Supporting Academic Administrative Processes in the Post-Bologna Era – the Experience of the Swiss Federal Institute of Technology (ETH)

Giorgio Broggi, Armin Wittmann

ICT-Services, Swiss Federal Institute of Technology (ETH), ETH Zentrum CLU, Clausiusstrasse 50, CH-8092 Zurich, Switzerland
broggi@id.ethz.ch
wittmann@id.ethz.ch

Abstract

The Swiss Federal Institute of Technology Zurich (ETH) is developing a new integrated operative information system, based on Oracle DBs connected by interfaces, with standard (SAP, Planon) and individual software solutions accessing them. For the academic core processes, ETH-proprietary applications have been developed, using the J2EE standard and the Sybase Powerbuilder platform for web- and back-office applications, respectively.

Keywords: Academic Operations Applications, SW-Development, J2EE.

1 Introduction

The Swiss Federal Institute of Technology Zurich (ETH), one of Europe's leading research-oriented technical universities with more than 12,500 students (one quarter of whom are attending PhD programmes), 330 professorships, and more than 5,000 research staff, is facing rapid change. On the one hand, as in most academic institutions in Continental Europe, the implementation of the Bologna Declaration makes a drastic re-organisation of the teaching curricula, and, therefore, of all related academic processes, absolutely necessary. On the other hand, the ETH has decided to take advantage of this reorganisation in order to re-position itself as an institution focusing mainly on graduate education. The goal is to achieve a balance between the number of undergraduate and graduate students as quickly as possible by recruiting more qualified candidates on the international market place. Most of the new Master Programmes will therefore be offered in English. The ICT-Services have been urged to act as enablers of this extensive change by implementing new applications that would allow support of flexible back office processes and extended participation of students and academic staff in the handling of teaching and curricular data, thus allowing for differentiation and augmentation of the offered academic degrees.

2 The challenge

When the project was launched in summer 2000, the re-engineering of the old terminal oriented administrative application had just begun. By then, only the student data had been moved to a modern database-system. Moreover, the ETH-Management had decided that the Bologna implementation should be completed by autumn 2005 and had launched the ETH-World initiative, aimed at creating a new Virtual Campus for research and teaching at the ETH. ETH-World required the ICT-services to build the infrastructure needed in order to render many of the fundamental academic activities location independent: this was clearly not compatible with the old, centralised, paper-based administrative processes. We basically had five years time to re-implement data repositories and applications, not knowing exactly how the ETH would organise its teaching in the post-Bologna era nor what the related administrative processes would look like. The challenge for the ICT-Services was big indeed.

3 Our approach

According to the general requirements that we have described, the mission of the project had to be twofold: first of all, it had to provide the ETH with a comprehensive, flexible information system. Secondly, the new system had to be designed in a way that would allow direct, location independent access to the central database and direct participation in the academic operations to students and academic staff, in harmony with the ETH-World vision. Moreover, since the ETH was going to offer very differentiated degree curricula at graduate level the new academic operations applications had to be able to support them.

Since the academic operations of the ETH are quite strongly centralised (i.e. there are no "Schools" or "Faculties", courses and exam schedules are planned centrally, and only some of the administrative sub-processes take place within the Departments), it was decided to establish a mixed project team with the ETH Rektorat (central administrative unit responsible for the academic operations, including the student Registrar's and the Faculty Services).

Our leading idea was that the new IT-applications should support administrative (sub-) processes comprehensively, not just allow for web based access to the data. This means that part of the business logic had to be moved to the Web. Moreover, we wanted to avoid any sort of "big-bang" deployment of new software, since we knew that the administrative staff were already dealing with rapidly

changing regulations and could not cope with the unavoidable problems related to large-scale system replacements. Accordingly, we decided to develop the new infrastructure within small sub-projects, each of which addressed a specific academic sub-process. This approach makes the building of several provisional interfaces unavoidable, but allows for immediate productive exploitation of the implemented application. In a phase of rapid business process redefinition, this is a major advantage, since the realisation of the components whose specification is still under discussion can be postponed without jeopardising time-critical deployments. Moreover, later developments can obviously profit from the lessons learned from the productive usage of the earlier ones.

4 The new IT-environment

Our IT-environment consists of an integrated operational information system (OIS) with a central data repository also serving the authentication and authorisation infrastructure (AAI), and several applications accessing it. Whereas for finance-, HR- and facility-management we deployed standard applications (SAP R/3, Planon1), we decided to develop the "Core-Components" (handling of names and addresses of ETH-members as well as organisational data), the AAI, and the academic operations applications as individual software solutions, since none of the available commercial products met the requirement of implementing full, detailed support for the ETH-specific processes.

4.1 Data repository

The first step in the realisation of the new IT-environment has been the development of the central data repository (Fig.1).

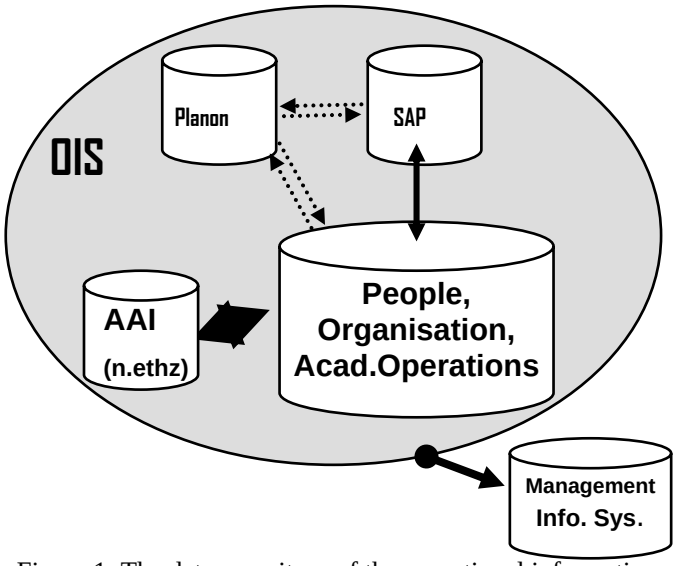

Figure 1: The data repository of the operational information system.

This is based on Oracle databases (DB). As it is required by the standard products we have deployed, we run three DB-instances (ETH-own, SAP R/3, Planon) connected by on-line (drawn-out lines in Fig. 1) or off-line interfaces (dotted lines). The components developed by the ETH contain about 500 tables. The core elements of the OIS are the tables in which the particulars of all ETH members (students, faculty, academic and administrative staff), their addresses and affiliation data are kept. Since in this DB the ETH members are modelled as individuals, not as students or staff members, with their detailed affiliation statuses as attributes, we are in a position to use a simple (on-line) replica of the data to set up the authentication and authorisation infrastructure (AAI).

4.2 Authentication and authorisation infrastructure

Originally developed for the Students' e-Mail Service, the ETH AAI system "n.ethz" has gradually become the key to all central IT-resources. Based on the OIS-data, the system automatically handles the creation, expiration and deletion of personal identification records for access to online resources ("services"). Default access-rights are also automatically assigned to individuals according to their roles and affiliations. In addition, administrative roles are supported (superusers, administrators), providing special privileges for the management of defined sets of services for defined groups of users. Accordingly, activities such as setting passwords, creating, locking, and modifying services, creating guest accounts, etc. can be delegated to local administrators. Some examples of current n.ethz services are as follows: the central e-mail and file-system facilities (based on MS Exchange and AFS, respectively), the VPN services, the dialup system, the automated SW-distribution system (IDES), the central inventory services and several groups of web applications.

Technically, the system offers LDAP, Kerberos, Radius and Active Directory authentication. An extensive, sophisticated web user interface makes n.ethz attractive for both users and administrators.

4.3 Portals

The access to the web-applications is presently guaranteed by simple, user-group specific web-pages with some portal-like functionality (see e.g. www.faculty.ethz.ch), which have been implemented in-house according to the J2EE standard. An ETH Portal is presently being developed based on the uPortal framework.

4.4 Web content-management system

The publication of general information related to academic operations is supported by a Web Content Management System based on an ETH-extension of the Silva[2] content management system (WCMS) and the Zope application

[1] www.planon-fm.com/uk

[2] www.infrae.nl/products/silva

server. The WCMS has an interface to the course catalogue (see Section 4.5) that allows for a simple embedding of course information in the web-sites of the relevant faculty members, groups or institutes. New, WCMS-based deployments have rapidly replaced many hardly maintainable departmental sites: very popular features are the simple-to-handle individual homepage generator (with access-rights manager) based on the OIS-data, the MS Exchange connector for the publication of (single or consolidated) calendar data, and the folder-based document publisher.

4.5 Web applications – Web services

The essential academic operational processes requiring interaction between students or teaching staff members and the administration (term registration, selection of a degree course and choice of classes within it, change of addresses or other personal data, planning-workflow for each course schedule, handling of course and exam related data, room reservation etc.) are supported by comprehensive web-applications that we implemented according to the J2EE standard. A further development in this group is the ETH course catalogue (www.course-catalogue.ethz.ch).

The choice of a technology for the development of the web application has been determined by the requirement of supporting 200 concurrent users or more with acceptable performance.

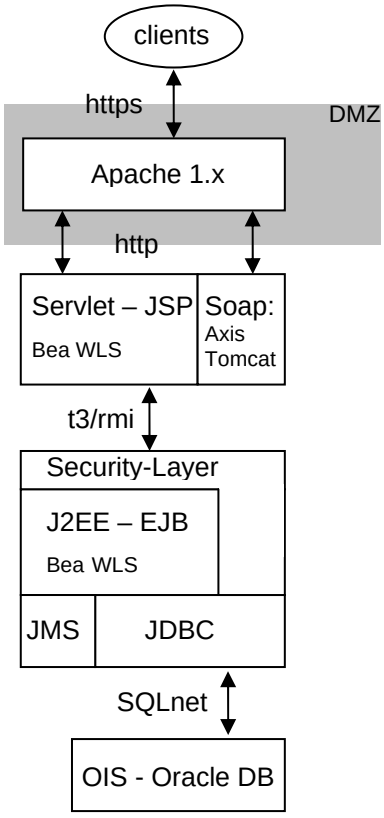

Figure 2: Architecture of the web applications.

Since we expected very strong load peaks at the beginning and at the end of each term, the technology of choice had to be scalable and established. Additionally, in order to achieve a reasonable independence from hardware and software producers, we opted for an architecture based on the Java 2 Enterprise edition (J2EE). We decided to implement the full 3-tier architecture with the Oracle OIS DB as persistence tier, the EJB container as business-logic tier and the Servlet/JSP frontend as presentation tier (Figure 2). After a careful evaluation of commercial products we opted for BEA Weblogic Server (WLS) with Apache as a Web-Frontend and as SSL-Encryption engine. The JSP and EJB containers are deployed in separate WLS instances.

The libraries and design patterns we use are the Struts Framework[3] for Servlet/JSP programming (Model View Controller Pattern), and the log4j-library[4] for the logging and monitoring functions. Further, in order to improve the scaling properties and reduce the complexity of the applications, we decided not to use Entity and Stateful Session Beans. For asynchronous messaging (typically e-mail) we employ the JMS API (inclusive persistence binding).

Another fundamental requirement that determined the general architecture of our applications was that of delivering data on-line to external environments without opening the DB to external users. For the standardisation and quality control of the data that are needed for the Diploma Supplements, the ETH decided to collect and store the course descriptions in the central DB. These data had been traditionally managed by the departments and published on paper. In order to guarantee their availability for the departmental environments, a Web service is offered, which permits the periodic download of the catalogue data. Our implementation is based on the SOAP technology, whereby the Apache Axis implementation is used as a supporting Tool[5]. This technology will also be used for data exchange between the OIS-environment and applications run by other ETH and external organisations. A further interesting usage of web services is that of accessing server business logic from within traditional windows clients, in order not to implement redundantly the same business rules in different environments. This is possible for instance in Powerbuilder 9 (see Section 4.6) using the native Web-Service interface. The different ways in which different client types can access the data are summarised in Figure 3.

The requirement of extending the business-layer access to authorised web services (not only the Servlet container should be granted access to this layer) is met by implementing control access to the business components (and, correspondingly, to the data themselves) in the Security Realm of the WLS and not at the level of the JSP/Servlets. Each access to an EJB is either allowed or blocked according to the finely granular access rights of the corresponding security roles. The security scheme is stored in the DB.

[3] HTTP://jakarta.apache.org/struts
[4] HTTP://logging.apache.org/log4j
[5] HTTP://ws.apache.org/axis

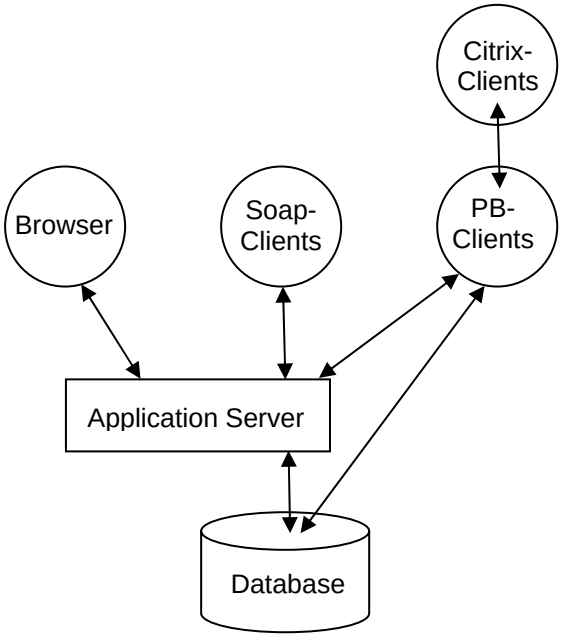

Figure 3: DB-access from different client types.

The interface between the Servlet/Soap containers and the EJB container is kept very "narrow". The data are exchanged by means of only one entity, the ValueObject, that allows for parameter (InputObjects) and result (OutputObjects) transmission within the command structure that implements the mapping (Facade) between EJB functions and command structure (CMDs). The individual CMDs and/or a combination of such CMDs (CMD blocks) contain the implemented business logic and are called by the Servlet Controller. Each CMD block is executed within a transaction (container-based transaction). All data modification and flow-related decisions of the web application are made (or made available) exclusively within these CMDs.

The data flows between the different components of a web application are described graphically in Figure 4.

While Eclipse6 and Macromedia Dreamweaver are used by our development team for Java-code and JSP development respectively, our major development aid is the so-called EETool, which was written at the beginning of the project. In this tool, the entire Struts Mappings are recorded, allowing for the automatic generation of the struts configuration descriptor (struts-config.xml). In addition, the tool manages all constants and error messages, all the SQL statements, as well as all the commands (with their IDs) that are used in each application. All listings mentioned above are used in setting up automatically generated Java classes, which in turn flow in the CVS (control and versioning system) of the application.

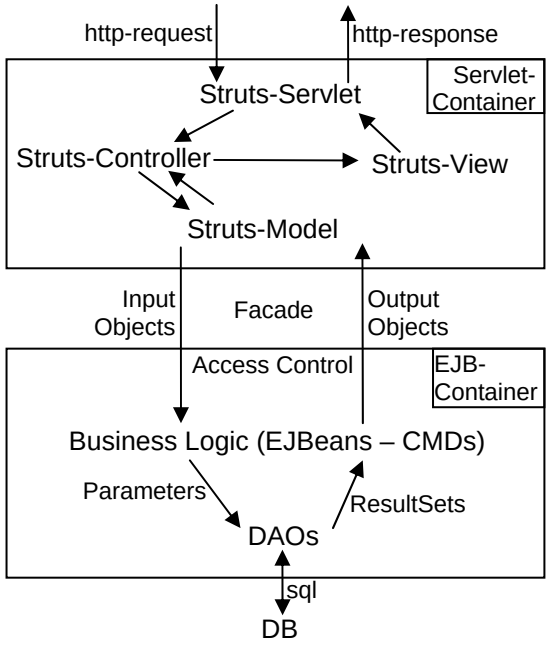

Figure 4: Data flow and interfaces within a web application.

In some steps of the administrative processes that we implemented, the user (either a student or a teaching staff member) is permitted (or required for signed confirmation) to print some data extracts. The corresponding technical solution is the dynamical generation of PDF files. We have chosen to implement the PDF generation purely with Java code, using a library that was completely integrated in the Servlet container. The choice fell here on StyleReport of Codeworks7.

For the generation of a version of a web-application we use exclusively ant8 (part of the Eclipse package). The EJB container is deployed as EAR file, the actual Web application (JSP/Servlet container) as exploded WAR. Since embedded texts ought to be changed frequently during productive operations, this is a practical choice, making the JSP-changes possible with little inherent risk.

4.6 Smart clients

All relevant back-office processes are supported by Windows clients that we have developed using Sybase Powerbuilder (PB) technology. The clients access the DB directly via Oracle Net, while in some critical cases write-access is guaranteed by DB Stored Procedures. Unity in data and user handling (look 'n feel) has been achieved by developing an

[6] www.eclipse.org

[7] www.inetsoft.com

[8] HTTP://ant.apache.org

ETH-proprietary framework (ETH foundation class library). Using a framework developed for our needs and purposes gives the users a common look and feel and ease of use among the various PB applications. Developers benefit from encapsulating and re-using functionality among the applications (Business Objects). Access for users on non-windows platforms (some Departments have MacIntosh or Linux desktop systems) takes place via Citrix Metaframe middleware. Three applications have been developed according to this scheme: Liseth-pro for the Students Administration, Sempro+ for the Academic Staff Administration and the Planning of Courses and Terms, and LKW for the Planning and Administration of Exams and Credits. These applications are quite complex and extensive, and implement very ETH-specific academic rules. Therefore, their existence can be considered to be a business advantage for the ETH. Their main functions are listed in Table 1.

Application	*Function*
LISETHpro (Student Administration)	Mgmt of Electronic Applications (back-office functionality), Open/Search for/Modify/Delete Student Records, Mgmt of Particulars, Addresses, Curricular Documentation, Billing, Mgmt of Term Registrations (back-office functionality) Mgmt of Registration- and Exam Deadlines, Mgmt of Multiple-Degree Curricula Communication with Selected Groups of Students, Communication (general): Data Sheets, Mailings etc. Mgmt of Degree-Course and Student Statuses, Ex-matriculation, Mgmt of Compulsory Placements, Mgmt of Funding, Grants etc. Personalisation of Student ID-Cards (Chip-Card), Reports
Sempro+ (Room Mgmt)	Search for Free Rooms/Free Time Slots, Modify Class/Teaching Activity/Meeting Data, Room Reservation, Commit Electronic Room Pre-Reservation (back-office functionality)
Sempro+ (Course Scheduling)	Mgmt of Courses, Classes and other Teaching Activities, Weekly Plan (Courses and Teaching Activities) Course Mgmt: Define Non-Overlapping Groups of Classes and Teaching Activities, Course Mgmt: Classes, Credits, Curricular Information,
	Mgmt of Scheduling Input Data from Teaching Staff, Mgmt of Course Catalogue Input Data from Teaching Staff, Mgmt of Exam Description Input Data from Teaching Staff, Mgmt of Course Structure, Generation of the Course Schedule of the Following Term
Sempro+ (Teaching Staff /Teaching Appointments)	Open/Search for/Modify/Delete Teaching Staff Member Records, Personalisation of Teaching Staff ID-Cards (Chip-Card), Create/Modify/Open a Teaching Appointment, Mgmt of Teaching Fees Mgmt of Applications for Teaching Appointments, Assignment of a Teaching Appointment
Sempro+ (General)	Communication with Selected Members of the Teaching Staff, Communication (general): Data Sheets, Mailings etc. Communication within the Sempro+ User Group (back-office functionality) Scheduling of Planning Activities (back-office functionality), Reports
LKW (Exam Administration)	Mgmt of Degree-Courses (Exam Regulations), Mgmt of Electronic Applications (back-office functionality), Evaluation of Applications and Admissions, Open/Search for/Modify/Delete Exam Activity, Scheduling of Exam Activities, Mgmt of "Block Exams", Mgmt of Degree-Exams, Mgmt of Exam Deadlines, Mgmt of Exam Results (Marks etc.) (back-office function, with several security-related features) Statistics; Ranking-Data, Mgmt Committing / Archiving Exam Results, Generation of Certificates, Handling of Special Cases (Suspension, Interruption, Withdrawal, etc), Communication with Selected Groups of Examiners/Candidates, Reports
LKW (Exam Scheduling)	Under Development (Operational Autumn 2005 / Spring 2006)

Table 1: Main functions of the back-office applications (Smart Clients).

5 IT and more

It is also important to understand that the development of the new Operational Information System and of the Academic Operations Applications is just one aspect of the re-organisation process that we mentioned in the introduction. As we pointed out in the preceding chapters, the relevant new administrative processes are actually being conceived and implemented during the project itself. Since the ETH-Management decided that no student should be forced to change his or her plans and join one of the new Bachelor or Master programmes, the administration is challenged with keeping the old academic processes going while implementing and running the new ones. The new flexible and personalised curricula and the direct interaction with students and academic staff on electronic platforms are also dramatic paradigm shifts that change the sort of qualification the clerical staff must hold. The choice of English as the standard language for graduate studies may further lead to the need of re-qualification or replacement of numerous administrative staff.

6 Conclusions

In this paper we have shown how the ETH-Zurich is developing a new integrated Operative Information System, based on Oracle DBs connected by interfaces, with standard (SAP, Planon) and individual software solutions accessing them. For the academic core processes ETH-proprietary applications have been developed, using the J2EE standard and the Sybase Powerbuilder platform for the web and back-office applications, respectively. The new systems allow for the ETH to offer very differentiated courses at graduate level, in line with the Bologna-reorganisation, while providing location independent access to the central database and direct participation in the academic operations to students and academic staff.

Acknowledgements

We are grateful to Dr. Andreas Dudler, Director of the ICT-Services, who encouraged us to write this paper and revised the manuscript.