PORTING THE SCIENTIFIC ONLINE PROPOSAL SYSTEM OF THE EUROPEAN SYNCHROTRON RADIATION FACILITY TO THE SYNCHROTRON RADIATION SOURCE ANKA

Wolfgang Mexner, Karl-Heinz Cerff, Michael Wuensch, FZK, Germany
Dominique Porte, Lilian Cardonne, Stefan Schulze, ESRF, France

Abstract

The Scientific Management Information System (SMIS) handles around 1000 applications for beam time and per proposal round at the European Synchrotron Radiation Facility (ESRF). It manages the complete workflow of scientific proposals from the initial submission to the scheduling of beam time up to the final reporting on an experiment, hereby simplifying significantly the collaboration between users, beamline scientists, the user-office and peer-review grading committees. The system is based on an Oracle SQL database as well as UNIX and windows web servers. It relies on Java and Microsoft Visual Basic technologies. Due to the similar need to handle proposal submissions electronically, ANKA decided in 2003 to adapt the SMIS to its needs. This decision was mainly based on the proven stability and improvements of the system during the last 10 years at the ESRF. Between ANKA's decision and the start-up of the system at 15 November 2004, about one year of time and manpower were invested to adapt the SMIS. We will present the architecture of the SMIS installed at ANKA and ESRF, the prerequisites for the migration and the unexpected issues that had to be solved. Our contribution will conclude with a discussion on a standardized online proposal system for synchrotron radiation facilities

THE SMIS AT THE ESRF

An Online Proposal System at the ESRF

The European Synchrotron Radiation Facility [1], located in Grenoble, France, is Europe's most powerful synchrotron radiation source.. In 1994, it started to develop its own information system to manage users, proposals and experiments. Based on an Oracle database developed by the Institut Laue-Langevin [2] several years earlier (1986), this client server application was intended to support the user office, which still handled the applications for beam time in paper form. With the increasing number of applications, the handling of paper forms became more and more difficult. The need for an online proposal system arose. The development started in 1999 based on the existing database. It gradually incorporated more and more web based applications for the submission and management of many different types of proposals and experiments.

The ESRF's Proposal Workflow

A standard ESRF proposal's workflow (see Figure 1) for a peer-reviewed experiment follows 4 steps:

1) The main proposer submits an online proposal using the SMIS. The application form contains information about the different proposers, the type of experiment, the necessary technical prerequisites, possible safety risks and references to former experiments and reports. Also a detailed experiment description in document format (word, postscript or pdf) has to be uploaded as part of the proposal.
2) Calls for proposals occur twice yearly: on 1st of March and 1st of September. The applications are collected by the user office, and are dispatched to scientific review committees for evaluation and beamline scientists for their appreciation of technical feasibility. Based on this information, the final decision to accept or decline a proposal is taken and the scientist in charge of the beamline, together with the user office, plans the experiments.
3) The main proposer of an accepted proposal submits further information concerning the experiment. This allows the user and travel offices to organise the users' stay and the safety group to survey the experiment's risk.
4) The workflow concludes with the submission of a report containing the experiment's results.

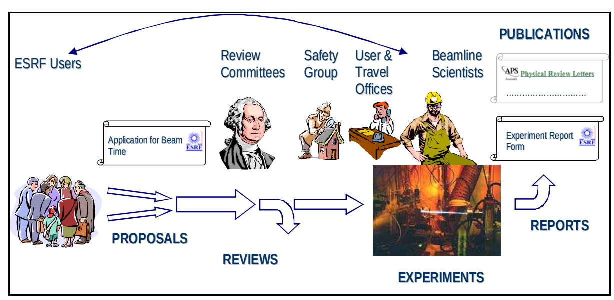

Figure 1: The ESRF's standard proposal workflow. All steps of the workflow are electronically supported by the SMIS. Almost 1000 proposals per call and 3500 users per year are handled.

In addition to the standard proposals, the SMIS manages special workflows for proposals of the industrial group, collaborating resource groups, in-house research proposals and so-called long-term proposals for experiments that span several runs.

The system also incorporates functionality that is not directly linked to the proposal workflow such as e-learning functionality to enforce safety rules, recording of staff interventions on beamlines and administrative tools including the generation of statistics.

The SMIS' Architecture

The SMIS technical architecture is heterogeneous because of its long development history. Figure 2 sketches the high-level architecture. The Oracle database has evolved over time but the data structure has not changed significantly. The different applications are essentially web applications based on Java (Servlet and JSP) technology that access the data via a proprietary middleware server relying on Microsoft Java (J++) using Microsoft Office applications via COM. One rich client application implemented in Visual Basic is still used by the user office to plan the experiments.

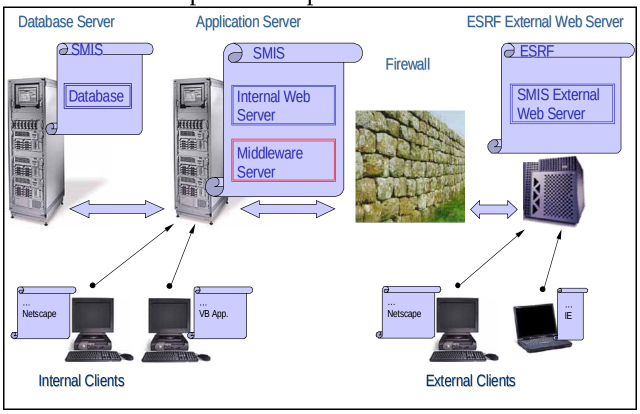

Figure 2: SMIS Architecture.

MIGRATION OF SMIS TO ANKA

The Need for an Online Proposal System at ANKA

The 2.5 GeV synchrotron radiation source ANKA [3] gave its first provision of beam time for the international scientific community via peer review in 2003.

After a year of manual proposal processing there was a demand by the users and the beamline scientists to develop an online proposal system. The demands of ANKA are:

- Processing of 2 calls per year
- Handling of up to 100 proposals per call
- Statistical evaluation of the proposal in terms of internal FZK or external proposers, funding and countries of the proposers
- Grading of the proposals by a single review committee
- Allocation and scheduling of the beam time
- Plan for the arrival of the users

Because the ESRF's SMIS appeared to closely match ANKA's needs, it was decided in November 2003 to adapt the ESRF scientific information system for ANKA.

Issues during porting

The SMIS was developed specifically for the ESRF and had evolved during a period of more than 10 years. It was therefore intricately coupled to the ESRF infrastructure, as it was never intended for use within a differing infrastructure. A lot of ESRF specificities like web page URLs, email-support addresses and server names were hard coded into the source code. It took two man months to find all references and to put them into external property files.

Additionally the appearance of the SMIS interfaces needed to be adapted to the web layout of the FZK. To avoid a total re-development of SMIS it was decided to replace only the graphics in the header of the web forms.

At first glance the needs of the ESRF and ANKA appeared to be perfectly matched. But a more detailed investigation quickly showed that the SMIS offers a lot of functionality, e.g. special detectors, which is irrelevant for ANKA. To solve this problem the unnecessary web form fields were simply deactivated. This could be done without reformatting the complete web forms because of the section-oriented style of the forms.

SMIS@ANKA was installed on an Oracle 9.1 database although it is running under Oracle 8.1 at the ESRF. Upgrading to the newer database version could be done smoothly by exchanging the windows oracle drivers. It was more difficult to configure the database, with its nearly 20 years of evolution, for ANKA since the tables were poorly documented and were not linked at the database level by foreign keys. This approach avoids immediate database errors due to inconsistencies but on the other hand errors in the configuration can only be detected at the application level at runtime. It took several months to find the last bugs in the database configuration for ANKA and to write an up-to-date documentation for the database. Last but not least the data of previous ANKA proposals in paper form had to be entered manually into the system to allow new proposals to refer to former ones.

Actual Status of SMIS at ANKA

The first call for proposal with SMIS@ANKA finished on the 5th January 2005. Only minor issues occurred during the call and the system was quite stable. It took the user office only a few hours to learn the operation of SMIS. Only a few complaints about SMIS have been expressed. In total the porting of the SMIS to ANKA required about 14 man-months for software development at ANKA and 2 man-months support by the ESRF. Regarding these figures ANKA's biggest benefit did not come from the savings in pure development time but rather from the adoption of the SMIS workflow, which has been optimised, with 10 years of experience.

A STANDARD PROPOSAL AND EXPERIMENT MANAGEMENT SYSTEM

Development Collaboration

Building, maintaining and enhancing a proposal and experiment management system like the SMIS is time and resource consuming. As shown by the ANKA-SMIS collaboration there is an increasing need for online proposal submissions and electronic management of experiments at synchrotrons. These interfaces are also being perceived as the standard by their users. The adaptation and installation of specifically developed applications like the SMIS on new synchrotron sites is one possible solution. Unfortunately this leads to two different versions of the same software making it difficult to evolve the software on both sites synchronously. An alternative could be to define and develop a new standard proposal and experiment management system in a combined effort by different synchrotron facilities. A similar approach has been proposed [4] based on the collaboration between research institutes and specialised private partners that take the ownership of the resulting product. The advantages of such an effort are obvious:

- Faster development cycles due to more developers working on the same software
- A complete and well settled set of requirements due to the input of several institutes
- A common look and feel for users working on different synchrotron sites
- Higher quality software due to tests on several sites
- A commonly defined and re-fined architecture eases evolution and re-factoring in the future
- Leveraging state of the art technologies

Of course development collaborations also bear risks and disadvantages:

- Necessity to find a common denominator for all collaborators without loosing the possibility to customise an institute’s installation
- Re-development of existing applications
- Migration of existing data
- Development time schedule depends not only on own priorities but also on the collaborators which may result in a longer response to user requests on a specific site

Having experienced the adaptation and installation of a very specific and well evolved software like the SMIS our opinion favours the combined re-development of a standardised application. We suppose this would result in a stable application that can safely evolve in the future on many different synchrotron sites thus paying back the investment needed up-front.

Organisational Matters

To avoid conflicts between the collaboration partners they should agree early on a general development road map, the organisation of the common development and necessary processes like requirements engineering. Existing open source projects can teach us best practices that have proven to be efficient in a distributed development environment. The collaboration between the ESRF and the French synchrotron SOLEIL regarding the development of the machine control system TANGO can serve as an example. Best practices to take into account include using a common code and document repository, setting up a test infrastructure for automated tests, release procedures that enforce code reviews by collaboration partners etc.

Architecture

The hardest technical part of a common proposal and experiment management system is the definition of a core data model together with core functionality that is valid for all collaborators while still enabling simple extension mechanisms to customise the different installations.

Similar problems are faced by enterprise resource planning (ERP) systems for small and medium enterprises like Compiere [5]. These products are delivered out of the box, have to be customised easily on the client side while still allowing later upgrades of the core system. Inspired by ERPs we can sketch a possible architecture for a standard proposal and experiment management system.

The typical customisable application uses standard components for the core system with well-defined external interfaces, a well-defined core data structure and separate name spaces for core and customised functionality. A workflow engine controls the processes in the system.

Adding fields to the existing data structure and including them in the user interface (either automatically or through configuration) customises the system. Customised workflow actions can access the additional data. For more complex tasks new components can be added that use the standard interfaces of the core system. Integration with other information systems is possible through remote access mechanisms like web services.

A core set of use cases could be the creation (submission) of an experiment’s proposal, the review (grading) of a proposal, the planning of an experiment and its sessions resulting from an accepted proposal and the reporting on an experiment.

Figure 3 presents a very basic core data model needed to implement these use cases.

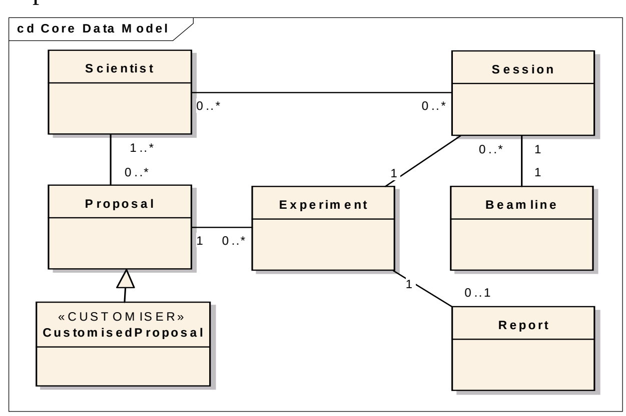

Figure 3: Core Data Model with Proposal Customisation.

The association of a specialised class, bringing its own specific data and database mapping, should allow a customisation of the system (see the example of class *CustomisedProposal* associated to *Proposal* in Figure 3). The customised classes contain specific data and map to customise tables in the database. The relation between core and specialised classes could be either hierarchical (inheritance) or aggregation both allowing easy upgrades of the core model.

A possible technology platform to implement this architecture is J2EE (Java 2 Enterprise Edition [6]) in combination with application servers like JBoss [7], workflow engines like jBPM [8] and relational database (e.g. Oracle [9]).
Analysis and design could make extensive use of existing best practices as found in [10] and [11].

CONCLUSION

Based on the experience of developing site-specific software and porting it afterwards to other sites we suggest a combined development of a standard system up-front. Thus creating a stable core version together with customised installations at different sites and a common developer and user community. Potential collaborators are invited to contact us (e.g. mis-group@esrf.fr). Based on the feedback we will organise a get-together of the different parties.

REFERENCES

[1] http://www.esrf.fr/
[2] http://www.ill.fr/
[3] http://www.fzk.de/anka/
[4] K. Zagar et al., “Co-Sourcing Development of Accelerator Controls”, EPAC 2004, Lucerne, Switzerland, July 2004
[5] http://www.compiere.org/
[6] http://java.sun.com/j2ee/
[7] http://www.jboss.org/products/jbossas/
[8] http://www.jbpm.org/
[9] http://www.oracle.com/database/
[10] M. Fowler, “Analysis Patterns: Reusable Object Models”, Addison-Wesley, USA, 1997
[11] E. Gamma et al., “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, USA, 1995