Laying the Foundation for Web Services over Legacy Systems

Janet Lavery and Cornelia Boldyreff
Dept. of Computer Science
University of Durham
Science Laboratories,
Durham, DH1 3LE, U.K.
Janet.Lavery@durham.ac.uk
cornelia.boldyreff@durham.ac.uk

Bin Ling and Colin Allison
School of Computer Science
University of St. Andrews
St. Andrews
Fife, KY16 9AJ, Scotland
lingbin@dcs.st-and.ac.uk
colin@dcs.st-and.ac.uk

Abstract

As the use of the World Wide Web becomes more pervasive within our society, businesses and institutions are required to migrate a wide range of services to the web. Difficulties arise where there are requirements to integrate existing systems within the migrated and often-extended web based services. The Institutionally Secure Integrated Data Environment (INSIDE) project is addressing the problems and issues surrounding the development and delivery of web based services for "joined up systems" for institutions within Higher Education (HE). The project is working with a variety of existing information systems, e.g. student record systems and managed learning environments, at two universities. To better understand the requirements for an integrated web based service, a common business process, the registration of new undergraduate students, has been analysed and modelled at both sites. Progressing from initial informal models to more formal models in a systematic way, following a meta-process incorporating good practice from domain analysis and requirements engineering has allowed the project to lay the foundation for its development of web-based services.

1 Background

Within Higher Education (HE) there are concerns with regard to the gap between what university central services has traditionally provided and what the academic and administrative communities within the universities currently require. Members of the administrative staff and academic community (staff and students) of HE Institutions (HEIs) are finding the performance of routine tasks difficult due to the nature of their institution's existing systems. These systems, usually comprised of multiple unconnected data repositories, require a user to expend extensive effort to accomplish what should be simple tasks. Users are often prevented from carrying out work by inappropriate access control mechanisms and the lack of appropriate client software. Additional difficulties occur as a result of the numerous *ad hoc* record systems developed at the departmental level that replicate work being done centrally, but that are not co-ordinated with central services. The systems in place at HEIs with their dated software, distributed data, and entrenched business processes, are fully functioning systems needed to support the ongoing business of the institutions whose continued existence is an economic necessity. However, their inability to fully support the changing business environment and the needs of their user communities consigns their systems to the category of systems known as legacy systems. Legacy systems are understood to have both bad and good sides [1]. The bad side is that the system no longer fulfils the needs of the institution and its members. The good side is that it contains those elements essential to the continuation of the institution as a business including the business process and rules, and the existing information base. Continuous change or evolution of large institutional systems requires the incremental capture and modelling of the valuable elements of the system and the subsequent use of both the elements and the models as the foundation for any approaching and subsequent evolutions of the system.

The Institutionally Secure Integrated Data Environment (INSIDE) project is a collaborative project between the Universities of St Andrews and Durham that has been addressing the issues surrounding the development and delivery of joined up systems for institutions. The remit of the project requires that we work with the existing information bases and explore the incremental evolution of the existing systems by building value-added services upon foundations derived from analysing and modelling the existing legacy systems. In addition, we have sought to identify the issues and solve the problems at a high enough level of abstraction to give sufficiently generic solutions applicable to other HEIs. For example, one of the problems identified with legacy systems is that users are trying to carry out work using inappropriate access control mechanisms and without appropriate client software; part of INSIDE's generic solution involves the use of Lightweight Directory Access Protocol (LDAP) and web portals. A pragmatic goal for the INSIDE project is to exploit web technology in the development of pilots for user-oriented institutional portals for end-users, both staff and students, to access existing services via the web.

The majority of HEIs have a "web presence" that is to say they have a web site that provides access to some of the available institutional information and resources. It is thought that the further exploitation of web technologies can aid HEIs in their efforts to overcome some of the problems associated with their legacy systems without destroying the valuable elements of their existing systems. It is believed that web technologies can be exploited to give a consistent, customised, and secure access to the information and resources of the institution that is currently awkward to access while having a positive impact on the business process of the institution [2]. The addition and introduction of value-added services, such as web-based access to the existing module enrolment process, requires a review and mapping of the institutional business processes both prior to and after the introduction of new value-added service. The analysis and subsequent evolution of an institution's system is based on the "analysis, model, and improve" approach usually adopted with incremental development. This allows incremental development of the foundation i.e. domain knowledge capture and modelling to be done in tandem with the evolution of the institution's systems.

The INSIDE project's exploration of this approach is described and defined in a meta-process that is based on the work products generated and evolved during analysis and development. The business process initially captured and modelled that provides the foundation for the introduction of the first web-based value-added service is new student registration and module enrolment. This process has been chosen because it is practised, in some form, in all UK HEIs and because the student data it involves is shared with a variety of systems in both academic and non-academic departments. Our experience of modelling of what was at first believed to be a simple and common process has provided some insights into the analysis and modelling needed to lay the foundations for web services over legacy systems. These have led to the formulation a meta-process presented here in Section 2.

Through the layered understanding of the domain obtained via informal and formal analysis and modelling, it is possible to achieve incremental domain model development upon which iterative evolution of large and complex systems can be based. In section 2, the meta-process applied in the INSIDE project is discussed in detail. In Section 3, the resulting models developed by applying the meta-process in the analysis and modelling of a business process are outlined. In Section 4, we describe how we have used the resultant models to identify areas of the systems ripe for improvement and the use of web-based portal technology to implement some pilot improvements. Finally, in Section 5, we identify open issues and further work for the INSIDE project.

2 The INSIDE Meta-Process

The INSIDE project is addressing the issues surrounding the development and delivery of joined up system for institutions constrained by the necessity of working with an existing information base and legacy systems. An "analyse, model, improve" or incremental development approach has been adopted in building the web-based value-added services accessible through portals. This approach, supported by the decision to breakdown analysis and modelling of the domain into manageable business process segments, allows the capture of domain knowledge about the section of the institution being analysed and places it in context with the evolving model of the institution. The meta-process depicted in Figure 1 below contains several dependant steps that are predominantly iterative, the exception is Initial Analysis, which is performed only once but provides the foundation for the work products on which the subsequent work products are established.

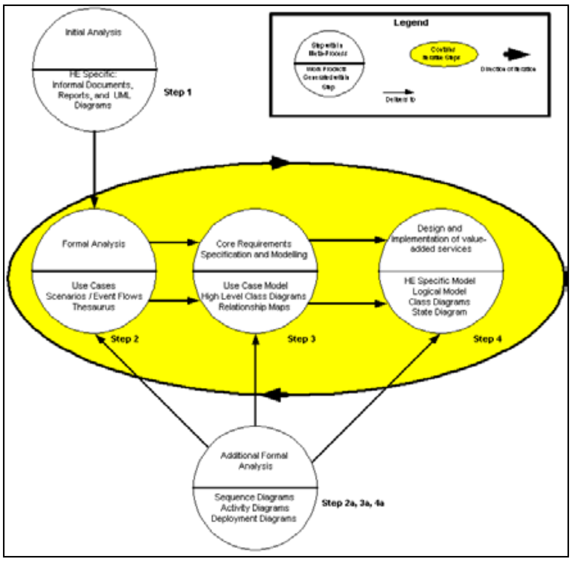


Figure 1 Meta-Process for Incremental Development and Modelling Evolution

2.1 Initial Analysis

Initial Analysis is the first pass over a specific business process area of an institution where the informal analysis and capture of domain knowledge occurs. Domain knowledge is gleaned from domain experts during informal interviews, and modelled by basic block diagrams. This type of diagram consists primarily of analyst-defined rectangles, ovals, and arrows. It is used in conjunction with textual or verbal narrative provided by the analysts for the following purposes:

- demonstration of the analyst's increasing understanding about a section of the domain;
- generation of discussion amongst domain experts; and
- validation of an increasing correctness and completeness in the analyst's understanding of the domain as a whole.

The basic block diagrams, with and without textual accompaniment, the records of the informal interviews with domain experts, and any details gleaned from existing domain specific documentation, are developed separately and not interleaved except perhaps in the analyst's narrative. However, as the analyst's knowledge about the domain increases, it is possible to bring cohesion to the developing model of the institution [3].

It is necessary to begin the rationalisation of the vocabulary during Initial Analysis. Differences in vocabulary between system users with different roles within the institution can cause misunderstanding during Initial Analysis that could impact on subsequent analysis. Vocabulary problems such as the same term having one or more meanings within the HE Institution, and different terms with the same meaning can be problematic if not resolved. During Initial Analysis the development of a simple dictionary of terminology containing an agreed list of appropriate phrases, each with a single meaning may be all that is needed. The content of the dictionary grows with the analysts' understanding of the domain.

Just as Kruchten [4] believes that no single diagram could accurately depict the complexities of a system, we believe that it takes the continual evolution of several work products, some of which may contain many diagrams, to model a complex system. Additionally, we believe that a model of an institution's system must evolve in conjunction with the evolution of the system itself. These beliefs have led us from Initial Analysis with its loosely coupled diagrams to the iteratively executed and mutually dependent steps of the meta-process.

2.2 The Mutually Dependent and Iterative Steps of the Meta-Process

The remaining steps in the meta-process are mutually dependent and performed iteratively. The captured domain knowledge is formally modelled and integrated into the evolving model of the institution. The understanding captured in the institutional model is exploited in the incremental system evolution including the development of value-added services based on existing systems. It begins with Formal Analysis where the captured of domain knowledge is exploited in the generation and modelling of the core requirements. The specification and modelling of requirements in turn provides the foundation knowledge necessary for the subsequent design and implementation of the value-added services. The improvements and changes to the business process resulting from system evolution and implementation of the value-added services are then analysed and captured to provide a model of the current state of the institution's systems. The model representing the current state of the institution's systems is then examined for opportunities to provide more value-added services.

2.2.1 Formal Analysis

Formal Analysis uses as its foundation the domain knowledge captured in Initial Analysis. However, it does not support the evolution of the work products developed in Initial Analysis. Those work products are fixed and once the development of the Formal Analysis work products has begun, Initial Analysis work products are abandoned. As can be seen in Figure 1 of the Meta-Process, phases of the process are closely coupled with the work products developed. The INSIDE project is using the Unified Modelling Language (UML) to support the analysis and modelling done in the iterative phases of the meta-process. UML is an object-oriented notation that is widely supported and in use at both Durham and St. Andrews. UML has been used to record each department's specific knowledge, to facilitate understanding between the analysts at the two university departments when the domain knowledge has been compared, and to support the modelling of the generic elements of the domain.

Further support for communication and interoperability is provided by the development of a thesaurus. A thesaurus is a collection of generic terms used to represent the concepts of a specific domain and is organised so that the relationships between the terms are made explicit [5]. The thesaurus is initially developed using the dictionary as its foundation. The thesaurus allows for the use of alternate definitions for a term and may also be used to provide a mechanism for defining correlations between terms in common use within specific areas of the domain and terms more commonly used throughout the whole domain.

The UML model consisting of a collection of conceptually related UML diagrams is divided into two distinct but complementary sections: the Use Case View and the Logical View. The Use Case View is the section designated for high level models of the business processes and is considered a core model element that supports domain analysis, requirements gathering [6] and exploration of alternative business process [7]. The Use Case View contains primarily Use Cases and accompanying scenarios. Detailed scenarios are used in conjunction with use case diagrams to capture the complexities of the business process of the domain hidden by the actual use case diagram [8]. Event Flows as defined by Quatrani [6] can be used to record the scenarios associated with each use case.

There are several alternatives to Event Flow statements. One is the Use Case Descriptions described by Cheesman and Daniels [9] that are quite similar. However, the Use Case Description has no specific area to identify the pre-conditions that must be met before the scenario can be enacted. Neither the Event Flow scenario description nor the Use Case Description actually handles parallelism within or between use cases [9]. Sequence diagrams are a more appropriate means of dealing with parallelism within and between use cases. Sequence diagrams can also be used demonstrate the interactions that occur within a scenario or use case. However, sequence diagrams are more typically associated with message exchange between objects in the Logical View of the model [6]. If concurrency were a major issue in the business process being modelled then it would be appropriate to use sequence diagrams [10] to supplement the modelling.

The Logical View is used to support those functions that are more strongly coupled to more specific institutional or departmental domains such as the low level design models closely tied to implementation [6].

2.2.2 Core Requirements

Formal Analysis provides a model of the domain knowledge that is used in the generation and modelling of the core requirements. Initially these will consist primarily of the use cases and scenarios describing the current state of the generic organisation generated in Formal Analysis. Subsequent requirements gathering and elicitation will produce additional use cases generated to explore and identify proposed value-added services, such as web access to legacy data stores. Specifying requirements necessitates a more detailed view of the organisation than the one needed in analysis. As a consequence high-level class diagrams concerning domain elements need to be developed. These class diagrams will model elements close to the domain and should be directly traceable to those diagrams closer to implementation [9, 6]. Once these high-level class diagrams are being generated, it is advantageous to map their modelled relationships to the corresponding relationships that exist between actual domain objects.

The thesaurus is expanded to support the detailed needs of core requirements to include an object-oriented classification applied to each generic term [11]. This is done primarily to assist with defining the requirements but also with a view to supporting the eventual design and implementation of value-added services and to provide traceability of the terms throughout the development process. This approach follows that found in Protégé 2000, which supports the construction of domain specific ontologies [12]. During the high level modelling of the generic business process contained in the Use Case View the object-oriented classification is usually super classes, classes and a few objects. As the model of the business process grows from the Use Cases View to the Logical View, the classifications become more detailed and elements such as objects, attributes and operations are identified.

2.2.3 Design and Implementation of Value-added Services

The value-added service identified, specified, and modelled in the previous step are then developed in the Design and Implementation step of the evolutionary iterative development cycle represented by the Meta-Process. Design and Implementation needs to exploit the more domain specific elements of a UML Model that are contained in the Logical View of UML Model. The Logical View is used to support those functions that are more strongly coupled to a specific institutional or departmental domains such as the low level design models closely tied to implementation [6]. The Logical View is generally comprised of domain specific use cases with accompanying scenarios, and class diagrams that are less abstract and close to the actual implementation are built. The Logical View section of the model is less abstract and of less use outside the institutional or individual departmental with the institution; however, there is domain knowledge captured in the implementation of new value-added services that must be passed into the next iteration of the cycle.

The improvements and changes to the business process resulting from the implementation of the value-added services are then analysed and captured to provide a model of the current state of the institution's system. This new domain knowledge is part of the foundation for evolution of the formal analysis work products and the model representing the new state of the institution's systems is then examined for opportunities to provide more value-added services. It is possible that as time passes the Use Case View that is used in Formal Analysis can become more tightly coupled to a specific HE Institution's domain. As the INSIDE project is using the Use Case view as the common model for both institutions and is attempting to keep it generic enough for use by any UK HEIs, the design will remain at a higher level of genericity.

2.2.4 Additional Formal Analysis

Additional Formal Analysis is performed to support analysis with a particular focus. It may be performed at any point in the cycle and requires the use of UML notation appropriate to the focus. As stated before sequence diagrams are generated to explore concurrent processes, an activity more suited to but not restricted to specifying or modelling core requirements. Activity diagrams support a focus on the systems actors by showing the consequences of their interactions with the system in the context of specific business processes. They are useful when very detailed examination of user activities are necessary, for example, in low-level function analysis and design. Deployment diagrams support the abstracting unnecessary detail from complex distributed systems and are useful in Formal Analysis. It is the motivation behind the activity that decides the selection of the supporting modelling notation and the appropriate time to use it.

3 INSIDE's Analysis and Modelling Experience

The business process selected for initial analysis is the registration of new undergraduate students. The intention of this process is to register students intending to meet a specific academic target such as gaining a Bachelor of Science degree with the HEI. In the UK, this process begins the same of all HEIs. All the HE student records for the new academic year entry cohort are distributed from a central "clearinghouse", the Universities and Colleges Admissions Service. These student records have an identical structure and content. They are held until just prior to the beginning of an HEI's academic year when they are distributed to the appropriate central registration service (admissions department) of the various HEIs. Each HEI's central registration services then ensures that all the relevant student data has been captured and where necessary shared with the academic and non-academic departments involved in the registration process. Once in the custody of an HEI the student record is manipulated to suit the needs of that institution.

3.1 Results of Initial Analysis

The initial analysis of each of the HEI's' undergraduate registration process was conducted separately by analysts at each institution. The general focus was on central services registration department and the analysts own academic department. Analysts at both institutions based their initial domain analysis process on informal interviews with members of staff (both academic and administrative) with direct responsibilities relating to the undergraduate registration. Any available HE Institution documents, unique to the undergraduate registration process, were also used. The knowledge gained from the informal interviews, and to a lesser extent the existing documentation, was recorded in basic block diagrams.

Figure 2 illustrates a simple block diagram resulting from the analysis at Durham. This diagram gives an annotated overview of the undergraduate registration process within Department of Computer Science at Durham.

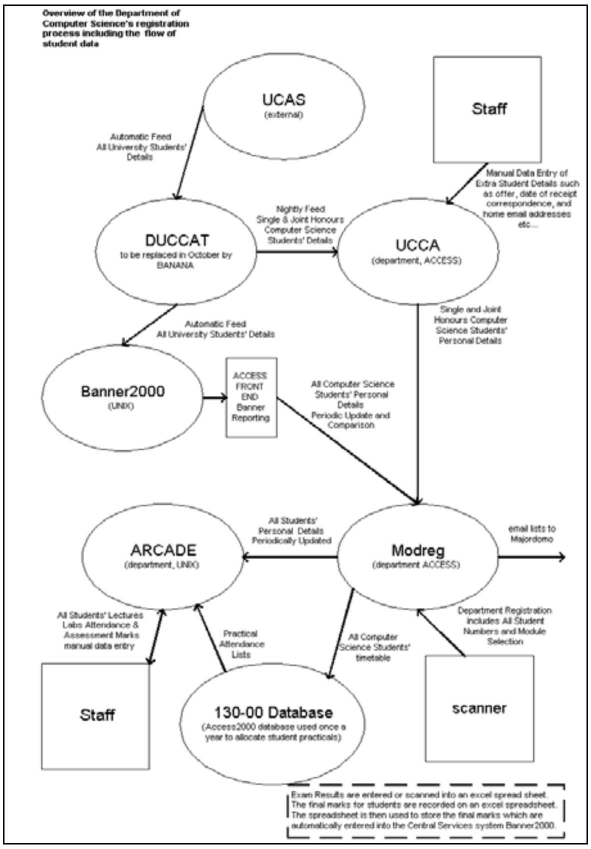


Figure 2 Sample Block Diagram Created at Durham

Analysts in the School of Computer Science at St. Andrews also examined their own undergraduate registration process from both the institutions' and the school's perspective. Figure 3 below shows a use case diagram of the original registration process at St. Andrews.

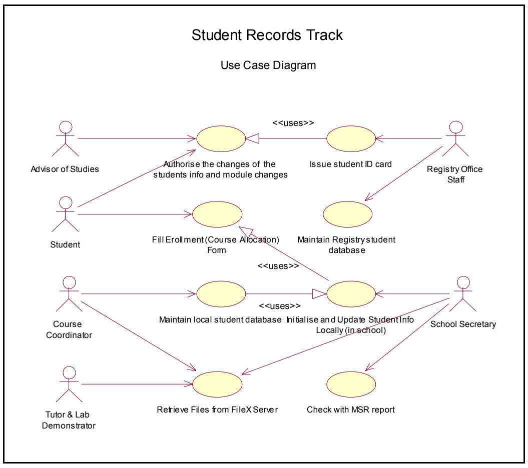


Figure 3 Use Case Diagram Created at St. Andrews

The initial domain analysis conducted separately at each of the HEIs provided the base work products and knowledge capture needed prior to the formal domain analysis. The standalone one-use-only work products developed early in the analysis such as block diagrams have been used to develop two final interrelated work products: the UML diagrams and the dictionary. These two work products are related in that the terminology used in the UML diagrams must be defined in the dictionary. The formal analysis based on these is described in more detail in the next section.

The work products, both the diagrams and the dictionary, generated during Initial Analysis of a section of the domain are fixed at the end of the phase. Combined they provide a snapshot or description of the business process as it exists as at the end of initial analysis. The work products developed at Durham and St. Andrews provide a snapshot of the domain knowledge specific to each individual department at specific HE Institution. As it is the intention of the INSIDE project to find generic solutions to problems that can be applied to many HEIs it was necessary to share and compare the captured domain knowledge about the student registration process. The intention has been to identify the commonalities and discrepancies between the two institution's departments to derive a Generic Model representing student registration process at any UK HE Institution. This work was done as part of Formal Analysis discussed in the next section.

3.2 Results from Formal Analysis

The formal analysis and modelling of the HE Institution's registration process has resulted in the development of an evolving Generic Model of HE Institution's business processes [11]. This model is comprised of a group of mutually dependent work products that are incrementally developed and evolved in tandem with domain knowledge understanding and the evolution of the institution's existing system. The model at present consists of a UML model developed using Rational Rose 2000™ that employs primarily use cases with accompanying scenarios, and a Thesaurus of generic terms. Combined they provide a Generic Model of the current undergraduate registration process. Figure 4 depicts an example Use Case, "Complete Registration with Academic Unit", held in the Generic Model.

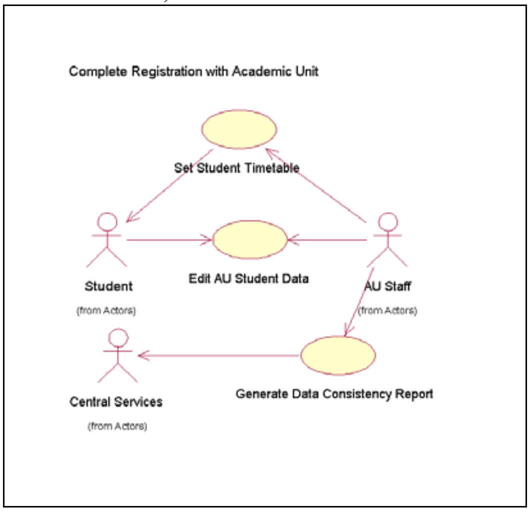
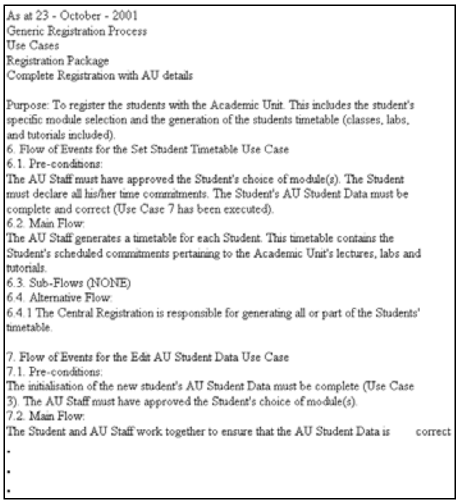


Figure 4 Sample Use case Diagram from Generic Registration Model

An excerpt from the Event Flows associated with the Use Case depicted in Figure 4 is contained in Figure 5 below.



As at 23 - October - 2001
Generic Registration Process
Use Cases
Registration Package
Complete Registration with AU details

Purpose: To register the students with the Academic Unit. This includes the student's specific module selection and the generation of the students timetable (classes, labs, and tutorials included).
6. Flow of Events for the Set Student Timetable Use Case
6.1. Pre-conditions:
The AU Staff must have approved the Student's choice of module(s). The Student must declare all his/her time commitments. The Student's AU Student Data must be complete and correct (Use Case 7 has been executed).
6.2. Main Flow:
The AU Staff generates a timetable for each Student. This timetable contains the Student's scheduled commitments pertaining to the Academic Unit's lectures, labs and tutorials.
6.3. Sub-Flows (NONE)
6.4. Alternative Flow:
6.4.1 The Central Registration is responsible for generating all or part of the Students' timetable.

7. Flow of Events for the Edit AU Student Data Use Case
7.1. Pre-conditions:
The initialisation of the new student's AU Student Data must be complete (Use Case 3). The AU Staff must have approved the Student's choice of module(s).
7.2. Main Flow:
The Student and AU Staff work together to ensure that the AU Student Data is correct
.
.
.

Figure 5 Excerpt from an Event Flow in the Generic Registration Model

The Thesaurus developed as part of the Generic Model for the INSIDE project is used to record domain knowledge to support analysts and designers using the UML modelling tools. The Thesaurus is used to promote understanding of terms and provide terms for use in the

UML diagrams that will be understood by all stakeholders in the system's current evolution.

The addition of the object-oriented classification area in the thesaurus required the clarification of the correlation between real world objects' relationships and object-oriented relationship implied by the object-oriented classification. The method that has proved sufficient is to expand the Generic Model to include linked hierarchical and class diagrams defining the relationships. There are several diagrams depicting the main relationships required to achieve clarity. The illustration in Figure 6 shows an example of a typical mapping.

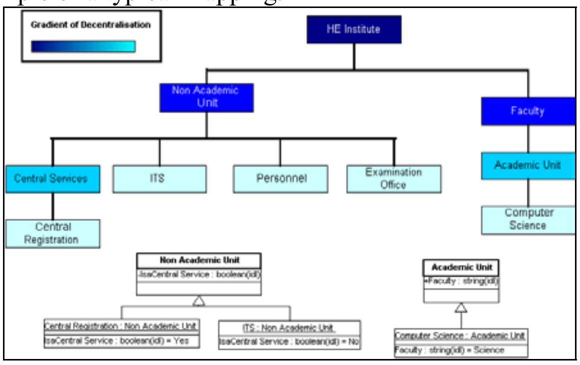


Figure 6 Mapping the Relationships between People in the Domain

4 Using the Generic Model

The Generic Model, comprised of a group of work products that provide analysis of the registration process, has been used to identify areas of the system, supporting the registration process and related process such as the management of student records, that are ripe for improvement. Some of the student-oriented areas of the domain requiring improvement are:

- Specific module enrolment,
- Academic feedback from staff,
- Submission of academic work,
- Tracking of assignment due dates, and
- Students' access to their own records.

Some of the staff-oriented areas of the domain requiring improvement are:

- Resource planning and allocation of timeslots for tutorial and practicals including communicating the allocated times to the appropriate staff and students,
- Tracking of assessed work's due data and marked by date,
- Module administration,
- Access to individual Student academic details,
- Cohort analysis, and
- Access to role relevant information and resources.

The case study for this paper presented in Section 4.1 describes the solution that has been applied to two areas ripe for improvement: student enrolment on specific course modules and communicating to the relevant staff and students the allocated timeslots for student tutorials and lab practicals. As one of the problems identified with legacy systems was that users were trying to carry out work using inappropriate access control mechanisms, and as the Internet is in common use by the staff and students at most HEIs, the decision to deploy a web-based front-end access to crucial legacy databases, including student records administration and department (school) information was obvious.

4.1 Case Study - Developing User-Centric Institutional Portals

Gleason [2] states that an institutional web site can be viewed three different ways: hierarchical, audience pages, and personal portals. A hierarchical view is the public view of the institution's web site that provides a structure used by visitors and casual users. Entry to the site is most often at the institution's home page and a pathway to the information is provided by the user's navigation through the site. For example, a casual visitor looking for information about a specific academic department may use the following path: from the institution's home page to a page listing all the academic departments, and then to a specific academic department's home page. Audience pages give specific large groups a view of a subset of related web pages contained in the web site that is particular to their requirements. For example, conference attendees will be interested in the location of specific buildings in the institution and the institution's conference facilities. Both these views, hierarchical and audience, are public views of an institution's information and resources. The third view identified by Gleason [2] is the institutional portal view. This view provides a view of the institution's web site that is restricted to members of the institution and is a primary concern the Institutionally Secure Integrated Data Environment project.

The institutional portal view provides a user with a single point of contact to a range of resources and information specific to the user's requirements and not generally available to wider audiences [2]. For example, a student portal could provide a view to the student's own personal details. Developing portal views is technically challenging in that a portal must provide consistent, customisable, secure access to the information and resources that the institution wants to make available to its various user communities. For example, lecturers need to be able to add student marks to a system, but it would not be appropriate to allow students the same privilege. Institutional portals must restrict the view of an institution's information and resources to that which is appropriate for the user based on his/her roles and responsibilities within an institution [13].

The intention behind a portal is to provide a single point of contact to a range of services appropriate for a diverse and international end-user community. The pilot scheme is based on the process of registering new undergraduates to particular modules. An agreed model of distributed functionality between the department and central services associated with INSIDE was necessary before a relatively comprehensive portal could be implemented, in view of the fact that both central services and the departments hold information that can be aggregated to form such a portal. Therefore, both needed to agree on their respective responsibilities before broadening the integration to accommodate students.

The user-centric institutional portals piloted for INSIDE initially have been developed as part of the investigation of migration to the web of tasks associated with student registration on specific modules. Figure 7 below illustrates how in carrying out the more detailed design of these portals, the associated diagrams once more become institution specific.

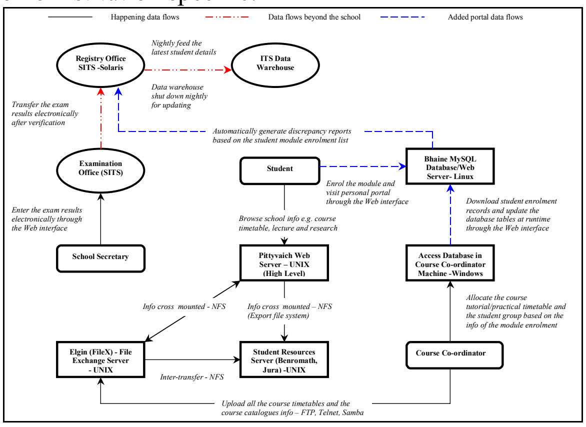


Figure 7 Data Flows within the School of Computer Science at St. Andrews

There are two distinct portals, the student portal and the staff portal, based on the institution members' roles and responsibilities in the context of the student registration process. The student portal provides the means for students to register on any available modules, restricted only by their having achieved the prerequisites. This effectively lets the student enrol from anywhere, anytime and with any web browser. The student portal also provides the means for students to view their own student records and identify any problems with the data. They may not access the data directly to correct it, but once aware of any discrepancies they can take the appropriate steps, in this case the generation of an email to central registration to start the data correction process. The student portal also provides a student's own tutorial and practical lab time allocations, which can be seen in Figure 8 below.

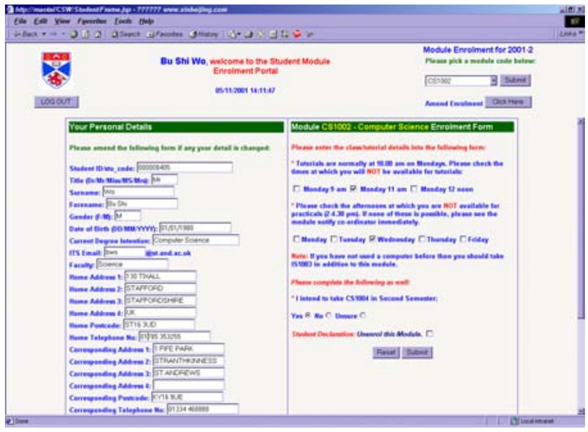


Figure 8 Student Portal Access to On line Module Enrolment

The staff portal provides access to data and resources needed in the performance of tasks related to the enrolment of students in a module specific to a department or school. The staff portal will allow staff to enrol students when necessary. For example, when students have had to seek advice on the appropriateness of a module or when staff choose to waive the module's prerequisites. Access to some resources and data is dependent on staff roles and responsibility within the department. All teaching (or lecturing) staff can view the student's tutorial groupings and practical lab time allocations. However, only a restricted number of staff have access to the resources and data necessary for the allocation of students to tutorial groups or practical laboratory time slots.

The migration of student module enrolment onto the web has provided the means to develop tighter integration between the web-based departmental learning environment, the Tutorial and Group Support (TAGS), used at St Andrews and students' official records. This has helped to eliminate the former need for students to have multiple usernames. More importantly, the tighter integration has provided some initial support for cohort analysis and student centred profiling which is useful in driving the educational process and evolution of the web-based learning environment.

4.1.1 Technical Issues

The pilot system at implemented at St. Andrews has been developed using Java 2 Platform, Enterprise Edition (J2EE) Java™ enabling the utilisation of its many features including: "Write Once, Run Anywhere" portability; JDBC API for database access; and a security model that protects data even in Internet applications. J2EE also fully supports the use of Enterprise JavaBeans (EJB) components, Java Servlets API, Java Server Pages (JSP), JavaMail API, Java Naming and Directory Interface (JNDI) and XML technology, which have been employed in the pilot system implementation

The pilot system utilizes an "N-tiered" approach, where a JSP (or Servlet) acts as a mediator or controller, delegating requests to JSP pages and JavaBeans. In an N-tier application, the server side of the architecture is broken up into multiple tiers.

In this approach, JSPs are used to generate the presentation layer, and either JSPs or Servlets perform process-intensive tasks. The front-end component acts as the controller and is in charge of the request processing and the creation of any beans or objects used by the presentation JSP, as well as deciding, depending on the user's actions, which JSP to forward the request to. There is no processing logic within the presentation JSP itself: it is simply responsible for retrieving any objects or beans that may have been previously created by the Servlet, and extracting the dynamic content for insertion within static templates.

The (INSIDE) pilot system is composed of five tiers, which consist of a client (tier 0), the middle tier - the JSP (tier 1), which interacts with the back end resources - database (tier 3) and the other remote objects (tier4) via Enterprise JavaBeans (EJB) component (tier 2). The EJB server and the EJB provide managed access to resources, support transactions and access to underlying security mechanisms, thus addressing the resource sharing and performance issues of the Five-tier approach, as illustrated in Figure 9 below:

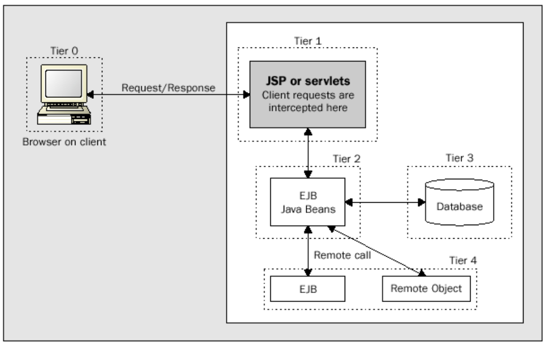


Figure 9 Five-Tier Architecture of Pilot

The system has also been ported for use on the Apache version 1.3.19 web server with Tomcat version 3.2.1 and a MySQL version 3.22 database under Linux Red Hat™ which makes it much more robust and maintainable.

4.1.2 Security and Authentication

As users expect their personal data to be managed securely [14] and the UK's Data Protection Act [15] legislates the support of that expectation in the UK, the gathering and displaying of personal user information in the portal required the definition and implementation of a robust security and privacy policy. The portal provides a secure environment that facilitates user authentication, user authorisation, and private secure communications. In keeping with the existing security measures provided by the institution's IT Services department, the Lightweight Directory Access Protocol (LDAP) servers manage authentication and authorisation where user accounts and user groups are defined and managed. The utilization of LDAP ensures that no resources can be used by an anonymous user, and that logon identification is authenticated. The use of LDAP authentication to authenticate people using the INSIDE system has brought several advantages. The INSIDE project remit to exploit existing systems has been met with the use of LDAP, which was already in use at the institution. Furthermore, users of the portal are required only to remember one set of username and password, the ones provided by the institution. In addition, the administration of (lost, revoked, out of date) usernames and passwords remains the responsibility of IT Services which provides a single point of access to this sort of information.

4.2 Application of the Generic Model after the Introduction of the Portal

The process of analysis and modelling of the student registration, together with the Generic Model and the knowledge gained through the implementation of the pilot portals provided the necessary foundation for the analysis, design and implementation of several more web-based value-added services at the departmental level. These include pilots at both institutions to provide students with web-based access to their specific attendance details, and their specific assessed work deadline dates and provisional marks. Other pilot services provide staff with web-based access to their students' attendance, deadlines, and provisional marks. Web-based electronic submission of assigned work has been introduced. An automated student attendance record pilot using card reader technology and student Matriculation Cards has been introduced at St. Andrews, and a module comparison generator that automatically generates a report identifying discrepancies between centrally held student data and departmental student data has also been developed.

5 Open Issues and Further Work

The next large HEI business process under investigation by INSIDE is the exchange of Student Records. This is to support the development of Student record exchange system that would allow student records to be exchanged between UK HEIs. This will provide the opportunity to progress the Generic Registration Process Model to a more useful Generic Student Information Model. As the exploration of the use of XML will be part of this investigation it is only sensible to expand the

Thesaurus to include the appropriate XML Specification classification. It is obvious that it will be come necessary to locate support tools for the Thesaurus that will allow changes within it to be reflected in other work products. One possibility being investigated is to use the Thesaurus as a foundation for an Ontology. This would support knowledge acquisition, sharing, and reuse by providing a repository for the general and detailed knowledge about specific domains. It could be used to classify the 'things' or objects of the domain [16], as within the context of the domain, objects are identified, classified, and defined by an ontology. This would appear to be a natural expansion of the domain knowledge currently captured in the Thesaurus. The ontology would contain at its core layer a collection of knowledge that is specific to a particular HEI's domain and progress to an outer layer of knowledge that is more general and useful to a broader range of HEIs.

There are several areas targeted for further investigation by the INSIDE project that revolve around student data access and its usefulness as an information resource. These include the access and security issues associated with cohort analysis to support learning and teaching objectives. Privacy issues related to the management of information about Disabled Students as well as accessibility issues in general are being investigated. All of these future investigations will be developed through further incremental modelling following the meta-process and evolving both the foundational models as well as the business processes and underlying supporting systems.

6 Acknowledgements

The work described in this paper is funded by the JISC Committee for Integrated Environments for Learners under the Building Managed Learning Environments in HE (7/99) programme. INSIDE is a collaborative project between the Universities of St Andrew and Durham. The work described here has involved contributions from colleagues at both universities. We are grateful to our colleagues in the project management group at the University of St. Andrews especially Alex Bain, Gail Ritchie, Chris Wimlett, Malcolm Bain, and Alastair Work; and to our colleagues in the Department of Computer Science at the University of Durham, especially Brendan Hodgson and Sarah Drummond.

7 References

1 M. Ramage and M. Munro, "It's Not just about Old Software: A Wider View of Legacy Systems", *Systems Engineering for Business Process Change Collected Papers from the EPSRC Research Programme*, Peter Henderson (Ed.) Springer-Verlag London Great Britain, 2000, pp. 279-290.

2 Bernard W. Gleason, "Institutional Information Portal Key to Web Application Integration", *Portal White Paper*, January 26, 2001. http:\\crayfish.mis.udel.edu /maria/uportal/whitepaper2001.pdf

3 Collette Rolland and Naveen Prakash, "From conceptual modelling to requirements engineering", *Annals of Software Engineering,* 10 (2000) pp. 151-176

4 Philippe B. Kruchten, "The 4 + 1 View Model of Architecture", *IEEE Software*, Vol. 12, No. 6, November 1995.

5 Roy Rada, "Maintaining Thesauri and Meta-thesauri", *International. Classification*, Vol. 17, No. 3.4, 1990.

6 Quatrani, T., Visual Modeling with Rational Rose 2000 and UML, Addison-Wesley, New Jersey, USA, 2000.

7 D.W. Bustard, Z. He, F.G. Wilkie, "Linking soft systems and use-case modelling through scenarios", *Interacting with Computers*, Vol. 13, 2000, pp. 97-110.

8 J.M. Carroll, "Five reasons for scenario-based design", *Interacting with Computers*, Vol. 13, 2000 pp. 43-60.

9 Cheesman, John, and John Daniels, UML Components A Simple Process for Specifying Component-Based Software, Addison-Wesley, New Jersey, USA, 2001

10 Martin Fowler with Kendall Scott, UML distilled : a brief guide to the standard object modeling language, 2nd edition, Addison-Wesley, New Jersey, USA, 2000

11 Janet Lavery, "Report on the Generic Model of the Process of Undergraduate Registration at Higher Education Institutions Version 2.0". An *Institutionally Secure Integrated Data Environment Report*, January 2002 http://www.dcs.st-andrews.ac.uk/inside/report.html

12 N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, & M. A. Musen. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems* 16(2): 60-71, 2001.

13 Colin Allison, Malcolm Bain, et al, "An Institutionally Secure Integrated Data Environment". *JCIEL Managed Learning Environments Circular 7/99 Executive Summary*, February 2000.

14 Steven Carmody, "Shibboleth Working Group Overview and Requirements Document", February 20, 2001. http://middleware.internet2.edu/shibboleth/docs/draft-internet2-shibboleth-requirements-01.html

15 Data Protection Web Site http://www.dataprotection.gov.uk/

16 B. Chandrasekaran, John R. Josephson and V. Richard Benjamins, What Are Ontologies, and Why Do We Need Them?, *IEEE Intelligent Systems*, Vol. 14, No. 1, January/February 1999.