions-->
<!ELEMENT ID (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Level (#PCDATA)>
<!ELEMENT Credits (#PCDATA)>
<!ELEMENT GradeType (No | Test | Exam)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT Hours (#PCDATA)>
<!--Grade types-->
<!ELEMENT No EMPTY>
<!ELEMENT Test EMPTY>
<!ELEMENT Exam EMPTY>
<!--List of courses student passed: course ID, Date, Grade-->
<!ELEMENT Grade (#PCDATA)>
<!ELEMENT Format (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT ReplacementCourseID (#PCDATA)>
<!ELEMENT Area (#PCDATA)>
<!ELEMENT Keyword (#PCDATA)>
<!ELEMENT Authors (#PCDATA)>
<!ELEMENT RequiredTopicID (#PCDATA)>
<!ELEMENT FinalDegree (#PCDATA)>
<!ELEMENT UniversalID (#PCDATA)>
<!ELEMENT InternalID (#PCDATA)>
<!ELEMENT Nickname (#PCDATA)>
```