USOS
Student Management Information System For Polish Universities

Janina Mincer-Daszkiewicz

jmd@mimuw.edu.pl

Department of Mathematics, Computer Science, and Mechanics

Warsaw University, Poland

Agenda

- How it all started — Tempus NET UM_JEP-14461-1999
- USOS — Requirements analysis
- USOS — Software design and implementation
- USOS — Some implementation details
- USOS — Software deployment
- UCI — Consortium of Polish Universities
- USOS — Education in Software Engineering (if time allows)
- Summary

How It All Started

- Tempus *New Educational Tools* (NET) UM_JEP-14461-1999
  November 1999 – December 2001
- 17 Polish (state) universities
- **Objective**: deploy common *Student Management Information System* (SMIS) at all participating universities
- **Early observation**: there is no such ready-to-use package on the market
- **Another observation**: software companies would charge a lot and lack necessary expertise
- **Conclusion**: do it yourself → Department of Mathematics, Computer Science and Mechanics, Warsaw University
- **Final effect**: USOS
  (*University Study-Oriented System* or *University SOS*)

USOS — Requirements Analysis

USOS should support handling the following data and activities:

- students’ and teachers’ personal data
- study programs and requirements of degree certificates
- course catalog
- course registration
- class schedules
- dormitories
- tuition and financial aid (what we pay studens)
- paid courses (what students pay us)
- issuing documents, gathering statistics, producing reports, etc.

USOS — Requirements Analysis

Non-functional requirements:

- university-wide
- flexible (configurable, adaptable to programs and procedures of participating universities, supporting standard and non-standard solutions)
- open (easily adaptable to changes)
- compatible with ECTS standards
- user-friendly (also for not IT professionals)
- support easy access for students, faculty members, and administration officers
- ensure security of data and lower administration expenses

USOS — Software Design

To meet the (almost) contradictory goals the following system architecture was designed:

- main database accessed on a strictly limited bases by administration staff — in Oracle technology (Oracle 8.0.6, Oracle Designer/2000, Oracle Developer, Oracle Forms, Oracle Reports)
- few Internet databases accessed through web browsers by academic teachers, students — in open-source technology (MySQL, PHP, Apache)

USOS — Some implementation details

Database synchroniser:

- adapts automatically to the changing structure of the Oracle database
- connects remotly to both databases, reinstalls chosen tables at the website (if changed), synchronizes tables contents — only changed records are exchanged
- in case of conflicts stops automatic data transfers and asks the operator for the decision

USOS — Some implementation details

Authorised access to subsets of data:

- users play various roles in real life (meaning various privileges)
- in USOS roles (meant as privileges to select, insert, delete, update rows and columns of database tables) can be defined by **Roles Administrator**
- roles can be freely granted to USOS users (they may be given various roles, one of them is default)
- USOS interface is transparent to the system of roles
- roles handling does not require Oracle programming

USOS — Some implementation details

Data filtering:

- solves the problem of having to deal with the excessible amount of data
- filters allow for filtering data user is authorised to handle
- filters are available from Oracle forms with a button
- users may have various filters available on various forms (one is used as a default)
- users can define filters by themselves using USOS interface
- filters handling does not require Oracle programming

USOS — Some implementation details

Checking degree requirements:

- module for student assessment supports flexible study programs
- USOS allows to state degree requirements for some predefined profiles and for individual students who dynamically build their profiles
- the requirements can be stated by pointing a fixed set of courses or a given number of compulsory courses picked up from some predefined group
- USOS allows for automatic checking whether the requirements have been met
- status of a student is updated accordingly
- there is still place for manual verification and update of automatic decisions (human is an ultimate authority)

USOS — Software Deployment (at Warsaw University)

The good ...

- Rector makes and announces the decision to deploy USOS at the university level and stop support for old desktop-like systems (e.g. the central financial system will only accept data on student financial aid in a format agreed upon and implemented in USOS)
- Deployment team is established. It involves system developers, university administration officers (student affairs, financial affairs), university computer systems administrators and programmers
- Deployment may proceed in stages, according to faculty priorities
- University authorities (rector, deans of the departments) and administration officers are strongly motivated to achieve the goal
- People seem to understand that if not now then never ...

USOS — Software Deployment (at Warsaw University)

The bad ...

- Departments of Warsaw University vary substantially in size
- Departments have high degree of autonomy
- Departments have different priorities
- The informatization of departments is different (often very low): hardware, software, peopleware
- Deployment of a large software system demands high organizational effort and a lot of (extra) work
- People are busy, salaries are low etc. ...

USOS — Software Deployment (at Warsaw University)

SUCCESS

depends on

PEOPLE

USOS — Software Deployment (at Warsaw University)

Steps:

- Common dictionaries for USOS and payroll and Human Resources database are defined
- The dean of the department asks (on paper!) for system deployment
- Departamental deployment team is established
- Staff personal data are transferred from HR database
- Students data are transferred from old desktop databases or spread sheets (only some faculties have such data in electronic form)
- Course and degree catalogs are built
- In the meantime USOS is installed on departamental computers and people are trained (at their work place)

- Roles and filters for various administraton staff members are defined
- Administration procedures are verified along the way

UCI — Consortium of Polish Universities

- 2002: UCI (Uniwersyteckie Centrum ds. Informatyzacji) - consortium of Polish universities (14 out of 17) is established. One of the main goals is to support further development of USOS
- Warsaw Politechnic joins (the biggest higher education institution in Poland)
- UCI partners pay membership due to fund system development
- **Committee for USOS** sets task and priorities
- **Working Team for USOS** coordinates all design and programming work, maintains software, issues new distributions, offers help desk
- In the future new USOS modules may be developed by: in-house university teams, student groups, commercial software companies

USOS — Education in Software Engineering

- The main goal of software engineering courses is **programming in the large**
- Those goals can hardly be achieved by artificial academic projects
- Only real-life software projects offer real experience
- USOS gave us opportunity to integrate educaton with production of commodity software
- This is definitely not the only such project ...
- ... but the scale and multilevel integration of design, development and deployment activities with education makes it really an interesting and succesfull experiment

USOS — Education in Software Engineering

Advantages (for students):

- take part in all stages of real software development process, including software deployment and maintenance
- learn how to be team members, how to collaborate with colleagues from other groups, study programs, faculties
- learn how to use professional tools, meet quality and performance requirements, apply standards, feel responsibility for the product
- get involved in the project even not participating in it personally: test, give advice

USOS — Education in Software Engineering

Advantages (for academic community):

- obtains high quality software
- owns source code
- changing requirements are not such a problem

USOS — Education in Software Engineering

Disadvantages:

- project demands much higher involvement of the academic staff, substantially exceeding the ordinary academic obligations
- necessity of giving priority to academic goals may be hard to reconsile with project goals (e.g. students are not fully productive during summer breaks or exams, but anyway new modules must be delivered on time)

Summary

- At the moment USOS is used on daily basis in four universities (at pilot faculties), a few others are at various stages of deployment
- At Warsaw University we started academic year 2002/03 with USOS at four faculties. Others will join soon ...
- **UCI Committee for USOS** works out details of the optimal organizational framework for future management of system development
- A few software companies are interested in integrating their (commercial) products with USOS and offer deployment services
- Some private higher education institutions asked for demo version
- Master and Bachelor Theses: 12 by now, many more to come

The End