

Information Systems Course Case Study: Wireless Campus Portal

Vangel Ajanovski

*Institute of Informatics
Faculty of Natural Sciences and Mathematics, Skopje
ajan@ii.edu.mk*

Information Systems Course

- Contents:
 - Concepts of IS Development, Components of an IS
 - Design Methodology, Structural and non-structural analysis
 - UML
 - Project Management (PERT and GANTT charts)
 - Reverse engineering
- Workload
 - 2h lectures
 - 1h theoretical exercises
 - 2h practical in-lab exercises
- 2003/2004
 - 30 students in the 4th year
 - Students are divided in 2 groups for lab. exercises

IS course materials and lab work

- Most of the SE course materials and books dealing with software analysis and design
 - Give quite a good presentation of the theoretical concepts and
 - Are a great value for reading comprehension, but...
 - Usually they lack full-scale practical examples with real-world scenarios in larger SW development projects
- Even more evident in IS development
 - Projects usually involve huge software development and managerial efforts
 - It is hard to present everything in detail and still confine to the volume of one or two-semester course
 - Authors, therefore, revert to simpler examples or even present unrelated bits of a large IS

...

- We want a real information system as a case study
 - based on a real already developed software system
 - complex enough to include several intertwined SE concepts, methods and tools
 - simple enough to run in a simulated environment
 - interesting, fashionable problem to solve



The lab project overview

- **Wireless Campus Portal (WCP)**
 - Consists of Web pages, WAP pages and SMS service providing useful and interesting info for the students, teachers, administration, etc...
 - Unites standard functions residing in the campus-wide IS
 - Schedule information (viewing timetables of scheduled lectures, exams, important events)
 - News and important announcement subscriptions (administrative announcements, notifications and reminders of schedules, etc.)
 - Enrollment for taking exams and presentation of exam results
 - Finding a free workstation in campus laboratories to do homework assignments, and others
 - Personalized - enables users to choose the information they need and the way they see it
 - Provides facilities for user intervention within the functions of the information system (for teachers and administration)

Project Phases

- **Phase 1** consists of development of test-bed web and wireless applications implementing functions of general interest:
 - Schedule information
 - News and important announcement subscriptions
- The students have to start this development with assessment of user needs and proposition of solutions in a detailed technical specification, continued with an established project management plan.
- **Phase 2** introduces reverse-engineering concepts to the students, and they go on with exploring and reverse-engineering existing work, given to them in source code
- **Phase 3** – semestral projects, students split in groups and add new features to the system model
- **Phase 4** – (optional) rebuild the system with new features

Phase 1

- With the start of the project, students are presented a partly completed design model, specified using UML
 - Created with reverse engineering of an existing platform
 - This platform was built completely in Java and itself took a substantial amount of work, partly in customizing the third-party open-source libraries that were used.
 - This is why it was decided to try a combination of forward and reverse engineering to specify the further development.
- Rational Rose is chosen as an UML modeling tool
 - There is possibility of acquiring a full version for free
- Rational Unified Process (or a variation) is used in order to finish the analysis and design phases of the new platform

...

- Students should start with business-level use-cases. Depending on the number, importance and complexity of use-cases some of them will be left aside and the work will focus on the rest
- While the model mostly revolves around describing the existing portal platform, it also serves to note some of the future feature requests. Students should prepare a detailed system model, precisely defining all of the system level use-cases that should be designed and developed as application features.
- Students have to prepare a complete use-case documentation set, including:
 - Description of the use-case in more detail
 - Pre- and post-conditions
 - Alternative scenarios
 - Activity diagrams for the most complex scenarios

Phase 2

- Phase 2 is done in two global steps
 - Reverse-engineering of SMS service system
 - Reverse-engineering of Web and WAP applications
- The gathered information from the already developed classes should be linked to use-case realizations and decided on other use-cases that will be realized.
 - This is lab work and is done in order to practice the process of reverse-engineering Java code, web and WAP generating JSP-s, and as a result get UML class diagrams
 - Students are given a package of Java source files, together with one open-source library and access to several required packages and other libraries.
 - The final result should be a class diagram(s) of associations of all the classes forming the background application processing services.

...

- The second step involves reverse-engineering of the web and WAP applications.
 - In this exercise it is important to show the associations for static and dynamic pages, how they link together and the results of the pages that are processed at server side.
 - The aim of the exercise is to point-out reusable and problematic parts of the web and WAP application and then do rework
- The third step is logical reverse design of the current realization
 - Students should try to mark classes that perform same elementary functions and
 - agree on the use-case realizations performed with these classes
 - Agree on the coverage of this realization with regard to the newly built model

...

- Sequences of actions
 - Sequences of actions as responses to user-given commands cannot be reverse-engineered, it is left as a task for the students to map the different class methods in sequence or collaboration diagrams specific for each use-case realization
 - This is a very useful process, and probably the hardest part of all the work, since they have to be able to read code written by others and provided with just minimal set of documentation
- Components and building blocks
 - As part of the reverse-engineering process all executable and library components should be discovered and a proper mapping of all classes should be performed.
 - It is of great significance to find out dependences among components in order to be able to build them in proper order and link together in packages the classes with similar or joint functions.

...

- During the last part of the process students are assigned with the packaging of components and deciding on the server platforms.
- This task also involves the creation of a deployment architecture.
- There are other steps that should be done in parallel to the packaging, like:
 - Deciding on the type of deployment – whether manual or automated;
 - Researching and deciding on the deployment operating system and overall software platforms and
 - Managing activities that consist the project development, deployment and documentation tasks.

Phase 3

- Divided in several study groups (depending on the total number of students)
 - Each of the students will attack the system from a different perspective and try to find resolutions for each imperfection
- For example, 30 students divided in 5 groups
 - One group could try to do reverse-engineering and refactoring of the code
 - The second group could try to fill-in missing design parts starting from the use-cases
 - The third group could work on a functional decomposition and perform mapping of the modules, and class operations to elementary functions
 - The fourth group could concentrate on the functional testing
 - The fifth group could go on the code documentation.

Students still have to start with something