Experience Using a Web-based Knowledge Management System in Support of Postgraduate Part-time Course Logistics

Graham McLeod, Senior Lecturer, Information Systems, University of Cape Town
Managing Partner, Inspired
mcleod@iafrica.com

Abstract

A new, South African-developed, web-based knowledge management product, dubbed "Archi", has been used in support of a Postgraduate Part-time programme at the University of Cape Town during the past two years. This paper illustrates the meta model of the information maintained in the dynamic site, discusses the functionality provided in support of the courses, the experience of students using the site, the challenges involved in evolving the site from one year to the next, security considerations and what has been learned in terms of the strengths and weaknesses of such an application. The site was implemented extremely rapidly and has been able to evolve in response to changing requirements as these arose. Areas for future improvements and research are suggested.

Background

The Academic Context

A new structure was introduced for a part time postgraduate programme in the Information Systems curriculum at the University of Cape Town during 2000. This comprises a postgraduate diploma in distributed commercial information systems, followed by a research-based honours module. The overall program runs over a period of two years, with new students being accepted each year. Students are drawn mainly from practitioners in industry wishing to formalise their qualifications, enrich their skills, or embark on higher level research-based study. In the first year of operation, 24 students were admitted to the diploma. Of these, 18 progressed to honours in the following year, while 32 new students were accepted on the diploma. In 2002, 42 students were offered places on the diploma, while 28 progressed to the honours program.

Diploma students attend classes twice per week, for three hours on each of two evenings. These sessions are in a plenary group one evening and in smaller study groups the alternate evening. They also attend two full time weeks (five full days each), one in February and another in July. Honours students attend a block of research methods instruction on a similar basis, then move to less frequent meetings as they progress their individual research, submit and review interim deliverables, and develop their technical reports (mini theses).

Since most students are employed full time, they spend relatively little time on campus, and most of that during evenings or weekends. This limits the access of students to normal support facilities in the University environment. On the other hand, surveying students, we find that virtually all students do have regular access to the Internet and e-mail. We have thus evolved a style of administration and interaction with students where we rely heavily upon electronic communication. In prior years, this was achieved mainly via e-mail and attachments. An informational web-site provided static information describing the course and its requirements. During 2000, we built an extranet site for the program, which was updated by staff on a weekly basis. This allowed us to capture and disseminate a variety of types of material including:

- Lecture materials
- Readings
- Assignments

- Student contributions (Presentations, Readings)
- Items of social interest to the group, such as planned events and photographs

This site was well received by students and provided valuable support to the administration of the programme, but proved very onerous to update, despite the use of high-level web authoring tools (Net Objects Fusion). We provided students with a copy of this site on CD prior to their November examinations. This was deemed to be very valuable by students. Beginning in 2000, we decided to adopt an available knowledge management tool to facilitate the management of a dynamic web site and knowledge base on a distributed basis.

Tool Background

The tool available was a knowledge management tool developed in industry by the author and colleagues, called Archi. This is a multi-user, web based tool which manages both structured and non-structured data (documents and links to other material). The tool was originally developed to manage knowledge related to the management of enterprise architectures within large organizations. During its development, abstractions reached in the design have allowed it to become much more general purpose and it is now deemed suitable for use in support of a variety of distributed, collaborative, professional knowledge intensive tasks. It had not previously been used in support of academic administration, but it does offer a runtime extensible meta model which allows the customisation and adaptation of the tool to a domain in a short space of time. The tool was at an early stage of production use. Readers interested in the tool architecture should consult [McLeod, 2001].

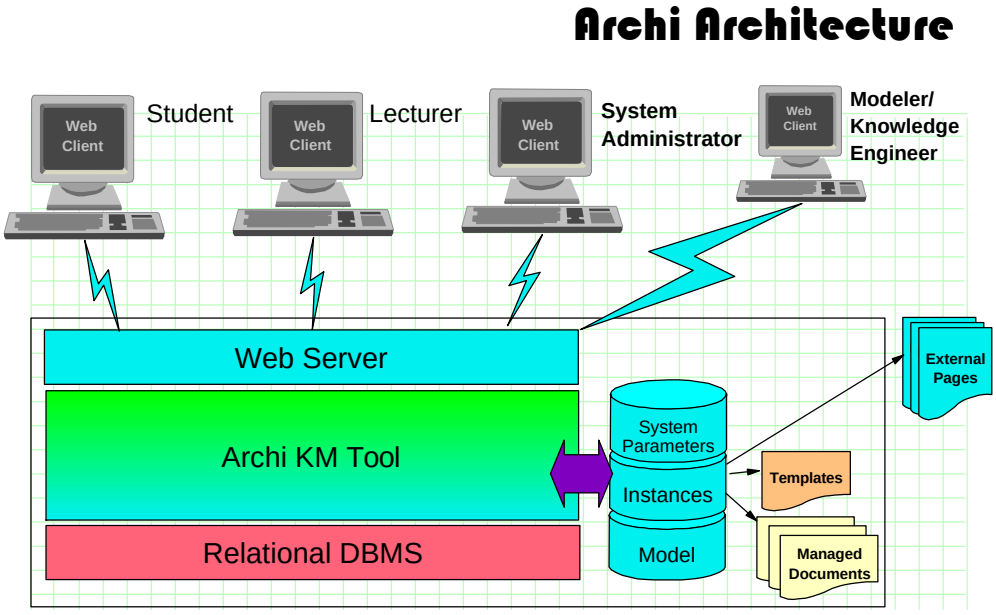

Figure 1 - Tool Architecture

The tool was installed on a web server within the Commerce Faculty in February 2001. This merely required access to the relevant server, copying software, setting paths, and setting up a suitable data set name (DSN). The server already had a database management system (Ms Access), web server (IIS) and operating system (Windows NT 4) installed. The available server proved to be well below expectations in terms of capacity, but we decided to go ahead with the installation anyway, since scheduling access to a production server is a time consuming process and we would have had no other option available before the scheduled start

of the program. The configuration was a 486 Intel machine with a 266 MHz processor and 128 Mb of memory. The Archi tool at that time required approximately 12Mb of memory at runtime. It installed successfully and appeared to perform adequately. We had our doubts about supporting multiple users, but did not really have an option available to us and our initial application was not mission critical.

Implementation

Modeling the Required Knowledge

The first step in customising the environment to support our purposes, was to model the application domain in a form suitable for expression in the tool. The model was developed in a period of two days, and was tested incrementally by creating the necessary types and relationships in the tool meta data and populating the types with suitable instances to verify that the model was suitable. The meta model devised to support the program is reproduced below.

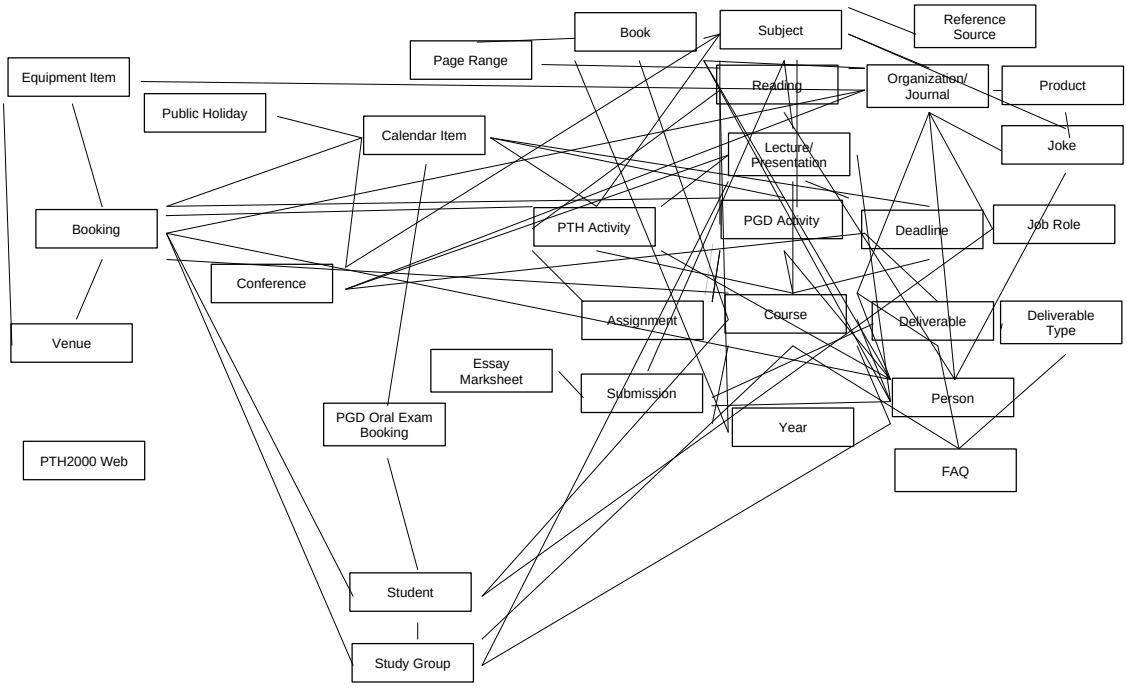

Figure 2 - Meta Model for the Academic Support Application

Not apparent from the diagram, but vital to this application, is the fact that several types support rich attributes in the form of documents (presentations, documents, Acrobat files) or linked web content (URL's) which may be held locally or anywhere on the Internet. Types supporting these capabilities include:

- Book
- Reading
- Reference Source
- Submission
- Conference
- Assignment
- Lecture/Presentation

- Deliverable Type and Deliverable

If the content is locally authored, it is normally uploaded to the server and is managed by the tool, which provides versioning via date and time stamps, keeping old versions while ensuring that the latest is linked for live retrieval. External content is referenced by web links. If this content is moved or removed by owners, these links will become "broken". Such content is considered "unmanaged". Where publicly available information was considered critical to the course, it was captured via Acrobat and held locally to obviate this problem.

Limitations encountered during the modeling included an inability to specify inheritance in the schema. This would have simplified things, especially from the perspective of the designer. It does not materially affect the view from a typical user perspective.

Environmental Setup

The second step was to identify the intended user community and to set up security permissions for their access to the tool and the repository information. We had 24 student users and 8 staff users. A few other users were established for Administration and Guest demonstrations respectively. Users are set up with individual identities and passwords, but are managed at group level with respect to security permissions. Users may belong to several groups. The tool permits control over:

- Access to functional areas within the tool (e.g. Browsers, utilities)
- Access to types within the repository (e.g. Users, Readings)
- Rights to deal with meta data, default values, instance data and control create, view, update, delete, template operations

Security setup is fairly straight forward, but time consuming when fine grained permissions are controlled at the type level. Due to time and resource constraints, we elected to keep things simple initially, controlling access to the tool and functions at a macro level, but allowing generous access to types, and trusting to students' integrity, once admitted.

No client setup was required, as all interfaces are served in standard HTML and Java Script - thus all that is required is a standard version 4 (or better) web browser (Internet Explorer, Netscape, Opera). This allowed students to easily access the site on or off campus.

Initial Content Setup

Content was initially established by the course administrator to provide structure and a seed knowledge base. This included:

- Subjects
- Readings
- Reference Sources
- Calendar, Modules, Course Activities
- Assignments
- Deliverable Types and Deliverable specifications
- Students and Student Groups
- Prescribed Books

Training

Very limited formal training was given, consisting of a half hour demonstration to the group, followed by a 1.5 hour lab session. The initial training session was approached more in the mode of an experiment, given our misgivings for the seriously underpowered server. This

indeed proved to be a problem, with very slow response times and time outs experienced. We found that we could sustain about 6-8 concurrent users, and grouped students accordingly for the latter part of the session. Despite the stress of twenty odd students attempting simultaneous logins, no crashes occured. Guide sheets of instructions were provided for later more complex tasks, such as uploading assignments. Students completed these in their own time.

Functions Supported

The tool immediately supported a variety of tasks. Initial content was largely limited to that provided by the administrator, but quickly grew as students made contributions and became familiar with the tool. Typical tasks supported include:

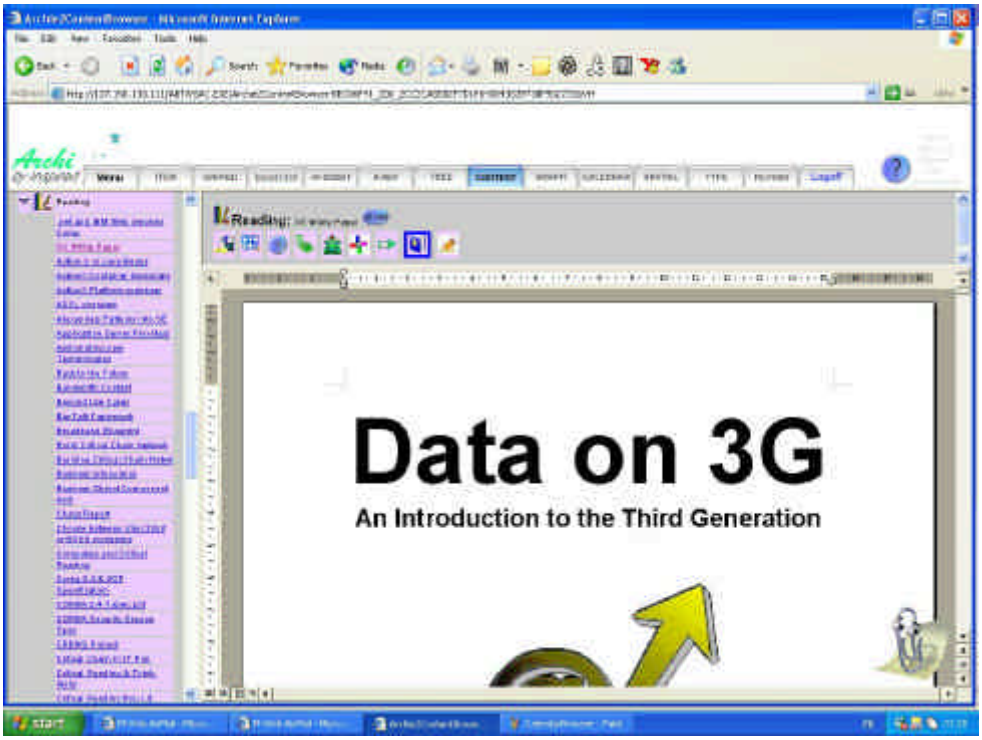

Figure 3 - Reading viewed in the Content Browser

- Reference: Looking up things by course, subject, contributor, study group, organisation, text search etc. Once items are located, their full content is easily displayed using the Content Browser.
- Calendaring: Consulting calendar to find dates, activities on those dates, related readings, assignments, and lectures

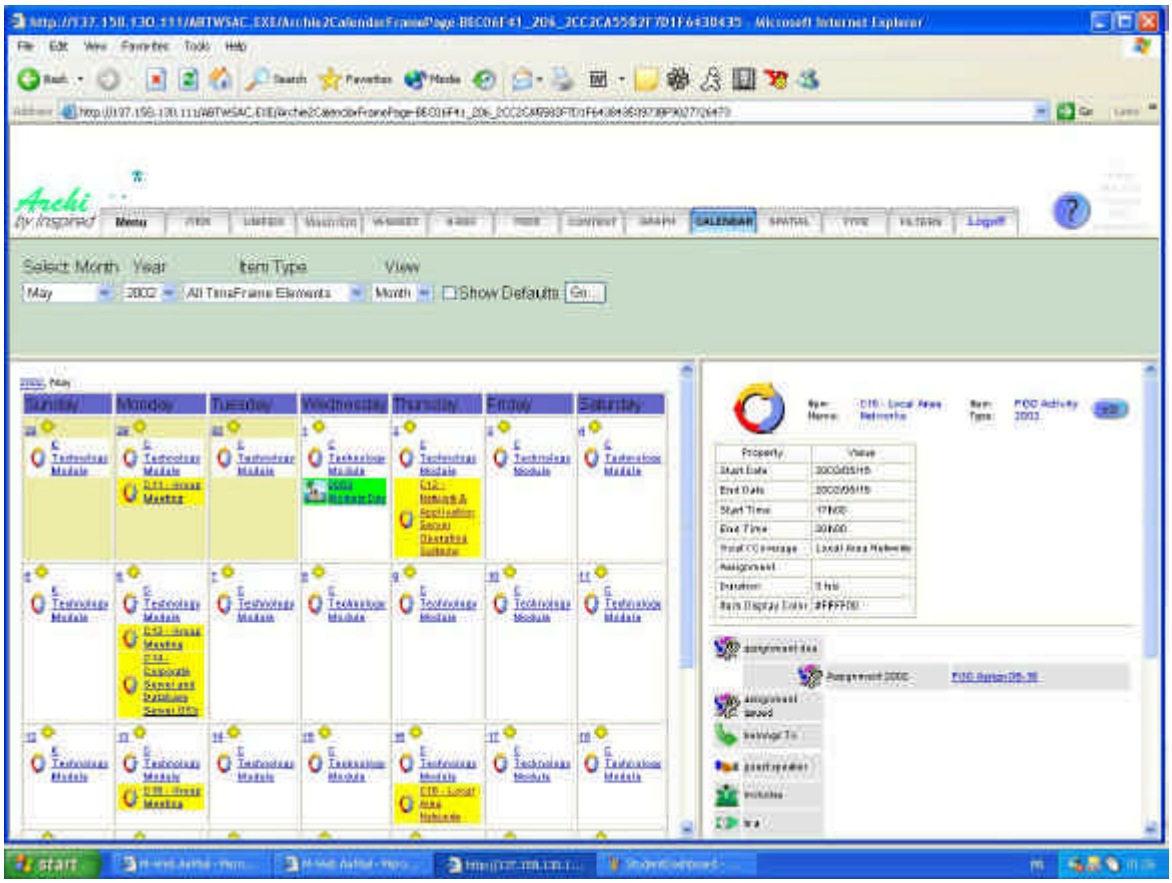

Figure 4 - Calendar view with Item Details

- Submission of material by students, including assignments, presentations, essays and readings
- Administration: Management of submissions, study groups, bookings, allocation of mentors, assignment of lecturers etc.

Web Site Generation

In past years, students were provided with two bound volumes of readings plus various handouts which would be distributed during lecture sessions. We wanted to provide more information, link to online sources and save on copying and distribution costs. At the same time, we did not want to force students to be online to access materials for study purposes, especially during preparation for exams. To this end, we used a feature of the tool which allowed us to specify a set of information (types) that we wanted to output as a web site. It was then a simple matter to generate a free-standing web site which we were able to distribute to students on CD. The generated site had approximately 1600 pages, many of which referred to electronic documents also placed on the CD, usually in Adobe Acrobat (.pdf) format. Other pages provided links to online resources, such as forums, vendor sites, standards bodies etc. on the world wide web. Two versions of the CD were generated: one at midyear with content from the System Delivery and Technology Modules; and a second at the end of the year with content from all modules (adding Project Management and Strategic use of Information Systems). These resources were highly appreciated by students.

Evolution

During the year, a number of additional requirements were identified, and various innovations became available in the tool, or were added to support requirements.

- **Instance Level Security** was required to support submissions by students in such a manner that they could be viewed, but not updated, by others. While desirable for contributions to the course, such as presentations and recommended readings, it was essential for marked deliverables, such as essays. This was supported by recording the "owning" user on the item and allowing update only for the designated user.
- Initially, lectures, submissions, assignments etc. were of specified types (e.g. Acrobat, Power Point, Word). This required having multiple attributes to allow students to submit in the available or preferred type. The multiple attributes introduced complication and confusion in retrieval. **A polymorphic document type** was added which supports uploading of any kind of document that clients can handle. The limitation thus became what is popularly available rather than the knowledge base. We chose to support Acrobat, Word and Power Point files as the preferred formats, with HTML also allowed.
- A powerful **new browser** was added, the Master Detail view, which allows showing detailed content from any two linked types simultaneously. It is particularly useful for listing items related in a one to many fashion. Examples include: Readings within a given Subject; Students within a Study Group; Presentations from a Lecturer. It also greatly facilitates setup of detailed security.
- To fully exploit the new browser, some **relationship types were renamed**. This allows listing various types of resources under a Master, provided the relationship type is the same. E.g. From Subject, we may list "includes" items for Reading, Book, Reference Source and Lecture from a single query.
- Enhancements to the tool also allowed easy addition of management features such as **tracking editors and dates** that items were modified.
- Our initial implementation did not fully utilise **security features** in the tool. This was in the interests of getting things into production and content loaded quickly. We also felt that we needed to gain experience and let things "evolve". As time went by, and under pressure from Commerce Faculty system administrators, we tightened up security considerably.

Feedback From Year One

During 2001, we gained significant experience and feedback. Specific items included:

- In the 2nd quarter, the installation was moved to a **new web server** for the Commerce Faculty. This was a powerful machine with high memory and fast dual Pentium processors. Initially, this was lightly loaded and response times improved remarkably. However, as the full Commerce workload was migrated across, the server became saturated. Major problems were also experienced with stability of Windows 2000, this being the first major server migrated to this environment in Commerce. Frequent application of patches and reconfigurations required frequent reboots. The knowledge base server is not an NT Service, since it is a cross platform product. On many occasions, operations rebooted the server without restarting the knowledge server.

- Problems have been experienced with **network traffic saturation** in the UCT environment, leading to frustrations, particularly with respect to uploading large files. We encouraged students to submit their work as Acrobat format (.pdf) files rather than larger Word or Power Point formats. In this way, most uploads were under 1Mb, which seemed to be a practical limit for reliability.

- The **Ms Access database** performed surprisingly well given the number of users on the system. This is probably attributable to the internal management of storage access by the persistence layer within Archi. Effectively, access sees one "user". There were three occasions during the year when we had to restart the database to deal with memory problems or cacheing errors.
- There was some resistance from students initially to the "confusing and complex" **user interface.** This was overcome with familiarity. We did conclude that a gentler introduction to the facilities of the tool was required for the following year. The complexity comes from the generic nature of the tool, the richness of views and information provided and the flexibility offered. By contrast, the tool was evaluated during its development for user interface efficacy for knowledge browsing by an earlier postgraduate team [Peters, Pieterse, Robinson, 2000], and found to be superior in locating and analysing information to other tools surveyed (Lotus Notes and The Brain). The results indicate that, while the interface is initially intimidating, it is, with familiarity, effective and efficient. Intermediate and experienced knowledge workers preferred the Archi interface, while novices preferred The Brain. Interestingly, all groups were more efficient in locating required knowledge using the Archi tool.
- A colleague, Jean-Paul van Belle, introduced a centrally mandated teaching support system, **WebCT,** in his INF316 undergraduate program on Information Systems Architectures during 2001. Following quite extensive experience with this system, his students were also given access to Archi and asked to provide opinions. Students particularly liked the off campus access facilities and the ability to download curriculum, readings etc. Also appreciated groupwork features (being able to upload and share, discussion groups). Calendar and integrated e-mail were not as highly appreciated and they stated that they did not need whiteboard (realtime) collaboration facilities. 56% concluded they would recommend Archi for future courses, 44% said maybe, none said "No" [Jean-Paul van Belle, 2001].
- **Students were surveyed** about their experience of the course overall at the end of 2001. Part of this concerned the knowledge base. The overall score achieved was an average of 4 out of 6 (min 2, max 6). Most negative comments pertained to availability (server down time) and network performance issues. Some mentioned difficulty in coming to grips with interface. Positive comments were received regarding usefulness, comprehensiveness, accessibility off campus, flexibility and the generated CD.

Year Two Challenges

On preparing the environment for the second year of operation, it became apparent that this was more involved than simply "refreshing" a static web site for a course, as was typically done in the past. The difficulty mainly arises from the desire to preserve rich information built up in the previous year, while still providing a "first time through" experience for the new group. Subtle difference in course structure, timings etc. mean that the course of the same code number this year, is not the same knowledge item as last year. Content which can be treated equivalently from a structure perspective may need to be separated from a security perspective - for example: we wanted to make previous year's student presentations available to the current group for reference, but not before the current year's students had attempted their own work and research. On the other hand, these presentations had to remain available to last years students moving on to Honours study. Some of the strategies we adopted include:

- Student, Subject, Reference, Reading, Text Book etc. knowledge was preserved

- Course details “cloned” rather than identical. Necessary subtle changes made.
- Events, Activities, Deadlines, Presentations were created as different, but equivalent types
- Progressively release last year’s student generated material to this year’s students
- Much tighter security, with differentiation between student groups as to what they are permitted to “see”.
- Use of filters to control which instances of record types are visible to a group. Filters automatically activated at login based upon user profile.

In addition to the migration issues surrounding the knowledge items and user base, we also took advantage of interim developments in the tool capabilities. These are detailed next.

Year Two Improvements

- A “**student dashboard**” was created using the Spatial Browser feature. This allows a quick entry point to commonly required information without having much understanding of the other capabilities of the tool.

- The custom menus feature was exploited to capture commonly used views and make these available from a **simple list menu**. This is the default entry point for students (until they chose to alter it for themselves). This strategy helped students to quickly locate useful information while exposing them to various views available in the tool.

- The **Simple Browser** (a view equivalent to the generated web site structure) was added as an easy and intuitive means for students to reach information until they become more acquainted with the knowledge structure. This view works as follows:

    - simply lists all types of information available
    - When a type is selected, lists all instances of that type
    - When an instance is selected, lists details of the item and relationships to other items as hyperlinks

- The Knowledge Server and database were **moved to a separate server** (a desktop Pentium II machine) communicating via TCP/IP with the Commerce web server over the LAN. Simultaneously, the server was ported to the open source mySQL database system. We had planned to use Microsoft SQL Server, but it transpired that the available licences covered teaching use, not administration. The move to the new server has alleviated problems associated with web-server reboots. Although the knowledge server is small (128 Mb, 466 Mhz) it is dedicated and performance is good. There are currently some 120 accounts on the system and peak traffic is around 20 concurrent users. Between March and May, the system had supported some 55 000 requests with about 12 reported errors. These were usually related to corrupt URL’s requesting services not present.
- Calculated Properties feature of the tool was used to allow capture of essay **mark sheets**, providing automated calculation of marks. The advantage is that users retrieving a paper can also see how that work was rated. Obviously, marks require tight security over who can edit them!

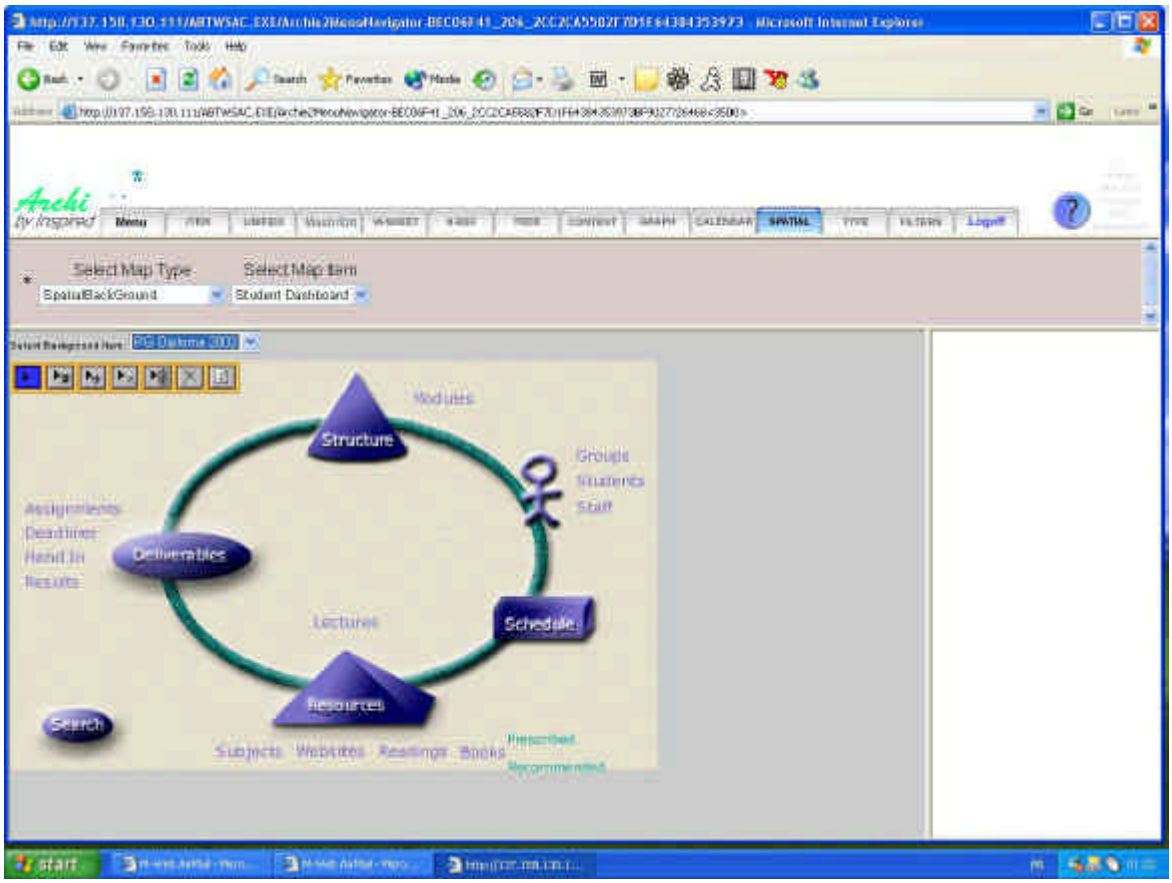

Figure 5 - Student Dashboard 2002

Conclusions

We had expected the knowledge management site to provide a richer, more engaging experience for students and to support their work off campus. These things it certainly has done. We also expected the dynamic nature of the site and offloading many of the maintenance tasks (e.g. uploading and linking readings, presentations, essays) to students to reduce the administrative burden of the web site. This too was achieved, on average - a caveat is that getting this established in the early stages of a course requires some higher inputs and support until students are competent and confident.

The move between years is not trivial, and requires more subtle model changes and migration than one would expect. This does not necessarily involve more effort than migrating a typical web site, but certainly requires higher levels of skill to achieve successfully. The upside is that a rich knowledge base, built by a community, can be carried forward and provide a platform for higher performance of the subsequent student groups.

Infrastructure performance is fairly critical, as the web ceases to be an occasional information source and becomes an essential operational tool in the administration and running of the course.

Facilities like this are increasingly important in providing a supportive and dynamic environment to students, particularly those studying part time. Unfortunately, the setting up of web sites, repositories, knowledge-bases and the like earns no credits towards job performance for faculty at UCT. This should be reconsidered for the future.

Future Work

We plan to take advantage of several new developments in the tool and its application. These include:

- Ability to create an **ontology**, with concept definitions, relationships, dependencies and references. This is achieved using an extension of the subject index built previously. A new custom view allows listing a concept and all prerequisite concepts, with their definitions and reference links. This has been prototyped with commercial clients and has been extremely well received.
- Definition of **Domains**, which group related types and allow easy hiding and showing of these sets, will be exploited to "hide" types which are not relevant in certain contexts, thus simplifying user interface and reducing "noise". These will also simplify the administration of security.
- **Events monitoring** capability will be exploited to notify students (and staff) when content of interest has been added or amended (e.g. A new assignment or marks are posted).
- **Text searching** capabilities should be expanded to allow searching internal content of managed documents and possibly even linked unmanaged documents. This is not yet operational and will probably require integration of a third party search engine.

Further survey work should be undertaken with the user community to determine usage patterns, likes and dislikes as well as actual measured effectiveness. These inputs can be used to further improve the knowledge base, the tool capabilities and the experience of the user community.

Bibliography

Adobe, 2000, Acrobat 4.0 Product Reference, Adobe, 2000

Inspired, 2002, Archi Brochure - Release 2.0, Inspired 2002. Accessible at www.inspired.org

McLeod, Graham, 2001, PAMELA: A Proto-pattern for Rapidly Delivered, Runtime Extensible Systems, Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD) Workshop, Interlaken, Switzerland, June 2001

Peters, Lindsay-Anne; Pieterse, Shelley-Ann; Robinson, James, 2000, Comparison of User Interface Efficacy for Knowledge Browsing, Empirical Research Report, University of Cape Town Information Systems, 2000

Van Belle, Jean-Paul, 2001, Using WebCT in IS: The INF316 Experience, Research Seminar, University of Cape Town Information Systems, 2001, jvbelle@commerce.uct.ac.za