Minerva

Overview and System Documentation

Updated on:
23/02/05

Minerva Web Team
Academic Unit of Medical Education
University of Sheffield
1. What is Minerva?

Minerva is the University of Sheffield’s Medical School's Managed Learning Environment. It is a web-based database-driven central resource for students and staff fulfilling many functions relating to the delivery and management of the undergraduate medical curriculum. These include information about the structure of the curriculum, timetables, course handbooks, learning materials and resources, clinical placements, student selected components, integrated learning activities, personal portfolios, assessments and news items. It also provides access to the core curriculum database.

The name Minerva was chosen for a number of reasons. Minerva was the multi-talented Roman goddess of learning, medicine, science, arts and crafts. She was the patroness of academics, physicians and artisans. In Roman tradition, she has been equated with Meditrina, the healing goddess, from which 'medicine' was derived. The figure of Minerva is also incorporated into the ceremonial Mace which was presented to the University of Sheffield by Pro-Chancellor Alderman Franklin in 1909.

1.1 Background

At the University of Sheffield there are 1,012 full time undergraduates in medicine with 220 staff contributing to the teaching. Over 750 clinicians (consultants and general practitioners) provide clinical teaching in hospitals and the community. The curriculum at Sheffield had in the past been fairly traditional with a lecture based pre-clinical component based on medical sciences followed by a clinical component largely based in hospital settings.

A major impetus for curriculum change had arisen from a visit by the medical profession's regulatory body, the General Medical Council (GMC, 1999) and the Quality Assurance Agency (1998) Subject Review. As a consequence a two staged strategy emerged for the curriculum review process: a refinement of the current curriculum and a plan for a more radical revision, which started in September 2003. The major revision is based around four common themes:

- Development of an outcome-based curriculum framework: the aim is to produce graduates who are able to fulfill their role as junior doctors in the National Health Service (NHS) and who also possess the generic skills expected of students attending a research-led university.
- Development of a case / problem-based learning educational pedagogy.
- Increasing student choice so that students can select a minimum of 25% of the curriculum in the form of student selected components (GMC 1993, 2002).
- Increased community-based learning (i.e. students going out to see patients in the community)

Although coordinated by a curriculum review team drawn form a wide range of scientists and clinicians, the design of the new curriculum has emerged collaboratively by involving all the major stakeholders at open meetings of teaching staff, the community and with students. There has been a commitment to wide consultation with the intent of developing a greater sense of ownership of the curriculum.
The School of Medicine and Biomedical Sciences, University of Sheffield joined a consortium of Universities from the UK (Newcastle, Durham, Northumbria and Nottingham) as a non-funded partner within the Teaching and Learning Technology Programme Phase 3 (TLTP3). The overall aim of the project was to develop and disseminate integrated communication and information technology (C&IT) based approaches to facilitate, manage and support the delivery of the curricula at the member Universities. Although the TLTP3 project ended (May 2001), the Universities involved continue to maintain close working relationships.

From this consortium, a basic software framework was developed which formed the basis for the first version of Minerva (originally called the Sheffield Networked Learning Environment). Minerva has further developed since 2001 and now plays a key role within the School of Medicine and Biomedical Sciences in helping to deliver and manage the undergraduate medical curriculum.

Ideally the process of creating a managed learning environment is an iterative process between the curriculum planners and the software developers, so that the cycles of planning and implementing change inform both the curriculum and the tools that support it. Minerva helps in supporting all aspects of the planning, delivery and evaluation of the medical curriculum. It is educationally led (specifically for the curriculum) and is not technology driven. It is designed to make use of existing information and to reflect the way in which people work. It is process (not product) driven so that it is mapped to the needs of the institution. This involves project staff visiting institutions and departments, listening to the way in which people work and to their needs and problems, forming ideas of possible solutions and obtaining copies or examples of processes, documentation and data. This contrasts starkly with projects where software is installed and institutions have to change the way they work to use the software.

Minerva is web-based, open sourced, flexible, customisable, database driven and was originally developed in consortium with other UK medical schools. It is core and project-based funding and is not a commercial system.
2. Key Minerva Functionality

2.1 Authentication

Minerva has a number of different views, dependant on the user's role. Roles currently in use are Staff (SysAdmin; SuperUser) and Student (Foundation Year; Phase 1A; Phase 1B; Phase 2; Phase 3A; Phase 3B; Phase 4; BmedSci). When a user logs into Minerva, they are automatically directed to their homepage. For example, a Phase 1A student is directed to the Phase 1A home page. A basic portal therefore exists for login. Minerva is restricted to Medical Faculty staff, NHS staff who are involved in teaching on the medical course and medical students.

Fig 1: Phase 1A Home Page
2.2 The Core Curriculum Database

The core curriculum database together with the authentication system is at the heart of Minerva.

The core curriculum can be described as a repository of refined outcome objectives for multiple clinical problems (Newble and Bax). It is derived from 92 clinical problems e.g. chest pain, breast lump etc, each one collaboratively produced to a blueprint. It defines aspects of clinical competency including professional behaviours and the underpinning sciences and forms the core curriculum. It is useful as it: supports curriculum planning / implementation e.g. at phase level; guides assessment; guides staff for teaching activities; guides students on the breadth and depth of learning.

The core curriculum is maintained in an interactive, searchable database which is accessible to staff and students via Minerva. A long-term aim is to be able to match outcome objectives designed at the strategic level to learning objectives as delivered by the teachers to students, as noted in the on-line course handbooks.

**Fig 2: 93 Problems Forming the Core Curriculum Database**
2.3 Resource Upload Facility

Minerva has a resource upload facility that allows any member of staff to upload a resource (e.g., PowerPoint, Word, Movies etc.) to support their lecture(s). The lecturer who uploads the resource personally 'owns' it. No-one else can delete it from Minerva or edit its title or keywords, which are used by the Minerva search engines. The upload template is designed in such a way that staff need very little IT skill to place a resource into Minerva. All uploaded resources must also adhere to current copyright laws.

Fig 3: Resources Uploaded to Minerva
2.4 Online Course Handbooks

All course handbooks are provided online and contain the lecture aims, objectives, course outlines, methods of assessment and recommended reading.

Using Microsoft Word 2003, Word Styles allow the document to be specifically structured for viewing and searching online. Relevant Styles are also converted into a tagging system and allows for templates and dynamic links to be automatically inserted into the online document (such as the resource upload template, dynamic links to the core curriculum database, dynamic links to the University’s online library system called STAR using ISBN). This also allows for curriculum mapping exercises to take place.

Fig 4: An Online Course Handbook

Reading List

Although all the core information in this module is given in the teaching sessions, some of the lectures have provided reading references at the foot of each lecture outline. The books identified as recommended reading references usually contain core knowledge to supplement the lecture and will usually refer to one of the core texts for the Phase. All of these books are available in the Health Sciences Library and the University Bookshop carries sufficient copies for the year. Other further reading references are for more general interest. Although most of these texts are in the library, we cannot always guarantee this.

The standard recommended textbooks to support the module are:

**Anatomy: a dissection manual and atlas**. Jocki S. Churchill Livingstone 1996

Look up the availability of this title [ISBN: 0443053562] in the Library using STAR.

**Essential Clinical Anatomy**, 2e. Moore KL, Agur Lippincott Williams and Wilkins, 2004

Look up the availability of this title [ISBN: 0781759404] in the Library using STAR.

The 1st edition is also acceptable.

**Human Histology**. Stevens A, Lowe JS, Mosby 2nd Ed 1997

Look up the availability of this title [ISBN: 0723424593] in the Library using STAR.

**Basic Medical Biochemistry**. Marks DB, Marks AD, Smith CM, Williams & Wilkinson 1996

Look up the availability of this title [ISBN: 068369962X] in the Library using STAR.
2.5 On-line Assessment

TouchStone is an online assessment system which has been specifically created to match the needs of the Medical School and to integrate seamlessly with Minerva. Currently, Touchstone is being used for weekly formative assessments in the first two years of the course, but there are plans to introduce formative on-line assessments to all years. The questions are formatted as extended matching questions (EMQs) and enable students to take the formative test at any time and from any computer. Automatic textual feedback is provided and subsequent attempts are recorded so each student can monitor their progress throughout the academic year. All marks are recorded, together with username, start date, duration and IP address.

In addition to the formative assessments, summative tests and surveys can be delivered. Each assessment paper, regardless of type, can hold any number of questions on each screen and can consist of up to 30 screens in total. In line with the Medical School’s assessment strategy, types of questions most commonly used are extended matching and multiple choice. Other types available are true / false, multiple response, fill-in-the-blank, ranking, extended matching multiple answer and survey.

Fig 5: An Example Extended Matching Assessment Paper
2.6 Professional Portfolios

Phase 1A and 1B students (first and second year) currently use their portfolio to hold their Student Selected Component (SSC) assignments and their Significant Events for the ICE (Intensive Clinical Experience) SSC. In addition, SSC assessments and Integrated Learning Activity (ILA) marks of overall competence and professional behaviour are also uploaded to their portfolio. The portfolio is primarily being used to support the assessment of the professional behaviours component of clinical competence by collecting frequent longitudinal evidence. A new facility for students to record any achievements, reflections and memos is also now in place. The portfolios will continue to develop and will roll out with the curriculum over the next few years.

The Phase 4 (final year) portfolio will be implemented for 2005. It contains a record of achievement for the various assessed elements of Phase 4. These include the Clinical Attachment, Clinical Competencies, the Observed Long Case and PRHO shadowing. The portfolio will be automatically updated to show their assessment marks. The Phase 4 portfolio will also provide a direct to the online logbook used by the South Yorkshire and South Humberside Postgraduate Dean’s office.

Fig 6: An Online Portfolio
2.7 Integrated Learning Activities (ILAs)

Minerva allows for enhanced teaching innovations, such as ILAs. ILAs consist of a spine of core problems and cases. The aims of ILAs are: to give students a clinical context in which to place all the medical sciences that they are encountering and to integrate this with their early clinical experiences; to introduce the core clinical problems; to develop self-directed learning skills; to develop problem-solving skills; to develop clinical reasoning.

Students are encouraged to work in tutorless groups within the lecture theatre to work out the learning objectives of carefully designed cases. The students use Minerva to point them to a number of prepared learning resources including on-line formative assessments and multimedia learning objects such as x-rays, microscope slides etc. Students then bring back to the second session a more synthesised understanding of the patient problem based on their research into the learning objectives.

**Fig 7: An ILA**
2.8 Student Selected Components (SSCs) and Online Marking

26% of the new curriculum is delivered in the form of SSCs, as required by the General Medical Council (GMC 1993, 2002). Their purpose is to extend studies and experiences beyond the content of the core curriculum. Students can choose and pursue topics of personal, academic and vocational interest, apply and develop research skills and enhance personal and professional development in relation to communication skills, self-direction, self-management and reflection.

All of the SSCs are managed via Minerva. An online marking system is also in place for the Intensive Clinical Experience SSC, where students spend a week with doctors, nurses and social services staff to gain some real experience of health and social care. To help ensure students get the most out of ICE they reflect on two significant events that occurred during their time on placement. These are submitted and marked online. There are plans to extend the functionality for on-line marking to other parts of the curriculum.

Fig 8: A SSC Home Page
2.9 Clinical Attachment Placement Software

The clinical attachment rotation blocks for Phase 2 and Phase 4 are managed by Minerva. Students are attached to doctors and consultants at one of the Sheffield hospitals or at one of the eight district general hospitals. This matching process is done by random rotation scripts (currently written in PERL). The students then log into the attachment system at the appropriate time to find which supervisor they have been placed with.

There are further plans to extend this system to the Phase 3A and 3B rotation blocks. The introduction of this system has considerably cut the length of time it required for school administrators to manually match the students to hospitals and supervisors.

Fig 9: Clinical Attachment Login Screen
2.10 Discussion Boards

All phases have access to online discussion boards, split into various categories such as social or academic discussion. These are very heavily used and give the students (and staff) opportunities to generate healthy online debate about the course or medicine in general. All boards are monitored.

Fig 10: Phase 1A Discussion Boards
2.11 Online Evaluation and Reporting

Students are given the opportunity to evaluate teaching quality for many parts of the course online. Both qualitative and quantitative data can be collated. The data is also made available to staff online for analysis and reporting.

**Fig 11: An Online Evaluation Form**

<table>
<thead>
<tr>
<th>SSC Theme</th>
<th>Please Select a SSC Theme:</th>
<th>If a text box is visible below, please enter your supervisor’s name:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>A: Meeting the Aims</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither Agree or Disagree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I was able to extend studies &amp; experiences beyond the content of the core curriculum (depth or content)</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>2. I had the opportunity to choose &amp; pursue topics of personal, academic and vocational interest</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>3. I was able to apply &amp; develop research skills in relation to information gathering, research methods</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>4. I was able to enhance professional development in relation to communication skills, reflection, self-direction/ self-management</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B: SSC Organisation</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither Agree or Disagree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Before I started I was able to obtain enough/satisfactory information about the SSC</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>6. The administration, before, during and after was satisfactory</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>7. The supervisor was welcoming</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>8. I understand what I was supposed to achieve</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
2.12 Search Tools

Each Phase in Minerva has a ‘Search My Curriculum’ link that allows searching for online resources. Current resource types available for searching are PowerPoint presentations, Word documents, PDF documents, web sites, online assessments, excel spreadsheets and videos. Resources can be searched by type, phase, keywords or uploaded date and can be ordered in a variety of ways, such as file size. On line course handbooks can also be searched by keywords.

Fig 12: Search Tools
2.13 News Announcements

News announcements, with attachments if required, can be posted to phase homepages by the Medical School's administration team or staff members.

Fig 13: News Announcements
2.14 Other Functionality

Other functionality includes links to online timetables, assessment criteria, tracking systems, mentor handbooks and phase specific contact information. Minerva can also provide many other functions and features. Due to the open-source philosophy of the system, new system requirements can be ‘bolted’ on with relative ease.
3. Technical Specifications

3.1 Technologies Used

Sheffield Medical School has developed Minerva in-house to support the new undergraduate curriculum rather than rely on a commercially available system, such as WebCT or Blackboard. This decision has advantages in terms of providing greater opportunities to develop a system that is closely matched to the needs of the institution and has a greater sense of ownership amongst staff and students.

Zope (z object programming environment) is currently being used as the content management system to assist in delivering Minerva. It is an open source web application server primarily written in the python programming language used for building content management systems, intranets, portals, and custom applications. It is object orientated and allows for simple connection to existing databases. More information, including a free download, is available at [http://www.zope.org](http://www.zope.org).

Other technologies used currently are the SQL database MySQL (others could be used), Microsoft Word 2003 (used for authoring the online course handbooks), Python, PERL and PHP.

We are currently housed on the University of Newcastle’s Faculty of Medicine server. This is not a permanent (or ideal) solution and there are plans in place to transfer the server to Sheffield University in 2005. The server delivery platform is UNIX and current system software version details are Solaris 8; MySQL 3.23.50; Zope 2.6.0; Python 2.1.3; Perl 5.6.1; PHP 4.3.6; Apache 1.3.31; OpenSSL 0.9.7d and OpenSSH 3.8.1p1.
3.2 Minerva Authentication System using Zope

The nature of the content of Minerva means that certain parts of it require user authentication before it can be accessed. In addition, by requiring authenticated access to Minerva, we can provide ‘layered authentication’ and this makes creating a portal approach for the end user of the system much easier.

The model chosen for authentication is one in which users are authenticated against an SQL database. This allows for easy management of user accounts and provides additional functionality. We use a MySQL database because it is fast, easy to use and we have experience of it. There is nothing to prevent PostGres, Oracle or databases accessed via ODBC from being used.

The Zope environment provides fine-grained control to authenticated resources within Minerva. This relies on users having one or more roles as defined by their authentication characteristics. For example, someone authenticated with the role of Staff has the permission to Access Contents Information whereas someone authenticated as a Student does not. In other words, staff members will be able to see this page, whereas students will be asked to authenticate.

It follows then that the minimum information needed to authenticate using Zope and an SQL database is:

**username**: a unique identifier of the person. We use the first part of the users Sheffield University email address (i.e. md1abc)

**password**: the password required to authenticate. At the moment, this information is sent ‘in the clear’, i.e. not encrypted, but there are several Zope products and methods of working that can encrypt the password for additional security.

**roles**: this is a list of roles the user will have once they have authenticated. People normally have more than one function in an institution and by providing a list of roles you are mimicking the way people work rather than forcing a new system upon them.

To satisfy the requirements for our model, we have used a Zope Product called UserDB. It is an authentication source that uses a MySQL table to hold user details.

This is most definitely not the only way that we could have authenticated users of Minerva as the User Management section of Products at http://www.zope.org/Products/user_management yields some 49 Products that could be used. UserDB is at http://www.zope.org/Members/otto/userdb.

All users who initially enter a Zope site have a variable defined in Zope called AUTHENTICATED_USER. Until someone authenticates using one of the methods above, the value of this variable is ‘Anonymous User’. To provide an authentication ‘hook’, you need to create an ‘acl_users’ instance in the folder that contains objects that require users to login. To enable the acl_users folder to do its work, it is necessary to change the ‘Security’ settings for any objects for which you wish to have authentication enabled. The Zope site has many resources to help with setting up security settings. Essentially, you turn off the ‘View’ permission for ‘Acquire Permission Settings’ and set it for the roles (e.g. ‘Staff’ and ‘Student’) you want to be able to use (or view) the resource. Zope will then take care of presenting authentication dialogs when an unauthenticated user wants to access the said resource.
Once the Zope system decides that you need authenticating, you are presented with an authentication dialog (in the case of UserDB and others) or a simple popup dialog box (as in the case of the stock authentication scheme that comes with Zope and the method we currently use).

If the system does not find a matching username, password pair in the password table, that is the SQL query returns zero rows, your authentication will fail. If you are a member of staff, and your username and password match those in the password table, the system will return a list of roles that you have. This allows for great flexibility, in that one user may have the Staff role, as well as SysAdmin, Admin and SuperUser role, whilst another user may only have the role of Staff. At the time of writing, students only have the one role, Student.

The open-source nature of Zope means that we can use many different authentication schemes, and choose the one that offers the most security and/or flexibility.
3.3 Creating the Online Course Handbooks

The Minerva course handbooks are built from Microsoft Word 2003 XML documents. These handbooks are received in Microsoft Word from the Medical School’s administration team. The conversion process to SQL consists of an initial transformation of the Word document into an XML document. This transformation used to be achieved by an off-the-shelf program written in the language OmniMark. It is now achieved by using Microsoft Word 2003 to generate the XML. The XML document is well word-processed, in that Word Styles are used extensively. Finally, a PHP program converts the XML document into a sequence of SQL statements that are fed into the MySQL course handbooks table.

These Word Styles allow the document to be specifically structured for viewing and searching online. Styles are also converted into a tagging system, and as mentioned earlier, allows for templates and dynamic linkage to be automatically inserted into the online document such as the resource upload template, dynamic links to the core curriculum database, dynamic links to the University’s online library system called STAR using ISBN.

The Word document is shown using the style area width preference set to at least 1.5cm so the style used can be viewed by the editor of the document. The word styles used in the document are translated into a form more suitable for inclusion into an SQL database by saving it as an XML file in Word 2003.

This is valid and well-formed XML, but is not sufficient for our purposes. This is mainly due to the document being ‘flat’ and having none of the structure that has relevance in the original Word document (such as headings representing hierarchical layers of content). We therefore use another program, currently written on PHP, which converts the XML into an XML format that is more easily changed into SQL.

The most important entry is the one for the ‘Heading 1’ of the document, i.e. the root of the document. It should be noted that the style ‘Heading 1’ can be named whatever you like – that’s the point of the translation document. It is important to realise that as things stand with the Minerva translation scheme, there can only be one ‘Heading 1’ (whatever it is called) in a translation document.

Current styles in use are:

- Heading 1 - represents the main heading of the Word document (one only).
- Heading 2 - represents the ‘child’ of Heading 1. This is used to structure the document into hierarchical layers. More layers could be used if required.
- BodyText - represents the main content of the document.
- ListBullet.
- Resources - resource upload template tag (dynamic link).
- ISBN - ISBN tag for the University’s library system called STAR (dynamic link).
- Core - lecture outcome link(s) to the complete problem(s) in the core curriculum database (dynamic link).
Others can be added in as required.

Below is an example of a formatted Word document, using the appropriate Word Styles.

**Phase 1A (2004/05): Introduction to Medicine and Medical Science**

Welcome!

Please read this Handbook carefully – it contains much of the information you need for the course, as well as important information about the examinations.

This module is composed of two parts. In the first four weeks you will encounter an overview of the anatomy and function of the cardiovascular, respiratory, gastrointestinal, genitourinary, endocrine and nervous systems. The remaining four weeks will introduce you to cell biology, structure and function, and to some of the common disease processes. Throughout the module you will have the opportunity to see real patients in the clinical demonstrations; these serve to highlight the clinical relevance of the core lectures and practicals. We will also introduce you to a number of integrated learning activities and student selected components. Integrated into this module is the start of the module “Medicine and Society”.

The module begins with a series of introductory sessions. Please make every effort to attend, since much of the administrative and general academic information relevant to this module will be discussed and it is your opportunity to ask questions.

As Module Coordinator, I will make every effort to be available to answer problems and queries throughout the module and you should not hesitate to contact me. Please feel free to give formal or informal feedback about the academic organisation of the module at any time. If for whatever reason you are getting into academic difficulties with the course, let us know sooner rather than later. Many problems when addressed early can be solved before they get too serious! Any purely administrative matters should, in the first instance, be discussed with Karen Kehtarnavaz in the Student Affairs section of the Faculty Office on C Floor, address as below.

![An Example of a Formatted Handbook](image-url)
The same document is shown below as it would appear online after the translation process.

Fig 15: An Example of an Online Formatted Course Handbook
Shown below is ‘The Heart’ lecture in detail, with dynamic links generated to the core curriculum database (using the core tag).

**Fig 16: An Example of a Specific Lecture Linking to the Core Curriculum Database**

---

**Objectives**

At the end of the lecture, students will:

- Have a general understanding of the structure and function of the heart
- Have a general understanding of the nature of cardiac disease, with especial emphasis on ischaemic heart disease

**Core Clinical Problems**

This lecture is relevant to problems:

- 46. Cardiac arrest/sudden death
- 47. Chest pain
- 51. Low blood pressure
- 53. Murmur
- 54. Oedema
- 55. Palpitations/abnormal heart rhythm
- 57. Shortness of breath

**Relevance to medical practice**

Introducing how fundamental knowledge translates to the understanding and treatment of cardiac disorders.

**Assumed knowledge**
Shown below is the structure of the course handbook table (called element2) and the field content for the above course handbook (omitting the content field and for the first ten entries only, as there are many!). It should be noted that ‘sequence’ and ‘parent’ should be renamed to ‘parent’ and ‘child’ respectively. This error is purely historical and needs to be amended.

```
mysql> describe element2;
+-----------------+-----------------+--------+---------+-------+-----------+----------+
| Field           | Type            | Null  | Key    | Default| Extra     |
+-----------------+-----------------+--------+---------+-------+-----------+----------+
| sequence        | int(9)          | YES    |        | 0     |           |          |
| thetag          | varchar(255)    | YES    | NULL   | NULL  |           |          |
| parent          | varchar(255)    | YES    | NULL   | NULL  |           |          |
| content         | text            | YES    | NULL   | NULL  |           |          |
| level           | int(8)          | YES    | NULL   | NULL  |           |          |
| guideID         | varchar(255)    | YES    | NULL   | NULL  |           |          |
+-----------------+-----------------+--------+---------+-------+-----------+----------+
6 rows in set (0.00 sec)
```

```
mysql> select sequence, thetag, parent, level, guideld from element2 where guideld = 'iims_lectures_04_05' order by sequence limit 10;
+---------+--------+-------+--------+---------+
| sequence| thetag | parent| level  | guideld |
+---------+--------+-------+--------+---------+
| 13674   | Reading1| 13674 | 1      | iims_lectures_04_05 |
| 13675   | Reading2| 13675 | 2      | iims_lectures_04_05 |
| 13676   | BodyText| 13675 | 2      | iims_lectures_04_05 |
| 13677   | BodyText| 13675 | 2      | iims_lectures_04_05 |
| 13678   | BodyText| 13675 | 2      | iims_lectures_04_05 |
| 13679   | BodyText| 13675 | 2      | iims_lectures_04_05 |
| 13680   | BodyText| 13675 | 2      | iims_lectures_04_05 |
| 13681   | BodyText| 13675 | 2      | iims_lectures_04_05 |
| 13682   | BodyText| 13675 | 2      | iims_lectures_04_05 |
| 13683   | BodyText| 13675 | 2      | iims_lectures_04_05 |
+---------+--------+-------+--------+---------+
10 rows in set (0.15 sec)
```

Fig 17: The Course Handbooks Table
3.4 Zope Objects used to Display the Course Handbooks

There are a large number of Zope objects used to run Minerva, namely Folders; DTML Documents and Methods; External Methods; ZSQL Database Connections; ZSQL Methods and Python Scripts. PHP and PERL scripts are also used outside of the Zope environment, namely used for running Touchstone and the placement software. It is not possible to go into detail regarding the current Zope folder hierarchy and object content within this document. For more information, contact the Minerva web team directly. Some examples of how Zope is used in supporting Minerva are shown below, specifically in terms of the course handbooks and resource upload system.

The screen shot shown below shows some of the Zope objects used to display the uploaded course handbooks (all contained in a folder called ‘grinder’). These objects work in conjunction with a Zope method called ‘show_frames’ contained in the folder above (‘minerva’). As Zope is an object-orientated web delivery platform, the concept of acquisition is brought into play. If ‘show_frames’ is not in the folder where it is called, then Zope looks further up the Object Database until it finds the method ‘show_frames’ (in the folder ‘minerva’).
Following this example through, when a user clicks to view an online course handbook, the `show_frames` object is called (see the dynamic URL below). The guideid in this example is `imms_lectures_04_05`.

```
<html>
<frameset rows="90,*" cols="*" frameborder="NO" border="0" framespacing="0">
  <frame src="standard_html_header" name="topFrame" scrolling="NO">
  <frameset cols="25%,75%" frameborder="NO" border="0" framespacing="0">
    <frame src="/medfac/minerva/grinder/show_left_frame?guideid=<dtml-if guideid><dtml-var guideid><dtml-else><dtml-var id></dtml-if>" NAME="LeftFrame">
    <frame src="/medfac/minerva/grinder/show_right_frame_header?guideid=<dtml-if guideid><dtml-var guideid><dtml-else><dtml-var id></dtml-if>" NAME="RightFrame">
  </frameset>
</html>
```

Below is the `show_frames` method.
To display the handbook, we use frames (not ideal, but it works in this context). The left hand frame is calling an object in the grinder folder called show_left_frame; the right frame is calling an object called show_right_frame_header (in both cases, the dynamic URL calls in the guideid reference that is required for display). These two objects in turn call in a number of ZSQL methods and other objects, such as the tag objects, to display the handbook online.

The screenshot below shows an object called show_right_frame. This object in turn calls in two ZSQL methods (show_parents and find_tag). If the tag associated with a particular guideid is found, the information (content) is displayed on-screen. For example, if a tag ‘Heading 1’ is found, the content associated with Heading 1 will be displayed. If a tag called ‘core’ is found, an object called core is displayed.

(The core object contains the following code:

```html
<a href="/medfac/minerva/core/retrieve_all_core?problem=<dtml-var content>">
<dtml-var content>
</a>.<br/>
```

This object dynamically displays a hyperlink in a handbook to a specific problem in the core curriculum database. See Fig: 16).
3.5 Resource Upload

Minerva has a facility to allow staff to disseminate documents to students. This is done by uploading documents into the file system of the Minerva server. The system also updates a resource database table that identifies the resource owner, location of the resource within the curriculum, suitability for specific year groups and some metadata about the resource. When students use the Minerva course handbooks, they are presented with the resources in the appropriate context, as shown below and can search the database using Minerva’s search tools in a variety of ways to locate and download the required resource.

![Learning Resources Displayed in Context within the Course Handbooks](image)

**Online Course Handbooks**

- Introduction to Medicine and Medical Science
  - Lecture series (in date order)
  - Additional Teaching Resources (PPT, Word etc.)
  - Self Assessment (Quiz / Test)
- Cardiovascular / Respiratory
  - Lecture Series (in date order)
  - Additional Teaching Resources
  - Self Assessment (Quiz / Test)
- Gastrointestinal / Liver
  - Introduction
  - Lecture Series (in date order)
  - Additional Teaching Resources
  - Self Assessment (Quiz / Test)
- Medicine and Society
  - Lecture series (in date order)
  - Additional Teaching Resources (PPT, Word etc.)

**Printable Course Handbooks (PDF)**

- MBChB & BMedSci Course Handbook 2004/05
- Introduction to Medicine and Medical Science
- Cardiovascular / Respiratory

**Fig 18: Learning Resources Displayed in Context within the Course Handbooks**
Users with the appropriate authentication (currently staff only) are able to see the upload link that allows them to add resources to the Minerva file system, as shown below. The resource owner can also edit a resource title and keywords and delete resources that they have uploaded to the system. They cannot however, delete or edit resources that others have uploaded to the system.

Fig 19: The Upload Facility
Shown below is the resource upload template. There are five stages involved in uploading a resource to Minerva. The entire process requires very little IT skill.

Please select a file or URL to upload

Study Guide Identification Code: immn_resource_04_05

1. Select Resource

File: [Browse]

The above file will be copied onto the Minerva web server.

(Please ensure that the name of the file that you upload does not contain ‘&’ e.g. a&b_lect.ppt. The file will not download properly.)

OR

Web Site: [Enter]

The web site will be pointed to by the Minerva web server.

2. Set Properties

Title: [Enter]

This title will be displayed in the ‘Additional Teaching Resources’ web page. Please use descriptive headings.

Keywords: [Enter]

Keywords are used for searching. Separate using commas.

3. Set Date Available

Date Available From: [Day] [Month] [Year]

4. Select Phase

(please ensure you select the correct phase)

Phase: [Select Phase]

5. Upload

Before uploading your resource, please ensure that you comply with current copyright law. An overview of the main issues can be found on the copyright information page.

Upload

Fig 20: The Resource Upload Template
Resources are labeled in the database according to where they have been uploaded from and who did the uploading. This allows some additional contextual metadata information to be added to the resource description in the database alongside the information explicitly provided by the resource provider.

The web form shown below has the following underlying structure once all the HTML tags have been removed. Included are Zope dtml tags for some completeness.

```html
<dtm-if "REQUEST['AUTHENTICATED_USER'].has_role('Staff') or REQUEST['AUTHENTICATED_USER'].has_role('SysAdmin') or REQUEST['AUTHENTICATED_USER'].has_role('SuperUser')">
  <form action="create_resource_upload" name="myForm" method="post" enctype="multipart/form-data">
    <dtm-if "REQUEST.has_key('identifier')">
      <h5>Study Guide Identification Code: <dtm-var "REQUEST['identifier']">
    <dtm-else>
      <dtm-call "REQUEST.set('identifier', 'Not Known')">
    </dtm-if>

    1. Select Resource
    File
    <input type="file" name="file:file" size="27" value="">
    OR
    Web Site
    <input type="text" size="40" name="aurl" value="">
    <dtm-if "REQUEST.has_key('url')">
      <dtm-var "REQUEST['url']" missing>
    </dtm-if">

    <dtm-if "REQUEST.has_key('identifier')">
      <input type=hidden name=identifier value="<dtm-var "REQUEST['identifier']">">
    <dtm-else>
      No module code specified
    </dtm-if>

    2. Set Properties
    Title
    <input type="text" size="40" name="extra_info">
    Keywords
    <input type="text" size="40" name="keywords" value="">
    <dtm-if "REQUEST.has_key('keywords')">
      <dtm-var "REQUEST['keywords']" missing>
    </dtm-if">

    3. Set Date Available
    Date Available From
    <select name="resday">
      <option selected value="">Day</option>
      <option value="01">1st</option>
      <option value="02">2nd</option>
      etc...
    </select>
```

33
Listing 1: Resource Upload HTML Form

We use ‘hidden’ variables to pass additional information to the upload script. The form then calls the create_resource_upload python script shown in Listing 2. This script uploads the resource to the local file system, populates the resources table, adds an incremental number before the resource title (this is used as the prefix for the filename uploaded into the local file system which allows us to have more than one ‘lecture.doc’ in the directory space) and deletes the resource from the table and file system if required. This script was originally developed by Dr Tony MacDonald, Newcastle University FMCC. We have since altered parts of it to suit our specific needs and requirements.

import os, sys, string, mimetypes, stat, urllib, MySQLdb
import DocumentTemplate
# from Acquisition import Implicit
# from OFS.Image import File
# from OFS.content_types import guess_content_type

folder = "~/home/sheff/Resources_New/"

def upload(self):
    REQUEST = self.REQUEST
    # Determine what folder to upload the file into
    db = MySQLdb.connect(host="localhost", user="mysheff", db="shef_nle")
    cursor = db.cursor()
template_str = self.template.read_raw()
# template_str = self.template.document_src()
template = DocumentTemplate.HTML(template_str)

try:
    file = REQUEST['file']
aurl = REQUEST['aurl']
    identifier = REQUEST['identifier']
except KeyError:
    results = "Please select a file or URL to upload"
    theresult = template(self, results=results, identifier='')
    return theresult

try:
    split = string.split(file.filename, '\')
    filename = split[-1]
    # get index number
    cursor.execute("select min(resources), count(resources) from upload_index")
    result=cursor.fetchall()
    nxtfile=result[0][0]
    count=result[0][1]
    okay = 0
except NameError:
    results = "NameError, need more information."
    theresult = template(self, results=results, identifier='')
    return theresult

if file.read(1) == '':
    if aurl:
        oname = os.path.join(folder, "%s_%s_url")
        f = open(oname, "wb")
        f.write(aurl)
        f.close()
        content_type = 'url'
        if string.find(aurl, 'http://www.minerva.shef.ac.uk/touchstone/') == 0:
            content_type = "assessment/caa"
        filesize=os.stat(oname)[6]
        filename = aurl
        okay = 1
        extension = 'url'
        usebrowsertype= 0
        results = "URL [%s] has been uploaded" % aurl
    else:
        # see if the browser gave us a content type
        headers = file.headers
        if headers.has_key('content-type'): # browser has provided content-type
            browsertype = headers['content-type']
        else:
            browsertype = 'application/octet-stream'

        file.seek(0)
        # now get the extension (if any)
        dotposn = string.rfind(filename, '.')
        if dotposn != -1:
            extension = filename[dotposn+1:]
            result = self.lookup(extension) # lookup mimetype
            if result != None:
                content_type = result.dictionaries()[0]['mimetype']
else:
    content_type = 'application/octet-stream'
    usebrowsertype= 0
else:
    extension = ''
    usebrowsertype = 1
    content_type = browsertype
filename = urllib.quote(filename)
filename = string.replace(filename, ' ', '_')
oname = os.path.join(folder, "%s_%s" % (nxtfile, filename))
f = open(oname, "wb")
body=file.read()
f.write(body)
f.close()
filesize=os.stat(oname)[6]
okay = 1
if okay:
    results = "Filename %s has been uploaded. It is %s bytes in size" % (filename, filesize)
else:
    results = "Sorry, the file was not uploaded. Please contact the system administrator."

if okay:
    try:
        self.add_details(filenumber=nxtfile, filename=filename, filesize=filesize, content_type=content_type,
                       identifier=REQUEST['identifier'], extra_info=REQUEST['extra_info'],
                       keywords=REQUEST['keywords'], extension=extension,
                       user=REQUEST['AUTHENTICATED_USER'].getUserName(),
                       resyear=REQUEST['resyear'], resmonth=REQUEST['resmonth'],
                       resday=REQUEST['resday'], phase=REQUEST['phase'])

        cursor.execute("select counter from upload_index where resources=%s", nxtfile)
        result=cursor.fetchall()
        counter=result[0][0]

        # if only one index entry increment the value else delete the used entry
        if count == 1:
            nxtfile = nxtfile + 1
            cursor.execute("update upload_index set resources=%s where counter=%s", [nxtfile, counter])
        else:
            cursor.execute("delete from upload_index where counter=%s", counter)

        db.close()
    except:
        str = "%s\n%s\n%s" % (nxtfile, filename, sys.exc_type, sys.exc_value)
        results = "Sorry, there was a problem adding the file uploaded details to the database\n%s" % str
else:
    results = "No file was specified"

template_str = self.template.read_raw()
template = DocumentTemplate.HTML(template_str)
theresult = template(self, results=results, identifier=identifier)
return theresult
def delete_entry(self, REQUEST):

    filenumber = REQUEST.filenumber
    datafile1 = REQUEST.filename
    extension = REQUEST.extension

    db = MySQLdb.connect(host="localhost", user="mysheff", db="shef_nle")
    cursor = db.cursor()

    cursor.execute("delete from res_data1 where filenumber = %s", filenumber)

    if datafile1 != '' and datafile1 != 'none':
        if extension=='url':
            deletepath = os.path.join(folder, "%s_%s" % (filenumber, "url"))
        else:
            deletepath = os.path.join(folder, "%s_%s" % (filenumber, datafile1))

        os.remove(deletepath)

    cursor.execute("insert into upload_index (resources) values (%s)", filenumber)

    db.close()

    # Display conformation screen
    template_str = self.DeleteReport.read_raw()
    template = DocumentTemplate.HTML(template_str)
    theresult = template(self)
    return theresult

def upload_assess(self):

    REQUEST = self.REQUEST
    # Determine what folder to upload the file into
    db = MySQLdb.connect(host="localhost", user="mysheff", db="shef_nle")
    cursor = db.cursor()

    try:
        aurl = REQUEST['aurl']
        identifier = REQUEST['identifier']
    except KeyError:
        results = "Please select a file or URL to upload"
        theresult = template(self, results=results, identifier='')
        return theresult

    #get index number
    cursor.execute("select min(resources), count(resources) from upload_index")
    result=cursor.fetchall()
    nxtfile=result[0][0]
    count=result[0][1]
    okay = 0

    oname = os.path.join(folder, "%s_%s" % (nxtfile, "url"))
    f = open(oname, "wb")
    f.write(aurl)
    f.close()
    content_type = "assessment/caa"
```
filesize=os.stat(oname)[6]
filename = aurl
okay = 1
extension = 'url'
usebrowsertype= 0
results = "URL [%s] has been uploaded" % aurl

if okay:
    results = "Filename %s has been uploaded. It is %s bytes in size" %
    (filename, filesize)
else:
    results = "Sorry, the file was not uploaded. Please contact the system
    administrator."

if okay:
    try:
        self.add_assess(filenumber=nxtfile, filename=filename, filesize=filesize,
                        content_type=content_type,
                        identifier=REQUEST['identifier'],
                        extra_info=REQUEST['extra_info'],
                        keywords=REQUEST['keywords'],
                        extension=extension,
                        user=REQUEST['AUTHENTICATED_USER'].getUserName(),
                        resyear=REQUEST['resyear'],
                        resmonth=REQUEST['resmonth'],
                        resday=REQUEST['resday'],
                        phase=REQUEST['phase'])
        cursor.execute("select counter from upload_index where resources=%s",
                        nxtfile)
        result=cursor.fetchall()
        # if only one index entry increment the value else delete the used entry
        if count == 1:
            nxtfile = nxtfile + 1
            cursor.execute("update upload_index set resources=%s where counter=%s",
                           [nxtfile, counter])
        else:
            cursor.execute("delete from upload_index where counter=%s", counter)

db.close()
except:
    str = "%s:%s\n%s:%s" % (nxtfile, filename, sys.exc_type, sys.exc_value)
    results = "Sorry, there was a problem adding the file uploaded details
    to the database\n%s" % str
else:
    results = "No file was specified"

back_url="http://www.minerva.shef.ac.uk/touchstone"
return REQUEST.RESPONSE.redirect(back_url)
```

**Listing 2: Resource Upload Python Script**
4. Contacts Details

Minerva is developed and supported by the Academic Unit of Medical Education at the University of Sheffield.

Contact details are as follows:-

The University of Sheffield
Academic Unit of Medical Education
1st Floor
Coleridge House
Northern General Hospital
Herries Road
Sheffield
S5 7AU
Tel: 0114 2266418
Fax: 0114 2424896

Unit Web Site: www.shef.ac.uk/~dme

Minerva: www.minerva.shef.ac.uk (restricted)

Minerva web team:-

Ash Self
eLearning Developer
Email: a.self@shef.ac.uk
Tel: 0114 2266766

Richard Davidson
ePortfolio Project Officer
Email: To be confirmed
Tel: 0114 2266082

Emma Hudson
Administrative Support
Email: e.hudson@shef.ac.uk
Tel: 0114 2266418
References

Publications

- General Medical Council (1993) Tomorrow’s Doctors. London GMC.
- General Medical Council (2002) Tomorrow’s Doctors. London GMC.

Conference Presentations

- ASME Annual Scientific Meeting, Edinburgh 2003 Developing an Outcome Focussed Core Curriculum - the Interactive Database.
- ASME Annual Scientific Meeting, Edinburgh 2003 – Managing the redesign of the undergraduate medical curriculum with a MLE - the Sheffield approach.
- UMSLG Conference, Northampton 2003 - MLE Case Study.
- Networked Learning 3rd International Conference, Sheffield - Towards a Managed Learning Environment in Medical Education.
- Spotlight Conference, Sheffield 2002 - Using a MLE to support PBL.
- TLTP3-86 Dissemination Event, Newcastle 2000 - The Sheffield NLE.