DfEE HIGHER EDUCATION QUALITY AND EMPLOYABILITY H.E.Q.E.

Recording Achievement in Higher Education Project
1998-2000

An Internet Based C&IT System For The Development Of Personal & Academic Records For HE

Paul Drummond
Geoff Hammond
Angela Smallwood

THE UNIVERSITY OF NOTTINGHAM
DfEE
H.E.Q.E.
HIGHER EDUCATION:
Quality & Employability

DfEE HIGHER EDUCATION QUALITY AND EMPLOYABILITY H.E.Q.E.

Recording Achievement in Higher Education Project 1998-2000

Final Report of Project Ref: 234/01/1998/99:

An Internet Based C&IT System For The Development Of Personal & Academic Records For HE

A collaboration between the University of Newcastle upon Tyne and the University of Nottingham

Paul Drummond
Geoff Hammond
Angela Smallwood

This project was funded by the Higher Education: Quality and Employability (HEQE) division of the Department for Education and Employment (DfEE). All work was produced under contract with the DfEE. The views expressed in this document are those of the authors and do not necessarily reflect those of the DfEE or any other Government Department.

Executive Summary

1.1. Aim

To design, evaluate and disseminate a robust and simple process for recording achievement in HE:

- facilitated by the application of Communications & Information Technology (C&IT) and accessible on the Internet
- generally consistent with evolving practice in recording achievement in pre-HE and post-HE sectors of education and employment
- to enable HEIs and their constituent departments to construct Personal & Academic Records (PARs) which can be tailored readily to meet individual subject-specific requirements
- to foster a sense of "ownership" of this approach to recording achievement in the HE sector.

1.2. Design Principles

The technological base of the project was provided by the experience of the team in the Faculty of Medicine Computing Centre in the University of Newcastle, who had previously developed a number of C&IT systems to support undergraduate medical students and their tutors.

The core pedagogical principles and the specific content of *Internet-PARs* were drawn from models of Personal and Academic Records (PARs) developed in a range of disciplines at the University of Nottingham as part of the University's PADSHE Project (Personal and Academic Development for Students in Higher Education: 1996-2000). The key features transferred from the paper-based system into the *Internet-PARs* included:

- to provide scope for customisation at any level to foster ownership: institution, faculty, school, individual
- to assist and streamline academic administration
- to support academic partnership between staff and students
- to foster students' skills in managing their own learning
- to enhance students' academic and career development
- to integrate academic processes with other forms of student support in the university

The move into the medium of C&IT brought new opportunities:

- greater flexibility for customisation and ease of updating and redesign
- the possibility of interfacing with universities' central data systems, improving the quality of student data and promoting closer integration of institutions' IT policies for academic and administrative activities
- automatic provision of a PARs framework for every student listed in the University's central data systems
- a flexible, interactive C&IT system to support personal tutoring
- scope to include a wider range of functions to support student reflection on progress and action planning linking across academic and extra-curricular activities
- scope to develop a bank of functions within the institution which could be used by departments on an opt-in basis

1.3. *Extent of the Development*

- *Internet-PARs* has been developed
  - in the context of discussions with representatives of 16-19 education and employers
  - in the light of pilot exercises in a wide range of academic disciplines at the University of Nottingham: Biological Sciences, Environmental Engineering, Economics, Education (PGCE), History & Art History, Mathematical Physics, Medicine, Physiotherapy.

At the end of two years, *Internet-PARs* has fulfilled the original aim by providing an interactive system which:

- simplifies the construction of PARs tailored to individual subject-specific requirements (Section 2.3.5.5.1)
- is capable of interfacing with University Management Information Systems (MIS) either directly or via data warehousing (Section 2.3.2.1)
- is of proven scalability to the entire student data set for a whole HEI
- facilitates and supports recording personal tutoring (Section 2.3.6)
- can be adapted within agreed constraints to suit the specific needs of individual faculties, schools, tutors and students (Section 2.3.5.5)

As of 31 March 2000, *Internet-PARs* requires further development to cater for more developed student reflection and for recording and reviewing extra-curricular activities.

1.4. *Dissemination*

The system has been demonstrated at HE conferences and, by invitation, in many individual universities and organisations including: Birmingham, University College of Ireland at Maynooth, UMIST, Leeds, Northampton, Plymouth, Queen's (Belfast), Warwick, University College London, UfI, BAE Systems Virtual University and even as far afield as Cape Town.

Internet-PARs have impacted upon a wide range of colleagues in HE:

- academics involved in the institutional review of personal tutorial systems
- academic departments reviewing personal tutor systems
- those with a responsibility for preparing for QA subject review or institutional audit
- students' union officers and staff in student services concerned with student support, skills development and careers management
- project leaders and advisory staff in academic careers services
- registrars and C&IT staff concerned with student records, development of transcripts, records of personal development planning etc
- those involved in developing integrated policies for information services.

Outside HE, *Internet-PARs* hold important potential for recording achievement and personal development planning in schools and colleges, in training and in continuous professional development in business and industry, especially where Internet-based processes are being explored.

1.5. *Outcomes and recommendations*

- The majority of students involved in the piloting has recorded positive responses to the prospect of using *Internet-PARs* in the future and is convinced that the electronic medium is the way forward
- Other institutions wishing to customise *Internet-PARs* will need to resource a period of preparation within their institutions to introduce innovative practice, develop and pilot C&IT adaptations and design an implementation strategy

- Training, to include both induction sessions and back-up support, is recommended for both students and staff
- Existing and new obligations, such as those under the Data Protection Act 1998 and the Human Rights Act 1998, should be considered in the planning of any new information systems such as *Internet-PARs*.

1.6. Deliverables

- A website providing illustrative information and an interactive demonstration option for *Internet-PARs* at *http://info-pars.ncl.ac.uk*
- Technical specifications of the *Internet-PARs* shell which can be used as a basis from which further to develop functionality and tools
- Project Report
- CD-ROM of software scripts developed during the project (available as Section 3.7 Appendix 8 of this report or obtainable from the Newcastle University contacts below).

1.7. Continuation

Both the University of Newcastle and the University of Nottingham are committed to institutional take-up of *Internet-PARs* and implementation has begun.

Dependent upon the necessary resources becoming available, further work is planned:

- to provide consultation for other HEIs interested in *Internet-PARs* at any level
- to explore the feasibility of linking with recording achievement in schools and colleges, with the UfI's Learning Log and with continuing professional development in employment.

1.8. Contact information

For further information on the ***Internet-PARs*** project please visit the information pages at *http://info-pars.ncl.ac.uk/* or contact:

Dr Geoff Hammond, Project Director
Director of Faculty of Medicine Computing Centre, University of Newcastle upon Tyne, NE2 4HH (*G.R.Hammond@ncl.ac.uk*)

Dr Angela Smallwood, Project Director (Nottingham)
Director of PADSHE Project, Department of English, University of Nottingham, NG7 2RD (*Angela.Smallwood@nottingham.ac.uk*)

Mr Paul Drummond, **Project Manager**
IT Advisor, Faculty of Medicine Computing Centre, University of Newcastle upon Tyne, NE2 4HH (*Paul.Drummond@ncl.ac.uk*)

Other members of the project team included:

Mr Terry Brown, C&IT Development Officer (Newcastle)
Professor Reg Jordan, Project Leader (Newcastle)
Dr Cliff Litton, Developmental Evaluator (Nottingham)
Mrs Gill Murray, Developmental Evaluator (Nottingham)
Mrs Pauline Rossen, Administrator (Newcastle)
Mrs Heather Sowter, Administrator (Nottingham)

2. Full report

2.1. Review of Aim and Project Objectives

Aim: To design, evaluate and disseminate a robust and simple process, facilitated by the application of Communications & Information Technology (C&IT), which would enable HEIs and their constituent departments to construct Personal & Academic Records (PARs) that may be tailored readily to meet individual subject-specific requirements, which will be accessible on the Internet, and will be 'owned' by the HE sector.

The objectives as set out in the original bid are as follows:

1. To define an agreed, discrete sub-set of personal development and enterprise skills, common to all degree subjects, which will form the basis of the shared generic content of a PAR robust enough to serve the diverse needs of HE
2. To develop a flexible, user-friendly process which will allow different disciplines the freedom to determine the specific academic skills component of a PAR designed to meet their own subject-specific requirements
3. To develop a software shell suitable for delivery on the Internet which will provide a framework within which universities - or more probably faculties and/or departments - could with relative ease construct and test their own PARs. (Years 1 and 2: prototype to be complete by October 1998; final version completed by February 2000)
4. To trial and evaluate use of the software shell in the process of development of PARs at each of the participating universities and to identify the staff development requirements for their construction and use
5. To ensure that PAR development is consistent with other developments in the theme area (e.g. use of NRA at pre- and post-university level)

2.2. Revision of the objectives

Objectives 1 and 2

Objectives 1 and 2 were reconsidered, with the approval of the Steering Committee, in March 1999, as a result of the developing experience of the project. The background to this change is explained here.

Early in the project, a Skills Working Party was formed, consisting of representatives of three schools/departments involved in the project, Economics, Medicine and History and three schools/departments who were developing paper-based PARs, namely Politics, Physiotherapy and Built Environment. Representatives from the other schools in the project were present at the meetings so as to gather the widest possible spread of opinions. This party met three times to consider the question of how to define and develop generic and subject specific skills and competencies for their subjects.

The discussions highlighted a significant divide between schools/departments that offered vocational programmes such as Medicine, Physiotherapy and Built Environment and non-vocational subjects. In the former group, subject specific skills and often generic skills were already quite well defined although not always made explicit to students. Such subjects believed that the PAR system would make these skills more explicit to staff and students alike.

In contrast, it was recognised that in non-vocational subjects there was the distinct need to raise awareness amongst both staff and students of the skills content of the degree programmes on offer. In particular it was felt that until this awareness had been raised it would be difficult to define generic skills that would be meaningful to different subjects. At this time advice was sought and obtained from members of Key Skills projects but this confirmed the difficulty.

It was also recognised that there was a great reluctance amongst some non-vocational subjects to engage meaningfully with the issue of skills, particularly generic skills. However it was noted that the current thrust towards benchmarking and the development of programme specifications, in which learning outcomes of degree courses would be expressed in terms of subject specific skills and generic skills, would do much to focus the attention of such subjects.

In the light of these discussions the working party recommended that the *Internet-PARs* system should include only a basic means of capturing the skills element in a student's experience and that this should be defined in the language of that subject so as to gain the support of academic staff.

This would mean that each subject should be able to define and input skills encountered within their degree programmes in the IT system and that the students should be able to record and reflect upon their experiences of these skills. To do anything more sophisticated would almost certainly mean carrying out a skills analysis for each subject and this was thought to be beyond the remit and resources of the current project.

Objective 5

Under 'Methodology' the contract envisaged convening a single working group to determine the outline needs of employers and schools. In practice it became essential to treat the two groups individually in order to set up worthwhile meetings at viable times.

The following report provides project results and commentary for the revised objectives, as explained above, under the following headings:

1. Developing the *Internet-PARs* shell
2. Trialling and evaluating the software
3. Determining the outline needs and views of (i) schools (ii) employers
4. Institutional take-up and wider dissemination.

2.3. Developing the Internet-PARs shell

The main concern of the whole development team was that the technological solutions adopted for the project should facilitate the pedagogical processes that underpin reflective learning, and place as few constraints on those processes as possible. To meet the project aim, the major design goal was to create a flexible framework capable of: (1) supporting student data sets ranging in size and complexity from a small departmentally based degree programme to that for a whole HEI (i.e. scalable); (2) meeting security/data confidentiality concerns; and (3) integrating with institutional management information systems. The framework should also be compatible with emerging 'standards' for interchange of data to support management of learning.

As a major part of this project was about appropriate application of C&IT, this report includes a necessary level of technical detail. In the interests of readability, the main body of this report includes the general approaches and issues likely to be of interest to senior managers considering implementation of PARs at the institutional level. Further, more detailed technical information, likely to be of interest to information management and computing services staff tasked with implementation of such systems, has been incorporated into a number of appendices.

2.3.1. A specification for *Internet-PARs*

The initial C&IT task was to define the specifications of an environment that would be suitable for the development of an electronic PAR. The term *environment* is used to encompass: the type of computer, operating system, and network protocol on which the project would be hosted and more importantly the approach to software development.

In addressing objective three of the project (Section 2.1), it was important that any electronic PARs system developed should be scalable to the institutional level and that appropriate quality assurance should lie at the heart of the C&IT approach adopted. This approach effectively ruled out a system where different academic departments/faculties develop their own electronic PARs storing different amounts of data in different formats. The project team wanted an institution to be able create a variety of aspects in the recording process which would operate for all departments while allowing each department/faculty to retain their own ways of introducing and discussing each aspect. The resulting aspect would have to be able to be included or excluded from a department's PAR (or even a PAR designed for a specific programme of study) and potentially have its appearance altered based on that department's preferences. Accordingly, the software development environment (Section 2.3.1.1) would not only have to support this anticipated degree of flexibility, but be capable of evolving in response to feedback from the pilot groups developing PARs at the departmental level.

2.3.1.1. The development environment

The team chose the UNIX platform as a base for the development of the software, since at the time the project was started (April 1998) it was felt that this was the most suitable multi-user platform available. The operating system used was Solaris 2.6 supplied by Sun Microsystems, who also supplied the computer, an Enterprise (E450) Server, which hosts the *Internet-PARs* for this project.

The software shell, which provides the framework in which to construct *Internet-PARs*, was developed using predominantly unlicensed or free software in order to minimise the cost incurred by any institution wishing to use or further develop the shell. This software is acquired over the Internet in the form of "source code", which must be compiled, or assembled on the specific computer on which it will be run. The function of the compilation process is to provide software that is finely tuned to the machine on which it will operate, optimising the speed of the software and minimising errors caused by software/hardware conflicts. An additional advantage of the process is that it allows links to be formed between database, programming language, and web-server as described below.

The database used to store the institutional and PAR specific data is MySQL (TCX DataKonsult, Sweden)[1], a robust, high performance database using the generic Structured Query Language (SQL). Generic SQL, as used in this project is governed by international standards[2] and is "understood" by the majority of commercial databases, creating a highly portable system capable of running on the majority of platforms. MySQL was chosen for the project because, it is both a powerful and widely used database and, unlike competitors such as Oracle or Microsoft's SQL Server, it carries no licensing implications and is free to use in non-commercial or educational ventures.

The programs providing the functionality of the shell were written in PERL (Practical Extraction and Reporting Language), a common Internet development tool. PERL is free to use[3] and widely supported in the Internet development community. The version of PERL used for the project (PERL 5.0) was compiled with the built-in PERL/MySQL interface provided with the MySQL software. If institutions using alternative database solutions wished to use *Internet-PARs*, they would have to compile PERL with the appropriate interface for their preferred databases.

The final component of the development environment is the web-server which delivers the *Internet-PARs* functionality to its users. The *Internet-PARs* shell is currently delivered to the

[1] *http://www.mysql.com/*

[2] ISO/IEC 9075:1992, "Information Technology --- Database Languages --- SQL"
ANSI X3.135-1992, "Database Language SQL"

[3] *http://www.perl.com/*

Internet by an Apache web-server (Apache 2.6.3)[4], which was compiled with built-in support for PERL to maximise speed and efficiency for users of *Internet-PARs*. Apache was chosen for the project because it is a robust web-server, widely used to deliver scalable services (e.g. Microsoft's popular Hot Mail service) and is free to non-commercial ventures.

While all development to date has been carried out in this environment, recent investigation has shown that with minor modifications to the scripts, the pilot work should transfer readily to the increasingly popular Microsoft NT environment. However, it should be noted that changing to an NT environment may incur licensing costs for the MySQL database[5].

2.3.1.2. The software shell approach

Software shells are frameworks within which developers can create their own applications/content for delivery to end-users. The intention behind the use of shells is to minimise the amount of technical skill required to develop such applications.

Shells operate at two levels:

- **End-user shells**: complete programs into which the developer only has to enter text based content. These shells were widely used in the multimedia industry in the early 1990's
- **Development shells**: frameworks to assist the rapid development of software in a known environment. These can vary from additions to an operating system (system shells) which simplify programming in that environment, to specialised development environments such as that used in the *Internet-PARs* project.

The *Internet-PARs* project makes use of both types of shell. The shell forming the core of the *Internet-PARs* system is a development shell formed from a combination of data structures and libraries of programming code. The libraries of code shorten the process of creating new tools (here defined as discrete units of functionality within a software application) by providing developers with simplified methods of performing complex programming tasks. Once a tool has been created and described to the shell (Section 2.3.5.5.1) it automatically falls under the control of the accessibility and configuration options included in the development shell. The data structures of the development shell, full copies of which can be found in Section 3.2 Appendix 2, are used to describe tools, system appearance, institutional structure, etc. and are discussed in greater detail later in the report.

The tools developed using the libraries and data structures of the development shell frequently take the form of end-user shells, with administrators able to alter the text based content of various areas within the tool (Section 2.3.5.5.2).

2.3.2. Developing an appropriate data model

For the shells described above to operate in a truly institutional manner, the system must contain a realistic model of the relationships between faculties/schools, departments, courses, modules, tutors, students, etc. which exist within an HE establishment. The most efficient method of storing this material is in a database, which can be simply defined as a collection of data that is organised in such a way that its contents can easily be accessed, managed, and updated. The most common type of database used in this environment is the relational database, a tabular database in which data is defined so that it can be efficiently reorganised and accessed in a number of different ways.

Databases have long held a key role in the administration of student learning in the form of Management Information Systems (MIS) which have been used by institutions to control the registration, placement and financial affairs of students within an institution. As their current function suggests, MIS systems have not traditionally been associated with the delivery of education, rather they are designed to deal with the management and financial aspects of the

[4] *http://www.apache.org/*

[5] ***http://www.mysql.com/Manual_chapter/manual_Licensing_and_Support.html***

educative process. However, the data they contain describing the complex relationships of students, (academic) staff, modules and courses is the natural base for any system intended to provide support for the learning process.

2.3.2.1. Interface with Management Information Systems

The project has undertaken work with both Newcastle's commercial MIS systems (MAC and SAP) and Nottingham's bespoke MIS system (SATURN) in order to simplify or automate the access to the data described above. Demonstrations of the *Internet-PARs* shell and discussions with senior staff at Newcastle's Management Information Development Service (MIDS) have led to the extension of existing "data warehousing" arrangements with the Faculty of Medicine Computing Centre (FMCC).

Data warehousing describes the process by which MIS data owners give permission for their system administrators to schedule regular data dumps to a secure area on the institution's networks. This data dump occurs at pre-arranged times and the data need only be left there for very short periods of time before it is retrieved by the project's computers which will incorporate it into their own data structures. For this project, the interval between data dump and retrieval has been of the order of a few hours, although this period could be reduced to a few minutes by closer co-ordination of the two processes. The new data either replaces existing data, or the two data sources are integrated. By this method, data, which may represent only a fraction of the total information held in MIS systems, can be made available to educational systems such as *Internet-PARs* without the security risks posed by allowing full MIS access to students and academics.

Automated warehousing has been established at Newcastle where data from the MAC MIS system is exported to a secure network location from which it is picked up by the FMCC for use in its student support systems. (Due to the developmental nature of the *Internet-PARs* project's data structures, the process of integrating the two data sources has not been automated at the current time. However, other projects which use the same data source illustrate the feasibility of developing small programs which control the integration of the data.)

Similarly, demonstrations of the *Internet-PARs* shell and discussions with senior staff in Nottingham's Corporate Information Systems (CIS) division have resulted in valuable collaboration. As one outcome, export queries have been built into the Nottingham SATURN system to produce the required data for Nottingham's *Internet-PARs* system when run. Continuing discussions indicate that in the longer term, as part of Nottingham University's commitment to the use of PARs, the CIS team would favour a live interface between *Internet-PARs* and SATURN rather than the warehousing model currently used at Newcastle.

The following information within the Newcastle and Nottingham *Internet-PARs* shell is derived directly from MIS data:

- **Student data:** student number (acts as a unique identifier[6]), name, address, email, department, course, modules, pastoral tutor
- **Staff data:** payroll number (acts as a unique identifier), name, email, department, modules, pastoral tutees
- **Institutional data:** faculties (name, code), departments (name, code, faculty), courses (name, code), modules (name, code, assessment method, credits).

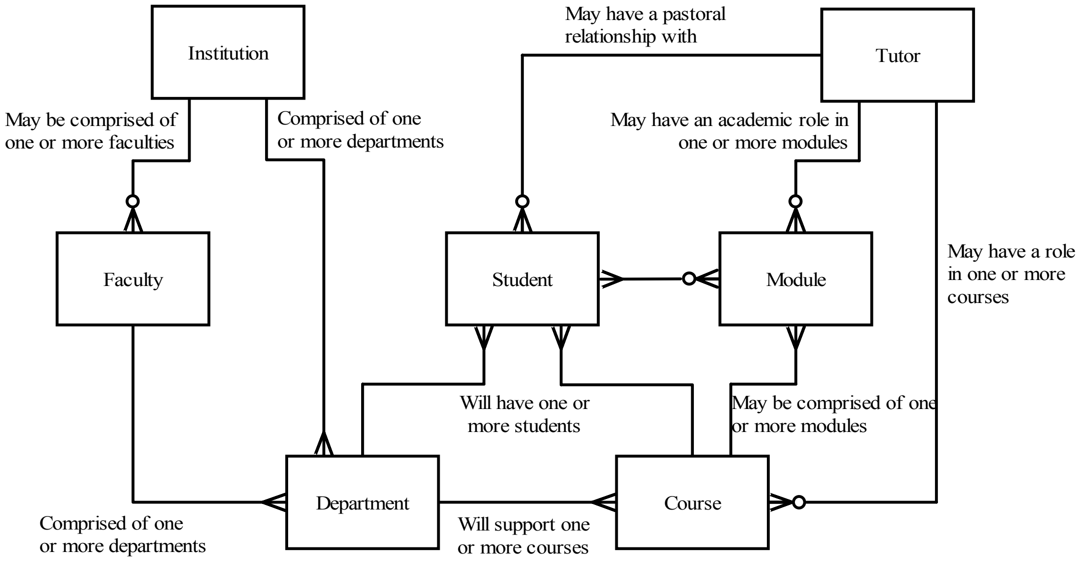
2.3.2.2. Working with MIS data

The Internet-PARs project makes use of the sub-set of MIS data described above to create its own model of the relationships between students, tutors, faculties, departments, modules and

[6] Identifiers other than student number or payroll number could be used by an institution as a unique identifier (e.g. email identifier). However, this would need to be balanced against the potential for compromised security implicit in using a more generally accessible unique identifier.

courses. A simple representation of these relationships can be seen in Figure 1 below, with the full entity relationship diagram presented in Section 3.2 Appendix 2 of the report.

Figure 1: Simplified illustration showing the relationships in the institutional data derived directly from the Newcastle and Nottingham MIS systems as used by Internet-PARs


Institution
May have a pastoral relationship with
Tutor
May be comprised of one or more faculties
Comprised of one or more departments
May have an academic role in one or more modules
Faculty
Student
Module
May have a role in one or more courses
Will have one or more students
May be comprised of one or more modules
Department
Course
Comprised of one or more departments
Will support one or more courses

The relationships derived from the MIS data provide the *Internet-PARs* system with the information it requires in order to represent both the relationships between individuals and the hierarchical relationships which, taken together, govern the behaviour of tools created within the shell. While this division into individual and hierarchical relationships is an artificial one, it makes it easier to describe the functioning of the shell for this report.

As can be seen in Figure 1, the predominant relationships are between students and tutors. These can be complex relationships: pastoral; academic; advisory (as with the careers service); or purely hierarchical, such as a student’s relationship to their Head of Department. The hierarchical relationships describe both the relationships between faculties, departments, courses, and modules, and the relationships between these entities and the staff and students who are either employed by or enrolled with them.

Relationships between individuals control much of the functionality of *Internet-PARs* shell and tools. Hierarchical relationships control the access to and appearance of tools created within the *Internet-PARs* shell for any given user. This relationship between the *Internet-PARs* data model, its functionality and access control mechanisms is discussed in greater detail in Section 2.3.5.5.3.

2.3.3. Data protection and data security

The data derived from the MIS systems of both institutions, and the relationships it describes are a crucial component of the *Internet-PARs* development shell. However, holding data about students, staff, and the institution itself will have implications for HE institutions which wish to develop the concept further, not least in relation to the provisions of the Data Protection Act 1998. This Act, along with other relevant legislation[7], will have a significant effect on the way in which higher education institutions store and process information about living people. This in turn has implications for the use of *Internet-PARs* and other such student support and administration systems within an institution.

[7] *Copyright, Designs and Patents Act 1988*; *Data Protection Act 1998*; *Human Rights Act 1998*; Draft Freedom of Information Act; Copyright and Rights in Databases Regulations (1997) and Intellectual Property Rights (Customs) Regulations (1999); Copyright and Licensing Agency guidelines.

For the moment, the Data Protection Registrar has agreed to allow higher education to define its own policies within the Act. This is being followed up by CVCP[8], JISC[9] and other organisations. However, it is reasonable to assume that certain obligations under the Act will be universal. These obligations, the most relevant of which are described below, should be considered in the planning of any new information systems such as *Internet-PARs*.

Being open about what data is processed

Under the provisions of the Act, anyone can request the Data Controller of an organisation to provide documentation identifying all data which is held about them, and copies of the actual data held.

The most notable effect of this provision is in relation to the use of the *Internet-PARs* by staff to record notes or comments about students in relation to either academic performance or pastoral issues.

Relevance and currency of data

The provisions of the 1998 Act require the Data Controller to destroy data which has no present value or purpose. For example, references provided by previous employers are often held 'on-file' for years without apparent justification. The 1998 Act requires that such materials be destroyed. This requirement should not have a direct effect on the project since information stored by students and staff within the *Internet-PAR* is always held in relation to a specific function. However, this requirement may have significant implications for a university or other organisation involved in the development of life-long records of learning.

A potential threat to all new information systems is the provision for data subjects to request the removal of data from a system which relates to themselves. While this is essentially an institutional issue, it may be that such requests are triggered by the presence of highly visible systems such as *Internet-PARs*. From the perspective of projects like *Internet-PARs*, this is likely to be one of the hardest parts of the 1998 Act to comply with, as many institutions do not yet have policies or procedures in place on which to rely.

Providing Adequate Security for Data

The Data Protection Act 1998 obliges data owners to provide adequate security to protect data held about a living person. 'Adequate' refers in part to the value of the data. Sensitive personal data may have to be more heavily secured than personal data. For example, normal security would include screen locks and passwords on computers, complete erasure when disposing of old computers, locks on doors and filing cabinets to prevent non-authorised access to paper records. The Act suggests that highly sensitive records, such as health records, should also be encrypted to a suitable standard to prevent access by unauthorised persons.

While health records undoubtedly fall into the category of highly sensitive data, it is less clear whether a student's academic and pastoral records would constitute highly sensitive records. The project has taken this into account in its response to the Act (see below) and in its exploration of models of data ownership, e.g. some areas of the *Internet-PARs* are private to the individual student.

2.3.3.1. *Internet-PARs* response to the Act

While HE policies in relation to the 1998 Act may not yet be completely defined, HE should be endeavouring to implement good practice wherever possible. This means implementing policies that respect the rights of data subjects, proactively giving access to information concerning the data which is held about them, and providing examples of that data. This includes implementing confidentiality and security policies and staff development to raise awareness of data protection. The *Internet-PARs* team has endeavoured to abide by the Act by making the following provisions:

[8] *http://www.cvcp.ac.uk/*

[9] *http://www.jisc.ac.uk/*

Informed consent and declaration of data held

All students and staff who participated in the pilot groups at Nottingham signed forms (held at Nottingham) indicating their consent for their data to be stored and used as part of this project. These forms explained the project to the participants, describing what the data would be used for, who would have access to it, and how long it would be stored. Students at the Faculty of Medicine in Newcastle University, whose data has been used to trial the PAR at Newcastle, already complete such a consent form in relation to other student support systems in use within the Faculty.

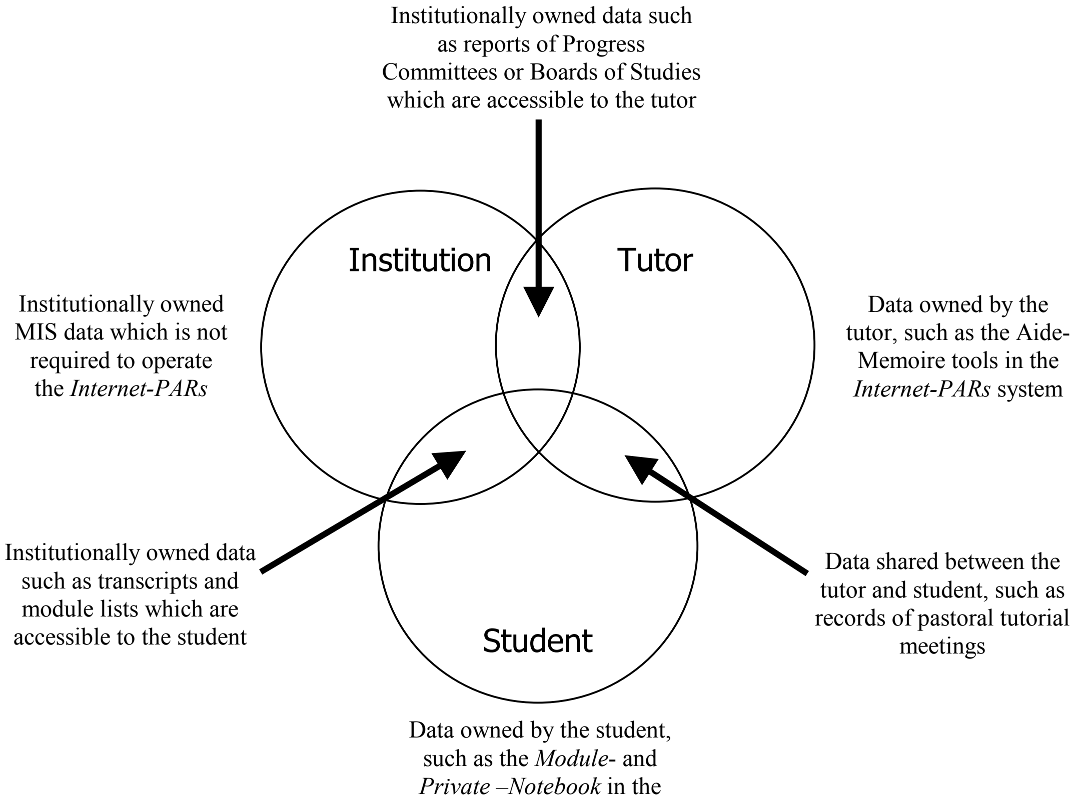
All data held within the *Internet-PAR* relating to Newcastle students has been declared to the University's Data Protection Officer in compliance with the University's policy on Data Protection, and in conjunction with other student administration projects making use of the same data source.

Examples of the data registration form and data consent statement used at Newcastle are included in Section 3.4 Appendix 4 of the report. In addition, an example leaflet, for use with students in the Faculty of Medicine at Newcastle University has been included. However, it should be noted that other institutions wishing to use such documentation will need to develop their own versions in conjunction with their own Data Protection Officer.

Confidentiality and security of the data

Confidentiality of data within *Internet-PARs* is provided by authentication processes based on an agreed model of data ownership (Figure 2). The model allows that for the three major categories of data owner (student, tutor, and institution), data held in the PAR will be either private, shared, or public. All access to the *Internet-PARs* system is based on this model with password protection ensuring that a user may only see private data which relates to them or data to which they have shared access. Students have access only to their own data, and tutors only to data relating to themselves and their own students. During the development of new tools within the *Internet-PARs* shell, members of the pilot groups at Nottingham were consulted to determine which data belonged in which category of ownership.

Figure 2: Showing the concept of shared ownership on which access to information stored in the Internet-PAR is based



Institutionally owned data such as reports of Progress Committees or Boards of Studies which are accessible to the tutor
Institution
Tutor
Institutionally owned MIS data which is not required to operate the *Internet-PARs*
Data owned by the tutor, such as the Aide-Memoire tools in the *Internet-PARs* system
Institutionally owned data such as transcripts and module lists which are accessible to the student
Student
Data shared between the tutor and student, such as records of pastoral tutorial meetings
Data owned by the student, such as the *Module-* and *Private –Notebook* in the *Internet–PARs* system

Data security is also provided in the form of computer hardware and network restrictions on access to the computer on which the data is held. Sun Microsystems provided time from their technical support team to set up the firewall systems they recommended to the project at its inception. This firewall configuration operated for the duration of the pilots at Nottingham University. The firewall prevented all off-site access to the computer on which the data was stored by any method other than the http web protocols required for legitimate access. It also restricted the functionality allowed to computers at Newcastle University, with the exception of the developers' machines accorded the status of "trusted clients".

To complete their commitment to the project, Sun Microsystems has been asked to return to Newcastle to recheck the security of the system and reconfigure/update the firewall in light of significant changes in both firewall technologies and local changes in system architecture at Newcastle. Any significant changes resulting from this consultation will be disseminated via the project information site.

2.3.4. Embedding flexibility in the *Internet-PARs*

Investigation of the existing paper-based PARs in use both in Nottingham and Newcastle Universities as part of the PADSHE project indicated that while most departments were using a standard design, there were several areas in which departments needed to be able to "customise" the basic PAR. These areas were identified as:

- **Terminology**: certain terms and expressions used within the paper-based PARs, eg tutorial meetings connected with a student's personal development could be referred to as "personal development meetings", "pastoral tutorials", "personal development planning meetings", etc.
- **Introduction**: departments introduced processes within the PAR in different ways according to the way they would be used in that department
- **Purpose**: closely linked with the introduction to a process within the PAR, departments frequently used more expansive text in order to contextualise the use of a particular process
- **Documentation**: it was felt that supplementary documentation should be able to be added to any of the three main processes within the PAR.

This degree of flexibility is delivered through the PAR-specific elements of the data model in combination with the scripts controlling the functionality of the PAR. The customisation process is described in greater detail in Section 2.3.5.5.2 below.

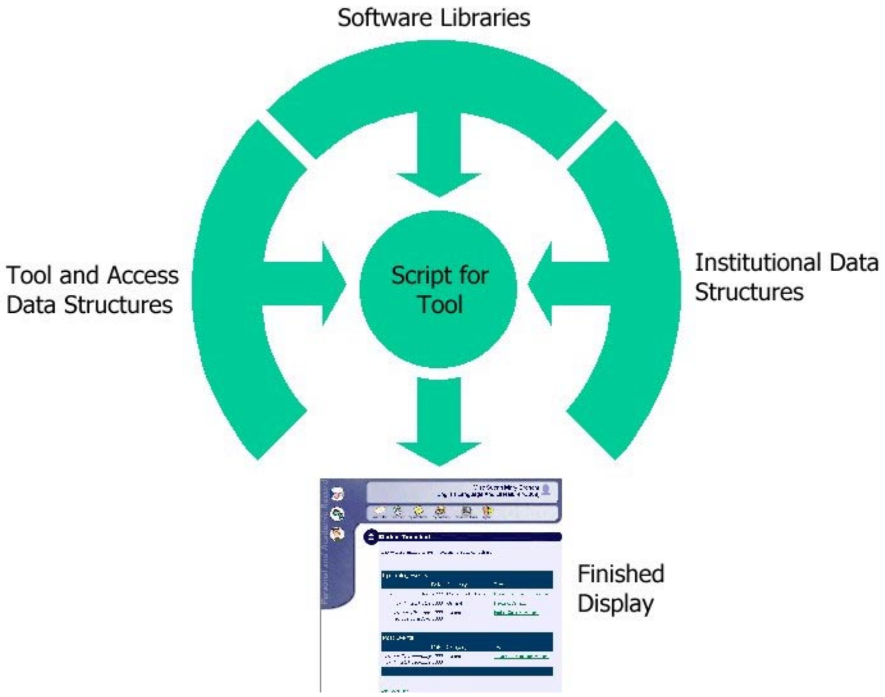
2.3.5. The *Internet-PARs* shell

In order for the concept of a customisable resource to work at the scale of the institution it is essential that all aspects of the customisation process should be an integral part of the electronic PAR that is developed.

The *Internet-PARs* shell has to be capable of recognising the position of the user within the organisation and reacting accordingly. This means in effect that either each tool built within the PAR has to include the complex programming required to support this, or that recognition and response to individual users should be part of a supporting shell in which tools can be developed.

The shell that was developed for this project is based around a combination of software libraries and data structures describing the institution in which the PAR is running. Individual functions in the *Internet-PAR*, referred to as tools, call on these components of the shell to create objects describing both the user and the tool itself. These objects are then used to create the customised on-screen display which is seen by the user (Figure 3).

Figure 3: Relationships between the components of the shell, scripts, and the on-screen display generated by the Internet-PARs shell.


Software Libraries
Tool and Access Data Structures
Script for Tool
Institutional Data Structures
Finished Display

The components of the shell are described in more detail below and their interaction in the creation of an exemplar tool is followed.

2.3.5.1. Software libraries

Software libraries act as compendiums of frequently used functionality within the *Internet-PARs* shell. They span simple functions, such as converting a date from numeric (05/11/00) to text (5th November 2000) format, through to highly complex functions, such as handling user authentication against the database. In most high level programming languages, when an aspect of functionality is to be used more than once in a program, rather than duplicating the code which creates that functionality, it can be parcelled up as a procedure, given a unique name, and invoked whenever required. The software libraries of the *Internet-PARs* shell take this process one step further by identifying those areas of functionality which are common to more than one program in the shell. These are then gathered together in a single location related to that generic aspect of functionality.

The software libraries currently supporting functional areas include:

general_lib.pl *mysql_lib.pl* *search_lib.pl* *html_lib.pl* *object_lib.pl*
department.pl *course.pl* *module.pl* *student.pl* *tutor.pl*
meeting.pl *time.pl* *admin.pl* *mail.pl* *tool.pl*
glossary.pl *cookie.pl*

(Full listings of these software libraries can be found in Section 3.1 Appendix 1 of the report.)

As well as simplifying the creation of new tools within the *Internet-PARs* shell (Section 2.3.5.5.1) this method of working also assisted the development team in responding rapidly to requests from the pilot groups at Nottingham. Since the library code existed in only one part of the *Internet-PARs* shell, modifications made in a single section of code were effective wherever invoked in the entire system.

2.3.5.2. PAR specific data

The *Internet-PARs* shell is reliant on two main data models: (1) institutional data, describing the relationships between staff, students, and the departmental structure of the institution (Section 2.3.2), and (2) data that is specific to the function of the *Internet-PAR* itself. This specific data contains information about the configuration and behaviour of tools within the *Internet-PAR* at all levels of the institution.

The use of this data is best illustrated by reference to Section 2.3.5.5 describing the creation and subsequent customisation of a new tool within the *Internet-PARs* shell.

2.3.5.3. Objects

As described above, the project team has taken an object-based approach to the development of the *Internet-PARs* shell. Initialisation scripts in the code-library of the *Internet-PAR* create objects for each entity the program encounters.

As an example, when a student logs into *Internet-PARs* an object is created based on that student's unique identifier and password. The "student" object holds information on the student themselves (unique identifier, name, email address, postal address), the course they are studying (code, name, their current academic year, all modules they have studied, current modules), and their relationship with the department (department, tutors). Once created, this data is immediately available to the tools within the PAR without repeated interactions with the underlying database.

The extent of the content of the objects used in the *Internet-PARs* shell has been defined over the course of the project based on analysis of data most frequently required by the tools developed for the pilot.

Full details of the "student", and all other objects can be found in Section 3.3 Appendix 3 of the report.

2.3.5.4. Tools

The expression "tool" is used to refer to a discrete area of functionality within the PAR. Tools range from static displays of text such as those used in the on-line supporting documentation, to highly interactive pages through which users can modify the content of the underlying PAR database. Within the development shell an individual PAR is created as a series of these tools which in the case of pilot *Internet-PARs* have been linked together to describe a process-driven view of student and tutor activities.

Most tools currently available in the pilot version of *Internet-PARs* take the form of programs written in the language PERL. These programs interact with the sections of the data structure describing the institution in which the PAR is running (Figure 1, Section 3.2 Appendix 2). This creation process is greatly assisted by the use of software libraries which have been created within the development shell of the IT-PAR (full listings of these libraries can be found in Section 3.1 Appendix 1).

2.3.5.5. How the shell components interact to simplify the creation of a tool

The most effective way to discuss the operation of the shell is to follow through the development of a new tool within *Internet-PARs*. Using this tool as an example, we can follow the process of creating and customising tools within the *Internet-PARs* shell and then look at how the shell functions as a user calls up the tool we have created.

2.3.5.5.1. Creating a new tool in the *Internet-PARs* shell

The process of creating a tool begins with writing its script or program. It is the script for a tool which determines its function and acts as the link to the software libraries and data structures comprising the shell (Sections 3.1 and 3.2, Appendices 1 and 2). All tools in

Internet-PARs take the form of scripts which can contain either complex programs that will interact with the user, or simple HTML formatted text to be displayed on screen.

To simplify the discussion, the new tool we create will have no built-in functionality, rather it will act as a placeholder to illustrate the interaction of the various shell components. The script of our new tool, shown in Table 1, represents the code required to produce a new tool and will form the basis for the following section.

Table 1: Showing the minimal amount of programming required to create a tool within the Internet-PARs shell. (The # symbol indicates the beginning of a descriptive comment in the program)

```
#!/usr/local/bin/perl                           # 1. Establish the location of the script language
require("../../lib/general_lib.pl");            # 2. Make a link to the libraries in the shell
initialise("student", "about me", "New Tool");  # 3. Initialise the tool
display_html_header( $TOOL{name_glossary} );    # 4. Display the title (checking glossary first)
display_user_details( );                        # 5. Show the users details
display_html_footer("", 1);                     # 6. Display toolbar and end the tool page
```

If we follow the script line by line it is possible to see how the shell components interact.

1. This line is mandatory in all scripts and simply acts to tell the web-server the language in which the script is written and where to find the programs it needs to run scripts written in this language.
2. The “require” statement in makes an explicit link between this tool and the control program for the software libraries called general_lib.pl (see Section 3.1 Appendix 1). Once this link has been made, all of the functions of the libraries are available to this tool. All lines beyond this point are calls to the software libraries of the *Internet-PARs* shell.
3. “Initialise” is a routine in the software library which creates the required objects, based on the type of tool and user. In this instance it will create an object for the student accessing the tool and for the tool itself. Full descriptions of all objects used in the *Internet-PARs* shell can be found in Section 3.3 Appendix 3.
4. This command is a call to the HTML software library requesting the script to generate the header for the tool (Figure 4). The name of the tool is passed to it in the form of a variable ($TOOL{name_glossary}) created as part of the initialise statement in the previous line. This routine will also check if any words or expressions used in the name of the tool have an alternative entry in the glossary for the faculty, department, or course to which the student user belongs.
5. This display command returns the users details from the “student” object that was created in the initialisation routines printing it to the screen in a standard format.
6. The final line of the script again calls the HTML script library which requests the script to generate the footer for the tool (Figure 4). The icons making up the student tool bar at the foot of the page are generated along with the footer.

As an indication of the efficiency of using the routines in the software library, if the software libraries did not exist, the script required to generate this single tool would be over 560 lines in length.

Once the script has been written, the tool’s inclusion in the *Internet-PARs* is dependent on an entry in two tables in the database (TOOL_general, and AUTH_access_rules). These entries inform other applications in *Internet-PARs* of the tool’s category, physical location, default name, icon, etc. These entries are shown in Table 2 and the structure of the database tables to which they apply can be found in Section 3.2 Appendix 2.

Table 2: The two database entries required to enable the tool to be visible as part of an institutional PAR.

```
insert into TOOL_general values ("student","about me","New Tool","Test of a new Tool","/cgi-bin/student/academic/new_tool.pl","academic.gif","");

insert into AUTH_access_rules values ("institution","","student","","","about me","New Tool","","Y");
```

Currently, entering a new tool into these database tables is a manual operation performed by interacting directly with the database using the Structured Query Language (SQL), and is only performed by the system developers. The entries shown above have the following purpose:

The first entry makes the new tool visible to other applications within the *Internet-PAR* describing the:

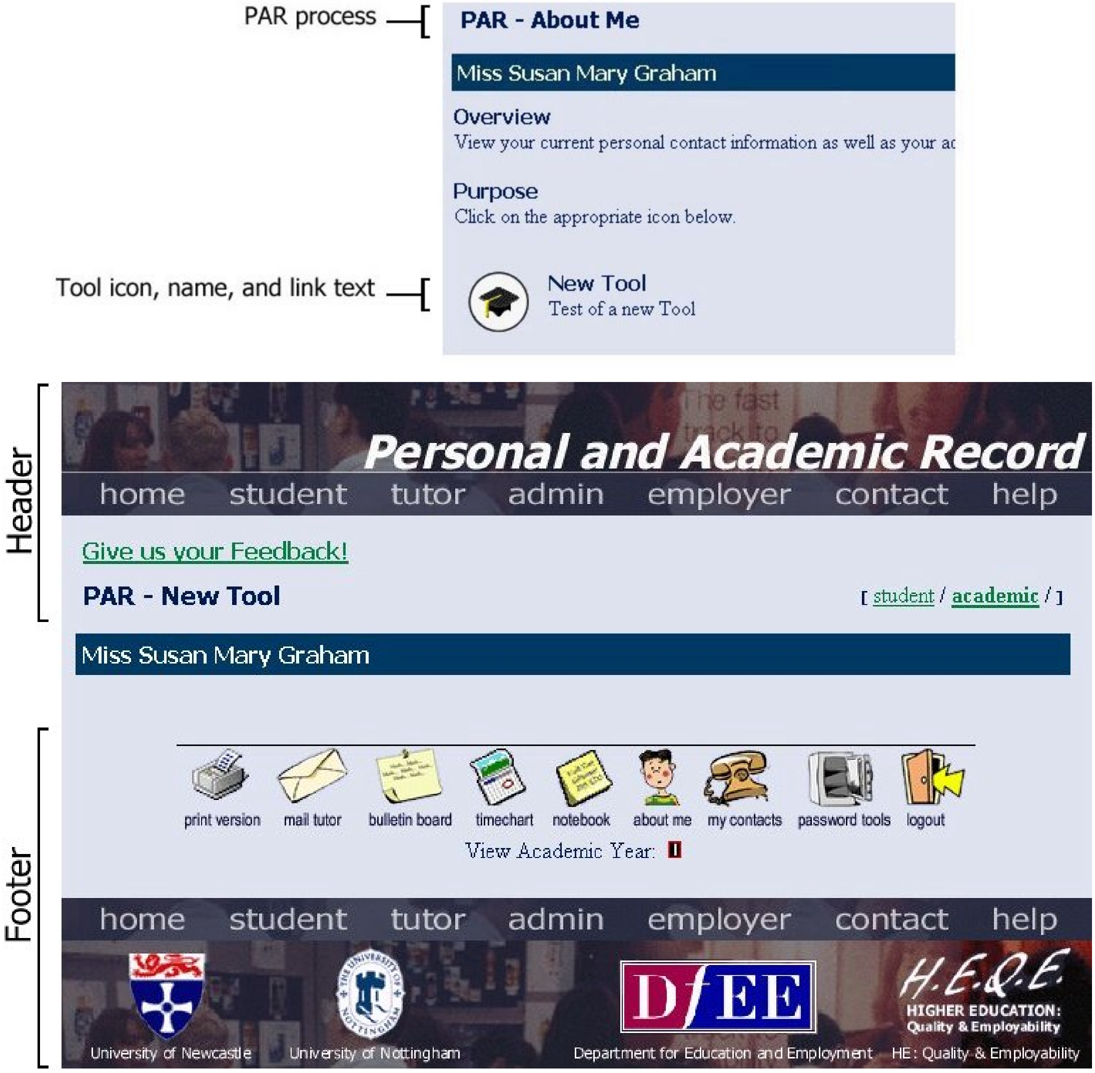
- category of user who will be able to access the tool (“student”)
- process within the PAR to which the tool belongs (“about me”)
- default name of the tool, to be used if no customised name, or glossary entry is found for the faculty, department, course, or module to which a student user belongs (“New Tool”)
- default description used, in the absence of a customised entry, for any links to the tool within the shell (“Test of a new Tool”)
- location of the script so that the shell knows where to call it from (“cgi-bin/student/academic/new_tool.pl”)
- the icon to be used for any links to the tool within the shell (“academic.gif”).

The second entry allows users of the *Internet-PAR* to access the new tool. It creates a master-level access rule describing the:

- level at which this rule will apply (“institution”)
- category of users to which the rule will apply (“student”)
- process within the PAR to which the tool belongs (“about me”)
- default name of the tool (“New Tool”)
- whether access is being allowed or denied to this category of user at this level in the institution (“Y”).

With these entries made in the database, the new tool will now appear in all lists of tools for this category (about me) and in the configuration menu of the Administrative section of the *Internet-PAR*. The appearance of the tool at this stage is shown in Figure 4.

Figure 4: The new tool as it appears after the above operation in the list of tools for this process (top), and as it appears to the user (bottom). The text at the left of the image indicates which parts of the tool are created by the database entries and the HTML script libraries



PAR - About Me

Miss Susan Mary Graham

Overview
View your current personal contact information as well as your ac

Purpose
Click on the appropriate icon below.

New Tool
Test of a new Tool

Personal and Academic Record

Give us your Feedback!

PAR - New Tool [student / academic /]

Miss Susan Mary Graham

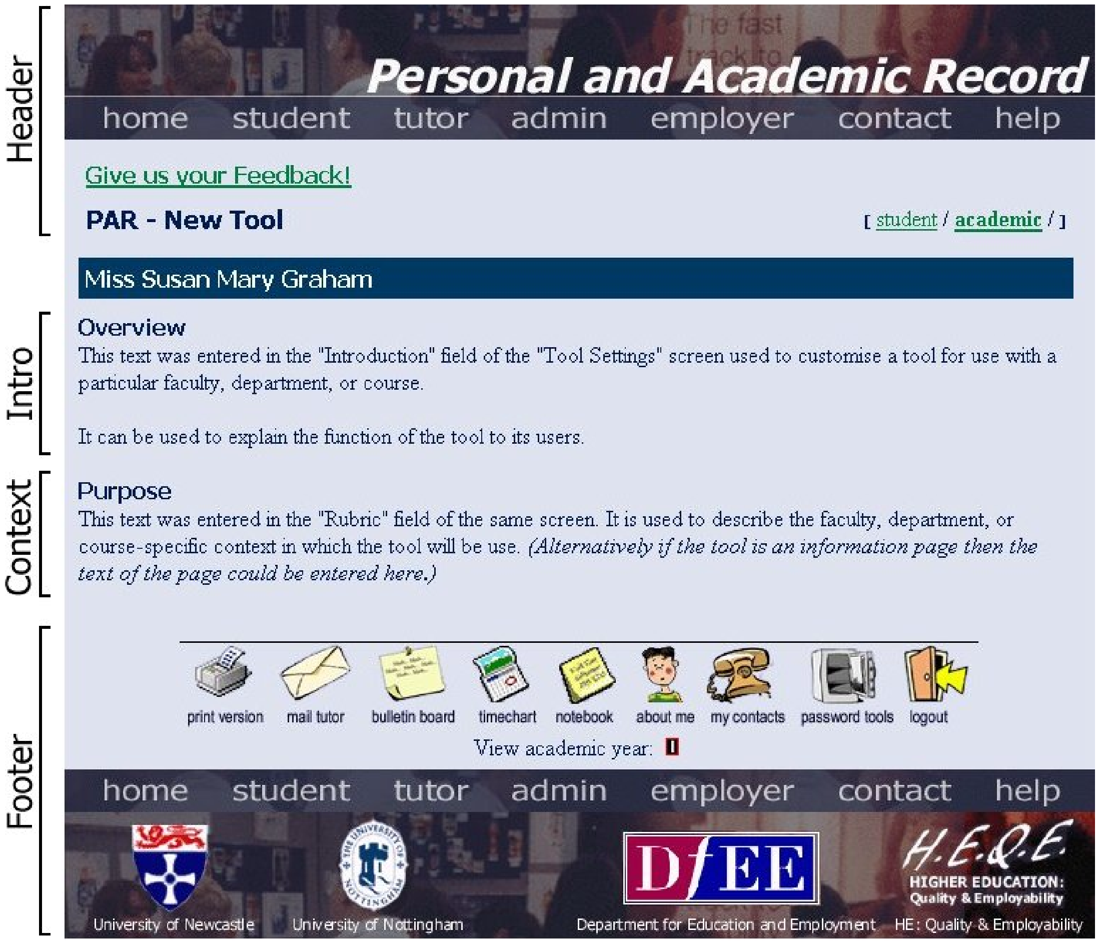
View Academic Year:

2.3.5.5.2. Customising the new tool

Once a default tool has been created, the shell allows for its customisation by administrators of the PAR at all levels of the institutional hierarchy. This is facilitated by a tool in the administrator section of the *Internet-PARs* shell (Section 3.1 Appendix 1 and "Admin" menu item in Figure 4) which uses a simple web form to allow an administrator to change those aspect of the tool described in Section 2.3.4. Because this form, like all other tools in the PAR, is aware of the level of the institution with which the administrator is associated, the changes they make will only affect this level and any subordinate levels in the organisation. Hence a change made at the level of the institution (highest level in the hierarchy) will affect all users of the PAR, whereas a change made by a departmental administrator will affect only that department and the courses it offers.

This customisation process is supported by three data tables (TOOL_name, TOOL_introduction, and TOOL_rubric) which allow a faculty, department, course, or individual to store their own descriptive text for the elements shown, as illustrated in Figure 5. These data tables are described in full in Section 3.2 Appendix 2.

Figure 5: The new tool with introduction and context entered using the administrator's tools


Header
Intro
Context
Footer
The fast track to
Personal and Academic Record
home student tutor admin employer contact help
print version
mail tutor
bulletin board
timechart
notebook
about me
my contacts
password tools
logout
home student tutor admin employer contact help
University of Newcastle
University of Nottingham
DfEE
Department for Education and Employment
H.E.Q.E.
HIGHER EDUCATION: Quality & Employability
HE: Quality & Employability

Give us your Feedback!

PAR - New Tool

[student / **academic** /]

Miss Susan Mary Graham

Overview

This text was entered in the "Introduction" field of the "Tool Settings" screen used to customise a tool for use with a particular faculty, department, or course.

It can be used to explain the function of the tool to its users.

Purpose

This text was entered in the "Rubric" field of the same screen. It is used to describe the faculty, department, or course-specific context in which the tool will be use. *(Alternatively if the tool is an information page then the text of the page could be entered here.)*

View academic year:

2.3.5.5.3. Accessing the new tool

As can be seen from the above description, tools draw on the components of the *Internet-PARs* shell in order to customise themselves based on user type and affiliations within the institution. Access to specific tools is controlled in much the same way.

Using the same customisation procedure (see Figure 4 and Figure 5) as described in the previous section, administrators can allow or deny access to a given tool for users in their own, and subordinate levels in the institutional hierarchy. This model supports the concept of the development of an institutional set of tools from which faculties and departments can select those appropriate to their own students.

Drawing all of the above processes together it is possible to see how the shell reacts to a user trying to access a tool. In effect this process begins as the user logs into *Internet-PARs*. As a student goes through the login process a 'cookie' is created for them. A 'cookie' is a short string of text which is created by a web-site in combination with the user's web-browser to store information, for example, about the user or their actions within the web-site. Such strings of text can either be written to a file on the user's computer or held only in the computer's memory until the user closes their web-browser. This cookie contains values which will uniquely identify that individual to *Internet-PARs*. In the case of both Nottingham and Newcastle the UCAS student number is used for this purpose.

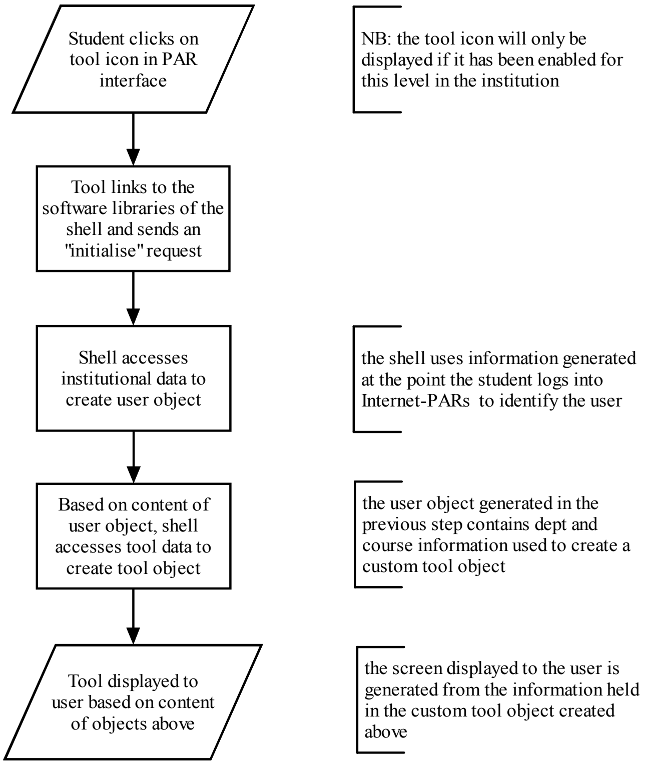
When the user calls up the new tool by clicking on its icon within *Internet-PARs*, its script, shown in Table 1 performs the following functions:

It establishes a link to the software libraries of the *Internet-PARs* shell with the "require" statement in line two of the script.

It creates two objects, student and tool, based on the content of the "initialise" statement in line three of the script. The student object is a fixed user object holding information about the user of the tool. The tool object however is unique to the circumstances in which it is created and is based largely on the contents of the object describing its user. The *Internet-PARs* system interrogates the student object for the lowest level in the institutional hierarchy with which they are associated. It then searches the tables described in Section 2.3.5.5.2 above to see if a customised version of the tool exists at this level. If this is the case, the system then displays the customised text to the user as part of the *Internet-PARs* interface. Otherwise, it will repeat the process at each successive level of the hierarchy until it reaches the default, or institutional level, at which point it will display the default text described in the previous section.

This process is summarised in Figure 6 below.

Figure 6: How the shell functions in response to a student accessing the new tool we have created in the above sections


Student clicks on tool icon in PAR interface
NB: the tool icon will only be displayed if it has been enabled for this level in the institution
Tool links to the software libraries of the shell and sends an "initialise" request
Shell accesses institutional data to create user object
the shell uses information generated at the point the student logs into Internet-PARs to identify the user
Based on content of user object, shell accesses tool data to create tool object
the user object generated in the previous step contains dept and course information used to create a custom tool object
Tool displayed to user based on content of objects above
the screen displayed to the user is generated from the information held in the custom tool object created above

2.3.6. Design and build tools …

In the early stages of the project a working group was convened at Nottingham to decide the approach taken to the development of tools within the *Internet-PARs* shell to be used with the pilot groups. Two models were considered based around the following views:

- a functional view where tools were provided to the users in a simple list
- a "process view" where tools were grouped according to the aspect of the educative process for which they provided support.

Following discussions with the pilot groups, it was decided that a "process view" provided the best analogy with the way in which the students would be using the pilot PAR. This had the additional advantage that an introduction and context could also be provided to accompany each process. Consequently, the interface presented in the pilot *Internet-PAR* used at Nottingham University was based on a process view covering three main sections:

- **Academic Support and Guidance**: Provides students with access to their programme of personal tutorial meetings and includes working with and modifying the agenda of their personal tutorials, recording and reflecting on the outcome of the meeting, and planning and organising actions arising from previous tutorials or in preparation for forthcoming meetings
- **Progress Review**: Allows students to take an overview of a range of academic feedback identifying themes and issues, to create and modify progress statements and to assess the development of personal and academic skills
- **Development Planning**: Providing students with tools to help them choose future modules, identify areas for skills development both inside and outside of the curriculum, and plan CV development.

Of these three areas, development was concentrated on the Academic Support and Guidance process, with the intention of moving in turn to Progress Review and then Development Planning if time allowed.

Within the Academic Support and Guidance process, the following tools have been developed and trialled with the pilot groups at Nottingham University:

User type	**Category**	**Tool(s)**
Admin	Administer PAR	1. Add agenda item for all students 2. Delete agenda item for all students
Student	Programme of personal tutorial meetings	1. View summary of previous personal tutorial meeting 2. Book appointment with tutor
Student	Working with the agenda of tutorial meetings	1. View the default agenda for a tutorial 2. Add new individual agenda items
Student	Tutorial preparation	1. Add preparation text
Student	Recording personal tutorial outcomes	1. Record discussion of specific agenda items 2. Edit existing records
Student	Tutorial follow-up	1. Record planned actions
Tutor	Programme of personal tutorial meetings	1. Allocate general availability 2. Allocate availability for specific meetings 3. View students who have booked appointments
Tutor	Programme of personal tutorial meetings	1. View accessible areas of tutee’s PAR

The following generic tools have also been developed and trialled with the pilot groups:

User type	Category	Tool(s)
Student	Contact tools	1. Easy email access to tutors
Student	Student timechart	1. View timechart of all events 2. View summary information associated with an event 3. Book appointment for meeting
Admin	Administer PAR	1. Add timechart item 2. Modify timechart item 3. Delete timechart item
Student	Student Notebook	1. View contents of student notebook 2. Make notes (general or module-specific) 3. Edit existing entries
Tutor	Tutor Notebook	1. View all notes relating to a single tutee 2. Make notes about assigned tutees 3. Edit existing entries
Student	Password Tools	1. Change Password 2. Request password (sent to student's registered email address)
Tutor	Password Tools	1. Change Password 2. Request password (sent to tutor's registered email address)
Student	About me	1. View academic information (course, modules, transcript, etc) 2. View personal information (contact details)
Admin	Administer PAR	1. Add new tutor 2. Assign tutees to new tutor 3. Delete tutor
Admin	Configure PAR	1. Alter customisable elements of any tool 2. Alter access rules for any tool and category of user 3. Alter terminology in PAR glossary 4. Alter colour schemes / font settings of PAR
Tutor	Contact Details	1. Enter contact information 2. Edit existing information

Development work has begun for tools in the Progress Review section of the pilot PAR, with data structures and interface designs taken to a first draft stage. However, these designs have not been tested with the pilot groups. As of 31 March 2000, *Internet-PARs* requires further development under the process headings of Progress Review and Development Planning to cater for more developed student reflection and for recording and reviewing extra-curricular activities.

2.4. *Trialling and evaluating the use of the software*

2.4.1. Organisation and methodology

Five academic schools representing a wide variety of disciplines (Economics, History, Education, Mathematical Physics and Medicine) at Nottingham were involved in piloting the partial prototype *Internet-PARs*. For many of the staff and students involved, the concept of a PARs, IT or paper based, was novel. Thus initial meetings with the parties concerned discussed the whole idea of Personal and Academic Records before moving on to an Internet version. The latter was usually introduced to students at a hands-on session with a member of the project team present. Feedback was gathered verbally and by the means of simple questionnaires.

The prototype *Internet-PARs* consisted of three inter-linked sub-processes, namely Academic Support and Guidance (recording the interaction between tutee and tutor on such matters as module and pathway choice, feedback of assessment results), Progress Review (the student, perhaps with some guidance from their tutor, reflecting on academic feedack and recording their achievements) and Development Planning (the student developing techniques and skills of forward/action planning with regard to module choice, CV development and so on).

In the light of this piloting, modifications were made to the *Internet-PARs* shell. The pilots were extended to three other schools/departments namely Environmental Engineering, Biological Sciences (a split-site school) and Physiotherapy. The School of Nursing was also interested but they were unable to commit resources at the appropriate time.

Permission was obtained from about 300 undergraduate students to use their personal data in this stage. The data was downloaded from SATURN (Nottingham's database of student records) and transferred to Newcastle where it was uploaded into the *Internet-PARs* database. Once a suitable protocol for doing so had been established, this was carried out on a regular basis so that the records held at Newcastle were reasonably up to date.

This influx of new schools brought new challenges, as the pilot *Internet-PARs* proved not to satisfy all their needs. In particular, as the project progressed it became apparent that the interface used in the initial stages was not user-friendly and so was subsequently redesigned. Also there were serious concerns about the ability to navigate in an obvious and intuitive manner around what was becoming a complex system. Another major difficulty encountered was that the time taken for individual Schools to tailor the system to their own needs was greater than anticipated. This customisation had to be carried out in the weeks leading up to the start of the academic year, an extremely busy period for the academics involved, especially as these colleagues were often responsible for student induction and tutoring in general. As a result the academic contacts in the schools did not have sufficient time to devote to the project.

2.4.2. Evaluation strategy

There have been two stages within the strategy.

1. It was established from the outset that the Nottingham evaluators would have a developmental brief, liaising frequently between

- departments/schools piloting the PAR
- piloting departments/schools and the Newcastle team.

Rapid collection and logging of feedback from Nottingham 'users' became the main focus for the Nottingham Project team discussions and in turn informed regular discussions with the IT development team at Newcastle. Modifications of the IT tool were thus made in response to user feedback. On the Nottingham campus liaison took the form of

- scheduled meetings which brought together all the departmental contacts ("administrators")

- scheduled discussions between a Project team representative and staff and students in each pilot department
- informal contact in response to departmental demand
- informal contact initiated by the Project team in response to emerging issues, both local and national.

2. In the latter stages of the Project, after consultation with the users, the team agreed to provide quantitative and qualitative data for the final report to the DfEE as follows (see also Section 3.7 Appendix 7 for external evaluator's report):

i. Numbers of piloting staff and students within each department/school as a percentage of the total number of staff/students in the relevant cohort; information about the existence or absence of any pre-existing PAR before the department/school joined the pilot

ii. Reactions from students to the experience of using *Internet-PARs*; this would be derived from a detailed questionnaire and from discussions with small groups of students

iii. Reactions collected from discussions with administrators and other piloting tutors.

2.4.2.1. Main findings

2.4.2.1.1. Quantitative information

Please see table in Section 3.5.1 Appendix 5 for breakdown by discipline of staff and student numbers involved in the piloting. Fewer than half the students had had any previous experience of formally recording and reviewing progress during their HE course.

2.4.2.1.2. Qualitative information

Student responses to Internet-PARs

A copy of the questionnaire used and full details of the range of answers provided are presented in Section 3.5.2 Appendix 5.

Despite the inevitable problems encountered in the first trials of a C&IT development project, the students have remained largely positive. While usefully reporting all encounters with bugs in the system, they were willing to think about the potential of the system for the future and many did so with enthusiasm. The main features of their responses to the questionnaire were as follows:

- their assessments of the value of the three main functions of *Internet-PARs* correlate exactly with the extent of development of each function in the time available within the project:

❑ formal record of academic progress (most developed)	**generally positive**
❑ reflective record of personal development and skills (less developed)	**evenly divided**
❑ reflective record of additional skills and competencies, developed by extra-curricular work/activity (least developed)	**unconvinced**

- a majority suggested a major role, potentially, for a system of this kind in relation to:
  - personal tutoring
  - academic interactions with staff
  - their own skills development

    - ❑ CV writing
- concerns involved:
    - ❑ security of data
    - ❑ level and reliability of IT provision
    - ❑ availability of training.

Staff responses to Internet-PARs

It has been evident that, in spite of good will, enthusiasm and willingness to take part in discussions to inform the design process, staff have had rather less time than students to persevere with the system when it has failed to work straightforwardly in the early stages. There has been little incentive to continue trying to use it to arrange tutorial appointments when it is easier and quicker to do this by phone, notice board or email. This has made it difficult to gather detailed responses to the use of *Internet-PARs* within the time-scale of the Project.

Scheduled discussions with representatives from each piloting school/department (administrators) did however prove most useful throughout the Project in critiquing design features and ensuring that the IT tool was capable of accommodating a wide range of needs.

2.5. Determining outline needs and views of schools and employers

2.5.1. Liaison with schools and colleges

In response to Objective 5 in the project contract ('Ensure that PAR development is consistent with other developments in the theme area (e.g. use of NRA at pre-university level)'), an advisory group was established, consisting of:

- a Head Teacher from an 11-18 comprehensive school
- an FE Vice-Principal
- a Sixth Form College Vice-Principal
- the Nottingham Project team.

This group met five times during the project (in December 1998 and in January, March and December (2 meetings) 1999). The 16-19 representatives also attended the national project conference held at Nottingham on 9 June 1999. The group's aims were as follows:

- to look for common ground between the tertiary and higher education sectors, in approaches to academic support and guidance; how successfully are we developing complementary structures in the two sectors to support lifelong learning?
- to keep in touch with national developments in the Progress File and to look for areas of overlap between that and HE PARs
- to consider the implications of the proposed changes in UCAS (the Tariff and Profile)
- to look at the ways in which students are being introduced to Key Skills in both sectors
- to share good practice.

2.5.1.1. Academic support and guidance

Discussions revealed a number of common strands perceived by all those teaching students in post-compulsory education:

- Increased student numbers, in response to government's educational targets and HE expansion, are putting pressure on support and guidance systems

- Greater diversity in this student body necessitates more, and more kinds of, support. Staff are inclined to favour only those systems which they see as streamlining administration, but -
- the culture is changing, so that resisting the implementation of some form of individual reviewing and target setting would now seem unreasonable, particularly if the PAR improves students' motivation and makes them more independent learners. ('Teaching' is moving closer to 'facilitating learning')
- The emphasis on learning needs to be kept at the centre of any PAR system, particularly in a school setting, where 'support and guidance' is still sometimes viewed as an exclusively pastoral concern
- To maximise its potential and avoid cynicism or short cuts, students, as well as staff, need some kind of training in the principles and processes of such a system.
- To make 'lifelong learning' more than a phrase, all those working in the post-compulsory sectors of education need to be well informed about one another's practices and to help students develop a language to discuss the improving of their own learning.

2.5.1.2. Progress File

The meeting between Jim Shaw of the DfEE and this group on 10 December 1999 provided opportunities to explore a number of issues involved with reviewing and recording in both the 16-19 and HE sectors. The discussion identified the following points of common interest.

- Shared challenges:
  - ❑ to encourage students to take an overview of their progress as a whole, including paid work experience
  - ❑ to encourage students to use Progress File in a sustained manner
- The need for evidence to show the impact of Progress File on academic achievement
- The need to explore its applicability to staff
- A culture of reviewing progress is growing in schools and the material produced by students is also shared with parents (An example of the use of IT to involve parents directly in the reviewing process at a school parents' evening is described in 'Independent learners - how to create them', by Allan, Burnett and Lewis, in *Managing Schools Today*)
- Further cultural shifts are however needed in schools:
  - ❑ developing partnerships between students and teachers
  - ❑ development of the sophistication of the process through the school
  - ❑ organising schemes of work and learning processes to highlight progress
  - ❑ provision of more feedback for learners on their skills development through formative assessment
  - ❑ helping students to study independently

It was agreed that:

- the spread of PARs across the campus at Nottingham (3000 students so far) provides a good opportunity to begin a study of the transition between pre-HE and HE-based experience to feed into the further development of Progress File
- Nottingham University's commitment to the concept of The University of the Community and existing Access initiatives provide an ideal basis for tracking local mature students with Progress File as they move from FE through HE at Nottingham

- there is scope for exploiting links between New College Nottingham and the University in tracking students using the Learning Log which is being developed by the University for Industry.

2.5.1.3. Changes in UCAS and the increased emphasis on Key Skills

The UCAS Tariff and Profile Initiative, taken together with UCAS's proposed electronic application form, was clearly a parallel development to the PAR. The group spent some time studying the UCAS proposals and exploring their scope and implications. The following aspects were of particular interest:

- the system's capacity to bring together information for and about the candidate in a single database comprising pre-HE and HE curriculum details
- the potential for overlap between a student's Progress File, electronic application form and PAR
- the visual highlighting of a student's key skills and 'other achievements', with the incentive this provides to write a more sharply focused Personal Statement -- one which reflects on attributes, achievements and aspirations rather than listing them.

2.5.1.4. Sharing good practice

After exchanging and comparing documents used for reviewing and recording in the three 16-19 institutions, the group arranged for the Project Team to run a demonstration of the *Internet-PARs* on 9 December 1999 for a group of students and staff at the Sixth Form College. The college students seemed to understand the underpinning philosophy behind PARs and were familiar with the process of reflection and recording of personal development. In the 'hands on' session, they found it relatively easy to navigate through the site. This was helped by the fact that they were already familiar with most standard web functions and processes. They particularly liked the function which enabled them to add their own items to standard tutorial agendas.

2.5.1.5. Outcomes

- Discussions within the 16-19 Advisory Group established extensive common ground between the pre-HE and HE sectors
- An awareness of the importance of establishing and maintaining stronger lines of communication between sectors -- the influence and autonomy of universities is already recognised; what is less widely recognised is what institutions of HE could gain from a more detailed understanding of the backgrounds and experiences of students and their previous institutions
- Links between the *Internet-PARs* at Nottingham and Progress File to be maintained.

2.5.2. Liaison with employers

2.5.2.1. Approaches to employer liaison

The Project Team's original plan was to establish a small advisory group of local employers that would function similarly to the 16-19 liaison group, perhaps coming together with that group at one meeting for a demonstration of the *Internet-PARs* system. This plan was subsequently modified in response to two operational problems:

- generating a suitable group of representatives of local SMEs proved very difficult
- those aspects of the PARs tool that were felt to be of most relevance to employers were still at an embryonic stage -- unsurprisingly, since piloting departments (and therefore IT development work) were concentrating on first-year undergraduate use,

whereas it was likely to be the PARs' capacity to stimulate students to think about 'enterprise skills' that would initially attract employers' interest and involvement.

Rather than continuing to seek to form an Employers' Group as such, therefore, it was decided:

- to utilise links with the Nottingham University Careers Advisory Service, built up during the PADSHE Project, to make individual contacts with local SMEs
- in particular, to do this via the Shell STEP Programme
- to attend relevant national conferences in order to gather relevant information on the perspectives of larger employers (CRAC: Key Skills at Work, 25.1.99; Strategic Alliances: Education-Business Links, 5.10.99)
- to network with local employers, both providers and delegates at Project events (e.g. Staff Development Workshop on Skills, Dave Wilson, BT)
- to draw on STEP contacts to hold one meeting with local SMEs towards the end of the first semester of trialling to consider local employers' expectations of graduate recruits.

2.5.2.2. Key aspects of Internet-PARs for employers

The insights gained through all these avenues formed an extremely valuable addition to the information obtained from the academics involved in piloting and from the 16-19 Liaison Group.

1. These insights raised the Project Team's awareness of the usefulness of *Internet-PARs* for students in the following areas:
    - helping students from the very beginning of their course to become more aware and more reflective about skills
    - helping them to develop a vocabulary for desribing their skills and personal attributes
    - highlighting the benefits of work, both paid and unpaid, and of community service for a student's personal development and the contribution this could make to their employability
    - offering a CV building facility
    - while training students in the processes which are central to effective Continuing Professional Development
    - emphasising that the responsibility for career management and for improving one's own learning are closer than students might imagine
    - giving employers access to a graduate database
    - that many students come into HE completely unprepared for the culture of work they will soon be entering and that academic tutors have usually received no training in preparing them for this -- within this context PARs have an important synthesising function, encouraging students to view their experience of university as a whole and to collect evidence to enable them to articulate what they have learned.
2. These insights also fed into discussions about facilities the *Internet-PARs* could offer end-users:
    - directly, as information
    - indirectly, by encouraging graduates to assess their own abilities and skills, thus becoming more attractive to employers.

The meeting with representatives of SMEs, held in November 1999, confirmed the perspectives already gained from a variety of employers and professional associations (e.g.

AGCAS, British Chambers of Commerce, North Nottinghamshire TEC, School-Business Link groups). The *Internet-PARs* would be directly relevant to graduates entering the workplace by helping to provide graduates who are:

- developing skills before, during and after HE
- able to develop productive skills and to apply these skills to achieve goals
- flexible and quick learners, able to pinpoint core strengths and provide evidence of abilities
- able to take responsibility for their own career management
- willing to develop a portfolio, rather than expecting a smooth and continuous career track
- able to articulate what they have learned from work experience, work placements and extra-curricular activities during their degree or gap year (either pre- or post-HE)
- able to 'add value' immediately in a small company
- able to offer initiative and inventiveness.

The meeting also gave the employers 'hands on' access to the *Internet-PARs* and stimulated discussion about how it might generate material for a graduate database which employers could access.

2.6. Institutional take-up and wider dissemination

Prior to the beginning of the *Internet-PARs* Project, the University of Nottingham had already decided to develop Personal and Academic Records (PARs) across the campus. However, it is now putting provision in place to support the schools and departments which have been piloting *Internet-PARs* as they continue to develop their customising of the tool and to appoint new C&IT staff, dedicated to this work, to take *Internet-PARs* forward at Nottingham. Likewise, the University of Newcastle identified in its institutional plan the need for PARs, and is piloting their implementation at a faculty level during 2000-2001. Both of the partner Universities appreciate that web-based systems are essential to the future of recording achievement not only at Nottingham and Newcastle but for the whole HE sector.

Interest from other HEIs is high. Six demonstrations involving the Nottingham team have been requested so far during the current session at: University of Ireland College at Maynooth, UMIST, University of Ulster, University of Portsmouth, University College, London, University of Warwick. Senior managers from 37 universities saw and discussed a demonstration of *Internet-PARs* at the PADSHE conference held at Nottingham on 9 June 1999. Similarly, the Newcastle team has demonstrated the *Internet-PARs* system to senior academics and administrators from 15 medical schools at a meeting of Medical School Secretaries and Registrars held at Newcastle University, 8 January 1999, and at a Physiological Society Teaching Symposium held in Newcastle, 14-15 July 1999. Please see Section 3.6 Appendix 6 for extensive list of current HEI contacts. Other key stakeholders who are still in contact with the project include: UfI, ILT, SCoP and QAA.

A national policy has been announced for transcripts and personal development planning in the HE sector. The *Internet-PARs* project made significant contributions to events and materials produced during the consultation exercise organised by CVCP/SCoP/QAA and was instrumental in developing the Internet perspective which is now firmly part of the HE scenario for implementing policy in this area.

The potential of *Internet-PARs* for wider use and further development is great. The Project Team has advised the DfEE of the demand for dissemination of *Internet-PARs* in the HE sector and of the need to disseminate not just web-based approaches to recording achievement but also the issues and practicalities of implementing them. Further development would have a great deal to offer the HE sector[10] and all the other sectors of education and employment with which it interacts. An Internet-based approach appears to be the key to linking personal development planning for the future through all the stages of an individual's education, training and career-building. *Internet-PARs* both make the case for national coordination in the next phases of development of C&IT in all these areas and exemplifies a solution which could be applied both in schools and in continuing professional development, as well as in HE.

2.7. Acknowledgements

The project team wishes to take this opportunity to express its sincere gratitude to the numerous colleagues, both staff and students, who have given so much of their time, knowledge and opinions to help take this project forward. In particular, we would like to thank Jane Austick and Jenny Wallace (DfEE *HEQE* Recording Achievement Managers), and Sue Otter (DfEE Advisor) – without their thoughtful and enthusiastic advice the project would not have made the progress it has.

[10] *DRUMMOND, P., HAMMOND, G.R. and SMALLWOOD, A.* (1999). *Internet-PARs*: A Web-based System for the Development of Personal and Academic Records (PARs) in Undergraduate Medical Education. CTICM Update **10**, 18-20.

3 APPENDICES

3.1 Appendix 1: Library Routines

3.1.1 General_Lib

3.1.1.1 Subroutine: INITIALISE

Required by every script, generates all startup parameters and user information

Input: Expects to see the following variables in the call

$type	student/tutor/admin
$location	Where in the par we are (tutorials/request)
$tool	Which tool we're accessing from

Example Call:

initialise("student","about_me/academic","New Tool");

Output:

This routine has no specific output but it will perform the following functions:

- Links in all of the libraries we need (object, mysql etc.)
- If there is already a username in the cookie make use of it, otherwise call the Cookie subroutine to create it. This is where unauthorised users are detected and returned to the login screen.
- Work out if this user has a usertype value in the cookie and if not create an object based upon the user type (student/admin/tutor)
- If the usertype is a tutor, create a subsidiary student object to allow a tutor to look at the records of their students (used for "View Student Par")
- Create a default set of tools for usertype, username
- Create an object based upon the tool we're currently accessing using the alternative 'create_tool_object(type, location, tool)'
- Check to see if this is the developers machine (this exists only for development purposes but allows the display of debug information only for system developers)

3.1.1.2 Subroutine: GENERATE PRINT FILE

This subroutine works out the Unix path to cgi-bin/print/files/, creates a display file with its name based upon process ID (PID), prints everything from the variable $file_contents into that file, and returns the name of the file

Future Developments:

Needs to do a tidy up on files that are a certain age

Input:

$file_contents

Example Call:

generate_print_file("$text_string");

Output:

Name of the file that was created

3.1.1.3 Subroutine: READ PRINT FILE

Reads in the contents of a print file to display it back to the user

Input:

$filename

Example Call:

read_print_file(“43211.prn”);

Output:

@file_contents

3.1.1.4 Subroutine: CREATE OBJECT

Is the basis for the object approach. Can create objects based upon type and value and will then create the object from the relevant library, e.g. student from student.pl

Input:

$usertype	student/tutor/admin/module/meeting
$username	the qualifier for the above type (e.g. 961160610)

Example Call:

create_object(“student”,”912345678”);

Output:

In itself this routine produces nothing, however it calls the appropriate subroutine to create the associative array containing the meeting schedule for the usertype and username provided

3.1.1.5 Subroutine: REQUIRE LIBS

Work out our unix pathname and include all of the libraries that will be needed relative to our current position. Modifies @INC to add /path/to/user/cgi-bin/lib to the search path for require.

3.1.2 HTML_Lib

3.1.2.1 Subroutine: DISPLAY HTML HEADER

Displays the HTML at the top of the page (including navigation buttons). In this (multiframe) version of the PAR, this script makes use of Javascript elements to control all of the frames together

Input:

$page_title	The title of the page
$nav_bars	Display the navigation bars or a back button

Actions:

Firstly, we need to check if the nav bars have been set (Javascript:hist....)

If so... we need to add one to the BACK cookie

If not... we need to reset the BACK cookie

Output:

Top and Left-hand frames seen in the Nottingham interface

3.1.2.2 Subroutine DISPLAY HTML FOOTER

Displays the HTML at the bottom of the page (contact details + all navigation buttons)

Inputs:

$fav	Add to favourites?
$print	Can they print?
$type	User type student/tutor/admin?

3.1.2.3 Subroutine: DISPLAY NAVIGATION BUTTONS

Called by the Display HTML Footer subroutine, this subroutine displays all of the relevant buttons at the bottom of the screen. The output of the routine is dependent on the user type.

3.1.2.4 Subroutine: DISPLAY PRINT HEADER

As display_html_header, except a different layout is output is for the printed version

Input:

$page_title	The title of the page

3.1.2.5 Subroutine: DISPLAY PRINT FOOTER

Display the print version HTML at the bottom of the page (this version includes none of the normal contact details or navigation buttons).

3.1.2.6 Subroutine: DISPLAY ERROR

This routine provides a standard handler for errors in the Internet-PAR. It prints error type and description to screen along with contact details to report the error

Input:

$error_type	e.g. "Could not authenticate your details"
$error_message	full reason for the error

3.1.2.7 Subroutine: HEADING FONT

Standardises the heading font and removes the requirement to keep printing <font face="tahoma, arial, helvetica">Blah blah</font>

Input:

$heading_text	string || &nbsp
$size	size || 2
$additional_tags	Anything to go <FONT $additional_tags>

Output:

Returns a bold version of the string in the desired font face and size

3.1.2.8 Subroutine: BODY FONT

Standardises the body font and removes the requirement to keep printing <font face="times roman">Blah blah</font>

Input:

$heading_text	string || &nbsp
$size	size || 2
$additional_tags	Anything to go <FONT $additional_tags>

3.1.2.9 Subroutine: START TABLE

Creates a basic table template: <TABLE border=0 cellpadding=5 cellspacing=0> To provide a uniform layout within the PAR. Requires: end_table() to be run at some point within the same script

Input:

$additional_tags	e.g. WIDTH=100%

3.1.2.10 Subroutines: TABLE HEADING, SUB HEADING & CELL

Will do the same as heading font but for tables. Will save time because we don't want to have to keep printing <font face="tahoma, arial, helvetica">Blah blah</font>

Input:

$heading_text	Text to be displayed
$additional_tags	TD tags to be added to
$new_row	Should a <TR> be printed?

3.1.2.11 Subroutine: NR

Print out a number of new rows

Input:

$number_of_prints	How many rows to print

3.1.2.12 Subroutine: END TABLE

Generates </TABLE>

Input:

$additional_tags	Printed after the </TABLE>
$print_rows	print 10 rows before ending the table (spacing)

3.1.2.13 Subroutine: GENERATE TEXT LIST

Takes a list of elements and creates an indented HTML (<UL>)list from them

Input:

@elements	The elements to appear on the list

3.1.2.14 Subroutine: NEW FORM

Print out all of the associated HTML related to new form creation

Input:

$action	Where is the form to go to?

3.1.2.15 Subroutine: DISPLAY PHOTOGRAPH

Displays student photograph

Input:

$student_number	to identify photograph to display
$additional_tags	to print as part of the <IMG> tag

3.1.2.16 Subroutine: DISPLAY USERNAME DETAILS

Displays the users (student/tutor/admin) Name Details in a blue border at the top of the page, followed by apppropriate tool Introduction and Rubric

Input:

$self	HASH
TOOL_TYPE	student/tutor/admin
USER_TYPE	student/tutor/admin :-)
USER_NAME	961160610
TABLE	Any table tags
CLOSE	1/0

3.1.2.17 Subroutine: DISPLAY HIDDEN VARS

Will take a reference to an anonymous hash and display hidden form vars as the result

Input:

```
display_hidden_vars( {mhairi => Golden Freeway; paul => Manager Type } );
```

Output:

```
<INPUT TYPE=HIDDEN NAME='paul' VALUE='Project Manager'>
<INPUT TYPE=HIDDEN NAME='mhairi' VALUE='Golden Freeway'>
```

3.1.2.18 Subroutine: OUTPUT FORM DATE

Return form elements (pulldown lists) for a particular date

Input:

$default_date	Anything passed to the script or today

Example Call:

```
print output_form_date($date);
or
print output_form_date();
```

3.1.2.19 Subroutine: OUTPUT FORM TIME

Output similar to output_form_date()

Input:

$default_time	The time in hours:mins

Example Call:

```
print output_form_time($date);
or
print output_form_time();
```

3.1.2.20 Subroutine: SUBMIT

All submit buttons take place on a table_sub_heading. Syntax is exactly the same as table_sub_heading except STRING is replaced by image_type

CALLING:

```
Submit("submit", "COLSPAN=2", 1);
```

3.1.2.21 Subroutine: DISPLAY DROPDOWN MENU

Display a pulldown menu (form element)

Input:

NAME	form element name

DISP_VAL	Values to go into dropdown
REAL_VAL	Values to go into VALUE=XXX>
DEFAULT	default value
SORT	should we sort it?

Example Call:

display_dropdown_menu({ NAME=>"TEZ"; DISP_VAL=>[@elements]; DEFAULT=>$default; SORT=>1 });

3.1.2.22 Subroutine: XPRINT

Used instead of print() this will add the contents of any print statement to the variable $file_contents which can then be passed to the GENERATE PRINT FILE subroutine. It is part of the print output concept which was built into all early versions of tools in the Internet-PAR

Input:

$print	string to print
$should_I_print	should this statement be output or not

3.1.2.23 Subroutine: SPACER

Will display a spacer gif $width pixels wide

Input:

$pixels	width of gif in pixels

Calling:

spacer(20)	20 pixel spacer

3.1.2.24 Subroutine: NOTEBOOK AND TUTOR NOTEBOOK

Create a link into the student/tutor notebook with a particular type (module code etc.)

Input:

$type	module code - ACC302
$sort	Not really useful
$date	Are we looking for a particular timechart entry?
$time	Are we looking for a very particular timechart entry?

3.1.2.25 Subroutine: TUTOR NOTEBOOK

Provide a tutor notebook or Aide Memoire. The Aide Memoire automatically picks up all students with whom the tutor is involved.

3.1.2.26 Subroutine: VIEW NOTEBOOK

Jump into the overview of the notebook entries for a particular type and academic year

Input:

$type	module code - ACC302
$academic_year	3

3.1.2.27 Subroutine: HREF

Display all of the relevant HTML that goes with an <A HREF...

Input:

$link_to	Where is this HREF going to?
$text	What should it display as it's link?
$title	Should there be a title (ala, IE5)?
$color	Should it appear in WHITE?

3.1.2.28 Subroutine: FONT COLOR DATE

Check to see whether a date is before the current date and if so, return a <FONT COLOR='#WHATEVER'>. This is currently an inefficient method of handling past/present dates and so may need to be updated.

3.1.2.29 Subroutine: DISPLAY PERSONAL TUTEES

Displays a list of the personal tutees of a tutor allowing them to be linked to another script in the PAR

Input:

$tutor_code	Who is the tutor we wish to look at?
$action	Where are the student numbers going to link to?

Output:

A list of students with their student number as a link to "form action"

Actions:

3.1.2.30 Subroutine: IMG

This routine will generate an image based upon three inputs:

Input:

$path_to_image	/project/images/buttons/tick.gif
$alt_tag	This is the descriptive text
$other_tags	ALIGN=RIGHT etc.

3.1.2.31 Subroutine: DISPLAY ADMINISTRATION LIST

Displays the list of all department/faculty/courses the admin has roles on.

3.1.2.32 Subroutine: DISPLAY PROCESSES

Display a list of the top level processes for the left hand frame for any particular user. This is incorporated because of the frames approach.

3.1.3 MySQL_Lib

3.1.3.1 Subroutine: CONNECT TO DB

Connect to $db on $host

Input:

$host	Host of the DB
$db	DB to connect to

Output:

$dbh	Global database handle

3.1.3.2 Subroutine: SEND SQL QUERY

Reads in an SQL query and sends it the DB

Input:

$sql	Formatted SQL query

Output:

@results	Only on certain types of query

3.1.3.3 Subroutine: UPDATE INFO IN DATABASE

Performs an update statement on the DB with the following syntax:

Input:

$table	the name of the table to be updated
$update_hash update	a reference to an anonymous hash of key value pairs for the
$where_hash where clause	a reference to an anonymous hash of key value pairs forming the

Example Call:

update_info_in_database("TABLE_NAME", {fieldname => $field_value, fieldname2 => $field_value2, fieldnameN => $field_valueN}, {wherefield => $where_value});

Output:

Creates and sends (via the send_sql_query subroutine in Mysql_Lib)

update TABLE_NAME SET (fieldname = $field_value, fieldname2 = $field_value2,fieldnameN = $field_valueN) WHERE wherefield = $where_value

3.1.3.4 Subroutine: REPLACE INFO IN DATABASE

Exact same syntax as update_info_in_database except if performs a REPLACE INTO ... Good if you aren't sure:

a) if the value is in the table to be updated (an update would fail if there were no record)

b) if you know what the key values of the table you're replacing into are

3.1.3.5 Subroutine: ADD INFO TO DATABASE

Take the information from the form and add it to the database

Input:

$table	
@_	array full of data values to input

3.1.3.6 Subroutine: STUDLYCAPS

Useful case-conversion routine for strings

Input:

$string	string to be converted

Example Call:

studlycaps(“this is a test”);

Output:

Converts any case into Title Case and returns this value, e.g.

this is a test	=> This Is A Test
engineering ii	=> Engineering II
TERRY BROWN	=> Terry Brown

3.1.3.7 Subroutine: CHECK LOGIN DETAILS IN DB

Looks in the correct username tables and returns true if the person is valid and false if not

Inputs:

$usertype	student/tutor/admin
$username	961160610
$password	abc123

Example Call:

check_login_in_db("student","912345678","abc123);

Output:

Returns a true or false value 1/0 depending on the result of the query

3.1.4 Admin

3.1.4.1 Subroutine: CREATE ADMIN OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Administrator Object described in Appendix 3

Input:

$admin_code	e.g. ntb1

Example Call:

create_admin_object("ntb1");

Output:

Creates %ADMIN associative array for the username provided in $admin_code

3.1.5 Cookie

3.1.5.1 Subroutine: GET COOKIE

This will check for the existence of a cookie based upon usertype (student/tutor/admin) and if not found will redirect the user to index_not_login.pl to force them to login to the Internet-PAR

Input:

$cookie_name	student/tutor/admin/year etc.

Example Call:

get_cookie("student");

Output:

$cookie_value if it exists or a redirect to index_not_login.pl to force the user to login

3.1.5.2 Subroutine: CREATE COOKIE

Creates a user cookie based on a hash of cookie name and value pairs

Input:

{cookie_name => $cookie_value, cookie_name2 => $cookie_value2...}

Example Call:

set_cookie({cookie_name => $cookie_value, cookie_name2 => $cookie_value2,...}});

Output:

Creates the cookie for the user type based on the cookie names and values provided.

3.1.6 Course

3.1.6.1 Subroutine: CREATE COURSE OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Course Object described in Appendix 3

Input:

$course_code	A106

Example Call:

create_course_object(”A106”);

Output:

Creates %COURSE associative array containing information for the $course_code provided

3.1.6.2 Subroutine: CHECK TRANSCRIPT

This subroutine returns the information in a students transcript record. The subroutine is currently configured to return transcript information by module and would require minor modifications to return a complete transcript for a student

Input

$student_number	e.g. 912345678
$course_code	e.g. A106
$academic_year	e.g. 1
$module_code	e.g. BUS301

Example Call:

check_transcript(“912345678”,”A106”,”1”,”BUS301”);

Output:

All transcript information relating to the specific student/course/year/module supplied in the call

3.1.7 Department

3.1.7.1 Subroutine: CREATE DEPARTMENT OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Department Object described in Appendix 3

Input:

$dept_code	5170

Example Call:

create_department_object(”5170”);

Output:

Creates %DEPT associative array containing information for the $dept_code provided

3.1.7.2 Subroutine: FIND DEPARTMENT CODE

Returns the department of the user described by usertype and username

Input:

$usertype	=> student/tutor/admin
$username	=> 961160610

Example Call:

find_department_code(”student”,”912345678”);

Output:

Department code

3.1.7.3 Subroutine: GET DEPARTMENT DETAILS

This subroutine returns the departmental details held in the institutional data corresponding to the supplied department code. Currently this is limited to departmental name, but could be expanded if required

Input:

$dept_code	e.g. 5170

Example Call:

Get_department_details(“5170”);

Output:

An array containing department code and name

3.1.8 General

3.1.8.1 Subroutine: CREATE DEFAULTS OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the default versions of the Objects described in Appendix 3

Input:

$usertype	student/tutor/admin

Example Call:

create_defaults_object(”student”);

Output:

Creates %DEFAULTS associative array containing the glossary and formatting options for the user type provided in $usertype. This subroutine will make use of existing objects to identify its user more accurately

3.1.8.2 Subroutine: WORK OUT USER AREA

A component of the create_defaults_object, this routine works out the area of the *Internet-PAR* in which the user is currently browsing, e.g. “Working with my Personal Information”.

3.1.8.3 Subroutine: WORK OUT DATES

A component of the create_defaults_object, this routine returns semester and academic year start and end dates for the current user. NB while the demo data only contains semester info for the institution, it is recognised that some courses such as medicine will have differing start and end dates.

3.1.8.4 Subroutine: PRINT OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. This is a diagnostic subroutine left in the libraries to help developers. It will print out all of the keys and values related to an object, and works in a similar way to PERLS Data::Dumper

Input:

student/tutor/admin/meeting......

Example Call:

print_object(”student”);

Output:

On-screen print out of all of the values held in the specified object

3.1.8.5 Subroutine: PAR GREP

This routine will search for a field in a given array. It will function for any hash array but is most frequently used to retrieve values from the objects created in the Internet-PAR

Input:

$search_term	search term
$array_ref	The array to search for the search term

Example Call:

par_grep(”surname”,”%STUDENT”);

Output:

The value associated with the search term in the given array

3.1.8.6 Subroutine: CHECK PROCESS

Relates UNIX directory names to the human readable processes making up the Nottingham version of the PAR. It functions by holding a hash lookup that maps a process to its full name, e.g.

tutorial	The Tutorial Process
personal	Working with my Personal Information
reflection	Reflecting on my Learning
timechart	Working with your timechart

Input:

$directory_name	e.g. personal

Example Call:

check_process(”personal”);

Output:

Assigns the value of the “full” process name to the variable associated with the call, e.g. “Working with my Personal Information”

3.1.8.7 Subroutine: PS - PARSE SPACES

Query strings in URLs are used to pass large amounts of data/values between the scripts in *Internet-PARs*. This method while valuable has problems in dealing with spaces in the data/values being passed. The PS subroutine will replace spaces in data/values with “_” when passing them as query strings and replace “_” with spaces when the data/values are dealt with by a script

Input:

$string	e.g. a string of text
$flag	1 to replace “_” with spaces, blank to replace spaces with “_”

Example Call:

ps("Lengthy String")	returns - Lengthy_String
ps("Lengthy_String",1)	returns - Lengthy String

3.1.8.8 Subroutine: CR TO BR

Handles conversion of the \cM or \n line feeds passed to the PAR in user text into <BR> line feeds for HTML display. This routine would require adaptations in order to handle Macintosh line feed characters

Input:

$string	e.g. text \cM more text

Example Call:

cr_to_br(“text \cM more text”);

Output:

text <BR> more text

3.1.9 Glossary

3.1.9.1 Subroutine: READ GLOSSARY

This subroutine creates a full custom glossary for a particular user of the Internet-PAR. It reads the values in the values in the GEN_glossary database table in a four-pass process. First a pass is made at the course level and any entity/attribute tupples are added to the associative array. Subsequent passes occur at department, faculty, and institutional level with tupples being added to the array only if the attribute is not already defined

Input:

$usertype	student/tutor/admin
$username	e.g. 912345678

Example Call:

read_glossary(“student”,”912345678”);

Output:

Creates %GLOSSARY associative array containing the glossary entries for a particular user. If $usertype and $username are blank the default institutional glossary will be returned

3.1.9.2 Subroutine: GLOSSARY

This subroutine reads in a string of text and converts any standard glossary entries in it to the correct ones for a given user. It replaces the longest glossary terms first, working down towards single words in subsequent passes

Input:

Passed as an associative array containing the following values

STRING	e.g. Pastoral Tutorial
L_C	1/0

Example Call:

glossary ({STRING => "student number"})

Output:

The corresponding term(s) held in the glossary for this particular level and user

3.1.10 Mail

3.1.10.1 Subroutine: PAR MAIL

This routine relies on the existence of a correctly configured sendmail hub on the host machine. It acts as a batch mailer and is used to contact groups of tutees

Inputs:

$to_address	Terry.Brown@ncl.ac.uk, Mhairi.Marshall@ncl.ac.uk;
$subject	This is a subject
$message	This is the message
$bcc	1 or 0

Example Call:

par_mail("joe.bloggs@home.com, h.wills@home.com","subject is …","text of message …");

3.1.11 Meeting

3.1.11.1 Subroutine: CREATE MEETING OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Meeting Object described in Appendix 3

Input:

$student	912345678

Example Call:

create_meeting_object("912345678");

Output:

Creates %MEETING associative array containing the meeting schedule for the student provided in $student

3.1.11.2 Subroutine: WHICH MEETING

Because academic and calendar years do not match, this routine is used to work out if a meeting occurring in the same calendar year is in this academic year or in the last, or next academic year

Inputs:

$year	e.g. 1999
$month	e.g. 10

Example Call:

which_meeting(“1999”,”10”);

Outputs:

A variable containing the string this, next, last or other depending on the outcome of the sub-routine

3.1.12 Module

3.1.12.1 Subroutine: CREATE MODULE OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Module Object described in Appendix 3

Input:

$module_code	ACC302

Example Call:

create_module_object(”BUS301”);

Output:

Creates %MODULE associative array containing information for the $module_code provided

3.1.12.2 Subroutine: FIND MODULE NAME

This subroutine returns the name of the module corresponding to the code provided

Input:

$module_code	e.g. BUS301

Example Call:

find_module_name(“BUS301”);

Output:

A scalar variable containing the name of the module

3.1.12.3 Subroutine: GET MODULE CREDITS

This subroutine returns the number of credits associated with a particular module identified by the code provided

Input:

$module_code	e.g. BUS301

Example Call:

get_module_credits(“BUS301”);

Output:

A scalar variable containing the number of credits for this module.

3.1.13 Student

3.1.13.1 Subroutine: CREATE STUDENT OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Student Object described in Appendix 3

Input:

$student_number	912345678

Example Call:

create_student_object("912345678");

Output:

Creates the %STUDENT associative array for the username provided in $student_number

3.1.13.2 Subroutine: STUDENT IS ALREADY IN DATABASE

This is a boolean function which queries the database using a student number and a table name to search to see if the student already has an entry in the table

Input:

$search_student	e.g. 961160610
$search_table	e.g. STU_name_details

Example Call:

student_is_already_in_database("912345678","STU_name_details");

Output:

1	Student exists
0	Student does not exist

3.1.13.3 Subroutine: GET TUTOR OF STUDENT

Returns the tutor code of the tutor who has the specified role in relation to the student, e.g. pastoral, academic, senior

Input:

$tutor_role	e.g. senior tutor
$student_number	either passed in the call or obtained from the Student Object

Example Call:

Get_tutor_of_student("pastoral tutor","912345678");

Output:

Tutor code	e.g. 97k054

3.1.14 Time

3.1.14.1 Subroutine: CREATE TIMECHART OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates a Timechart Object

Input:

$student	912345678

Example Call:

create_timechart_object(”912345678”);

Output:

Creates %TIMECHART associative array containing the meeting schedule for the student provided in $student

3.1.14.2 Subroutine: CONVERT DATE

MySQL handles dates in the format YYYY-MM-DD which need to be converted to the more human readable DD-MM-YYYY format before outputting to the user screens

Input:

YYYY-MM-DD

Output:

DD-MM-YYYY

Example Call:

Convert_date(“2000-07-12”);

Actions:

Takes a mySQL format date field (YYYY-MM-DD) and converts to DD-MM-YYYY

3.1.14.3 Subroutine: GET MONTH STRING

Returns the text version of a numerical month for display purposes

Input:

$month	09

Example Call:

get_month_string(“09”);

Output:

Text value of input month, e.g. “September”

3.1.14.4 Subroutine: GET DAY STRING

Returns the text version of a numerical day for display purposes

Input:

$month	4

Example Call:

get_day_string(“4”);

Output:

Text value of input day, e.g. “Thursday”

3.1.14.5 Subroutine: FIND DATE

Uses existing UNIX system calls to return the current date in the format YYYY-MM-DD

Example Call:

find_date;

Output:

Date, e.g. 2000-07-18

3.1.14.6 Subroutine: FIND COMPARATIVE DATE

Given and date (1999-12-17) and a string (+/- N days) this routine returns the YYYY-MM-DD formatted string corresponding to the calculated date

Input:

$date	1999-12-17
$when	+30 days
$noparse	1/0

Example Call:

find_comparative_date("2000-07-19","+10 days","1");

Output:

The date string - 2000-07-29

3.1.14.7 Subroutine: TEXT DATE

Converts a date string input in the format YYYY-MM-DD or DD-MM-YYYY into a text version of the date

Inputs:

$date	DD-MM-YYYY or YYYY-MM-DD

Example Call:

Text_date("2000-07-18");

Output:

Text version of the date supplied, e.g. 18th July 2000

3.1.14.8 Subroutine: GET CURRENT YEAR

Designed to replace the 0000 year value in recurring timechart events, this subroutine has been largely superseded by the TRANSLATE 0000 subroutine described later, and should be removed as soon as all checks for extraneous calls have been made.

3.1.14.9 Subroutine: GET BOOKED DATE

If a student has booked an appointment for a tutorial within its date range, this routine will search for and return this date

Inputs:

Passed as an associative array containing the following values

student_number	e.g. 961160610
event	e.g. Second Pastoral Tutorial
start_date	e.g. 1999-10-07
end_date	e.g. 1999-10-14

Example Call:

get_booked_date({student_number => 912345678; event=>Second Pastoral Tutorial, start_date=>1999-10-07, end_date=>1999-10-14});

Output:

Returns two variables representing the date and time at which the meeting was booked

3.1.14.10 Subroutine: WHICH YEAR

Because academic and calendar years do not match, this routine is used to work out if an event occurred in this academic year or if it was in the last, or next academic year

Inputs:

Passed as an associative array containing the following values

year	e.g. 1999
month	e.g. 10
academic_year	e.g. 2000

Example Call:

which_year({year=>1999; month=>10; academic_year=>2000});

Outputs:

A variable containing the string this, next, last or other depending on the outcome of the sub-routine

3.1.14.11 Subroutine: TRANSLATE 0000

Recurring events are described to the *Internet-PARs* system as having a date in the format 0000-06-24 and are translated as in this example as occurring on the 24th June every year. This subroutine will accept the month value from this format of date and transpose the current academic year upon the student viewing that event

Inputs:

$month	06
$academic_year	2000

Example Call:

translate_0000(“06”,”2000”);

Outputs:

The year in which the event will take place based on the current academic year.

3.1.14.12 Subroutine: GET SEMESTER DETAILS

Takes the student number and displays the relevant semester information for that course/dept/institution. Currently the demo data only contains semester info for the institution, but it is recognised that some courses such as medicine will have differing start and end dates

Inputs:

$student_number	961160610

Example Call:

get_semester_details(“912345678”);

Outputs:

Creates the %SEMESTER associative array containing the following information:

NUMBER_OF	1/2/3/4...,
SEMESTER 1 NAME	Autumn,
SEMESTER 1 START DATE	20/09…
SEMESTER 1 END DATE	02/02 ...

3.1.15 Tool

3.1.15.1 Subroutine: CREATE TOOL OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Tool Object described in Appendix 3

Input:

$user_category	student/tutor/admin
$tool_category	tutorials/request
$tool	What are we looking at

Example Call:

create_tool_object(“student”,”about_me/academic”,”New Tool”);

Output:

Creates the %TOOL associative array based on the $user_category, $tool_category and $tool information provided

3.1.15.2 Subroutine: CHECK FOR TOOL COMPONENTS

Tools in *Internet-PARs* can have component parts which can be switched on or off depending on the access flag for a given component for a particular course or department. This routine uses the PAR GREP subroutine to check whether a tool has component parts and returns them as an array

Input:

$tool_location	e.g. /cgi-bin/tutorials/tutorial.pl

Example Call:

check_tool_for_components(“/cgi-bin/tutorials/tutorial.pl”);

Output:

An array of components for the specified tool

3.1.15.3 Subroutine: CC

Tools in *Internet-PARs* can have component parts which can be switched on or off depending on the access flag for a given component for a particular course or department. This routine checks the values of this access flag and provides its output accordingly

Input:

$component	e.g. student photograph
$return_if_true	value to return if component is enabled
$return_if_false	value to return if component is disabled

Example Call:

cc("student photograph", $STUDENT{photograph}, "XXXXXXXXX")

Output:

The appropriate return_if_… value

3.1.15.4 Subroutine: DISPLAY TOOL LIST / DISPLAY MENU LIST

Will read in from TOOL_general the correct tool descriptions etc. and display them

Input:

$type	student/tutor/admin
$category	tutorials/request
$q_string	is there a query string value for these tools
$sort	0/1 - should the output be sorted

Example Call:

display_menu_list("student"," tutorials/request","","1");

Output:

On screen menu of tools in the category specified

3.1.15.5 Subroutine: GET TOOL NAME

Read in (from TOOL_name) the user specific name for this particular tool

Input:

$usertype	student/tutor/admin
$tool_location	/cgi-bin/tutorials/tutorial.pl

Example Call:

get_tool_name($tool, $tool_category);

Output:

The user specific name of the tool, or if there is no user specific entry, the default institutional tool

3.1.15.6 Subroutine: GET LINK TEXT

Read in (from TOOL_name) the user specific link text for this particular tool

Input:

$usertype	student/tutor/admin
$tool_location	/cgi-bin/tutorials/tutorial.pl

Example Call:

get_link_text($tool, $tool_category);

Output:

The user specific link text for the tool, or if there is no user specific entry, the default institutional link text

3.1.15.7 Subroutine: READ IN TOOL INTRO

Read in (from TOOL_introduction) the user specific introductory text for this particular tool

Input:

$usertype	student/tutor/admin
$tool_location	/cgi-bin/tutorials/tutorial.pl

Example Call:

read_in_tool_intro($user_category, $tool_category, $tool)

Output:

The user specific introduction for the tool, or if there is no user specific entry, the default institutional introduction

3.1.15.8 Subroutine READ IN TOOL RUBRIC

Read in (from TOOL_rubric) the user specific rubric for this particular tool

Input:

$usertype	student/tutor/admin
$tool_location	/cgi-bin/tutorials/tutorial.pl

Example Call:

read_in_tool_rubric($user_category, $tool_category, $tool)

Output:

The user specific rubric for the tool, or if there is no user specific entry, the default institutional rubric

3.1.16 Tutor

3.1.16.1 Subroutine: CREATE TUTOR OBJECT

The data objects used in *Internet-PARs* to hold information about users, tools and events are created as globalised associative (hash) arrays. Once created they are available to all areas of the program. This routine creates the Tutor Object described in Appendix 3

Input:

$tutor_code	9507K054

Example Call:

create_tutor_object("9507k054");

Output:

Creates the %TUTOR associative array for the username provided in $tutor_code

3.1.16.2 Subroutine: GET TUTOR ROLES

Returns the roles in which a specific tutor operates within an institution, or if no tutor is specified, the tutor roles which are listed in the GEN_defaults table

Inputs:

Passed as an associative array containing the following values

Tutor	tutor code, e.g. 970k054
Level	level in institution, e.g. department
Name	name of entity, e.g. Economics

Example Call:

get_tutor_roles({tutor=>970k054});

or

get_tutor_roles({level=>department; name=>Economics});

Output:

An array containing all of the roles as described above

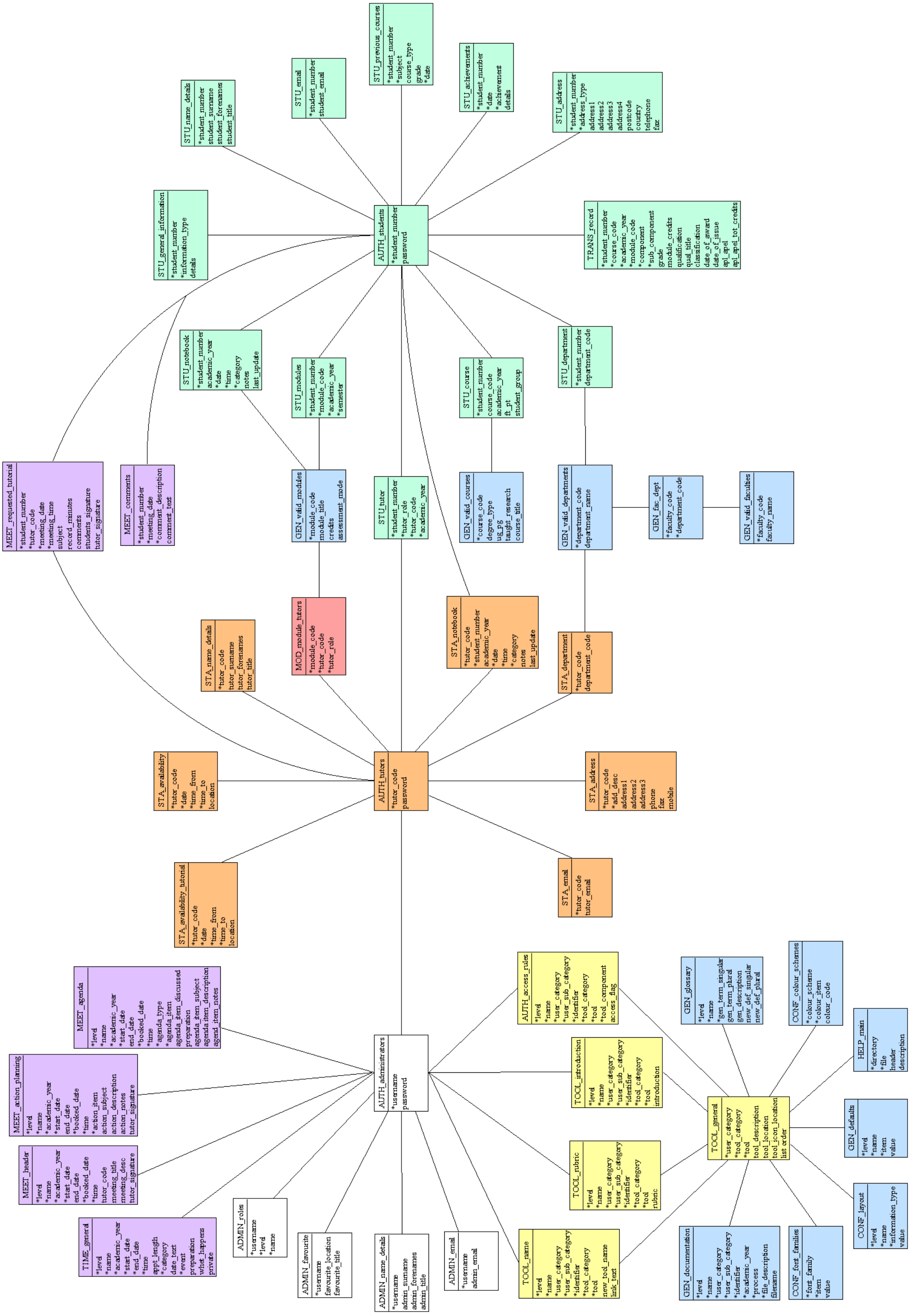
3.1.16.3 Subroutine: THIS IS A TUTOR

Because tutors need to be able to look a student entries in the PAR (albeit with certain restrictions) we sometimes need to know if the user is a tutor or not. This subroutine checks whether the usertype of the user is the same as the usertype for the tool. If these are not the same then we can assume that we have a tutor viewing a record via a student tool

Output:

True of False (1/0)

3.2 Appendix 2: Entity Relationships Diagram and Data Tables



ADMIN_email					
Field	Type	Null	Key	Default	Extra
username	varchar(9)		PRI		
admin_email	varchar(150)	YES			

ADMIN_favourite					
Field	Type	Null	Key	Default	Extra
username	varchar(50)	YES			
favourite_location	varchar(255)	YES			
favourite_title	varchar(50)	YES			

ADMIN_name_details					
Field	Type	Null	Key	Default	Extra
username	varchar(9)		PRI		
admin_surname	varchar(50)	YES			
admin_forenames	varchar(100)	YES			
admin_title	varchar(10)	YES			

ADMIN_roles					
Field	Type	Null	Key	Default	Extra
username	varchar(9)		PRI		
level	varchar(40)		PRI		
name	varchar(100)		PRI		

AUTH_access_rules					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
user_category	varchar(40)		PRI		
user_sub_category	varchar(100)		PRI		
identifier	varchar(9)		PRI		
tool_category	varchar(40)		PRI		
tool	varchar(200)		PRI		
tool_component	varchar(200)		PRI		
access_flag	char(1)	YES			

AUTH_administrators					
Field	Type	Null	Key	Default	Extra
username	varchar(9)		PRI		
password	varchar(20)			abc123	

AUTH_students					
Field	Type	Null	Key	Default	Extra
student_number	varchar(9)		PRI		
password	varchar(20)			abc123	

AUTH_tutors					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(10)		PRI		
password	varchar(20)			abc123	

CONF_colour_schemes					
Field	Type	Null	Key	Default	Extra
colour_scheme	varchar(128)		PRI		
colour_item	varchar(128)		PRI		
colour_code	varchar(255)				

CONF_font_families					
Field	Type	Null	Key	Default	Extra
font_family	varchar(128)		PRI		
item	varchar(128)		PRI		
value	varchar(255)				

CONF_layout					
Field	Type	Null	Key	Default	Extra
level	varchar(128)		PRI		
name	varchar(128)		PRI		
information_type	varchar(255)		PRI		
value	varchar(255)	YES			

GEN_defaults					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
item	varchar(255)		PRI		
value	varchar(255)	YES			

GEN_documentation					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
user_category	varchar(40)		PRI		
user_sub_category	varchar(100)		PRI		
identifier	varchar(9)		PRI		
academic_year	int(1)		PRI		
process	varchar(30)		PRI		
file_description	varchar(255)		PRI		
filename	varchar(255)				

GEN_fac_dept					
Field	Type	Null	Key	Default	Extra
faculty_code	varchar(10)		PRI		
department_code	varchar(10)		PRI		

GEN_glossary					
Field	Type	Null	Key	Default	Extra
level	varchar(128)		PRI		
name	varchar(128)		PRI		
gen_term_singular	varchar(128)		PRI		
gen_term_plural	varchar(128)	YES			
gen_description	blob	YES			
New_def_singular	varchar(128)	YES			
new_def_plural	varchar(128)	YES			

GEN_valid_courses					
Field	Type	Null	Key	Default	Extra
course_code	varchar(10)		PRI		
degree_type	varchar(10)	YES			
ug_pg	char(1)	YES			
taught_research	char(1)	YES			
course_title	varchar(255)	YES			

GEN_valid_departments					
Field	Type	Null	Key	Default	Extra
department_code	varchar(10)		PRI		
department_name	varchar(255)	YES			

GEN_valid_faculties					
Field	Type	Null	Key	Default	Extra
faculty_code	varchar(10)		PRI		
faculty_name	varchar(255)	YES			

GEN_valid_modules					
Field	Type	Null	Key	Default	Extra
module_code	varchar(10)		PRI		
module_title	varchar(255)	YES			
credits	int(3)	YES			
assessment_mode	varchar(50)	YES			

HELP_main					
Field	Type	Null	Key	Default	Extra
directory	varchar(255)		PRI		
file	varchar(100)		PRI		
header	varchar(255)	YES			
description	blob	YES			

MEET_action_planning					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
academic_year	char(1)		PRI		
start_date	date		PRI	0000-00-00	
end_date	date	YES		0000-00-00	
booked_date	date		PRI	0000-00-00	
time	varchar(5)		PRI	00:00	
action_item	char(2)		PRI		
action_subject	varchar(255)				
action_description	varchar(255)				
action_notes	blob	YES			
tutor_signature	char(1)	YES			

MEET_agenda					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
academic_year	char(1)		PRI		
start_date	date		PRI	0000-00-00	
end_date	date	YES		0000-00-00	
booked_date	date		PRI	0000-00-00	
time	varchar(5)		PRI	00:00	
agenda_type	varchar(20)		PRI		
agenda_item	char(2)		PRI		
ageneda_item_discussed	char(1)	YES			
preparation	blob	YES			
agenda_item_subject	varchar(255)	YES			
agenda_item_description	blob	YES			
agenda_item_notes	blob	YES			

MEET_comments

Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
meeting_date	date		PRI	0000-00-00	
comment_description	varchar(255)		PRI		
comment_text	blob	YES			

MEET_header

Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
academic_year	char(1)		PRI		
start_date	date		PRI	0000-00-00	
end_date	date	YES		0000-00-00	
booked_date	date		PRI	0000-00-00	
time	varchar(5)		PRI	00:00	
tutor_code	varchar(10)				
meeting_title	varchar(255)	YES			
meeting_desc	blob	YES			
tutor_signature	char(1)	YES			

MEET_requested_tutorial

Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
tutor_code	varchar(10)		PRI		
meeting_date	date		PRI	0000-00-00	
meeting_time	varchar(5)		PRI		
subject	varchar(255)	YES			
record_minutes	char(1)	YES			
comments	blob	YES			
students_signature	blob	YES			
tutor_signature	blob	YES			

MOD_assignment_receipt

Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
module_code	varchar(10)		PRI		
academic_year	int(1)		PRI		
semester	varchar(15)		PRI		
assignment_title	varchar(255)	YES			
marking_tutor	varchar(10)	YES			
submission_type	varchar(10)	YES			
hand_in_date	date	YES			
due_date	date	YES			
tutor_comment	blob	YES			
tutor_signature	blob	YES			
tutor_signature_date	date	YES			

MOD_module_assessment					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
module_code	varchar(10)		PRI		
academic_year	int(1)		PRI		
semester	varchar(15)		PRI		
q1	int(1)	YES			
q2	int(1)	YES			
q3	int(1)	YES			
q4	int(1)	YES			
q5	blob	YES			

MOD_module_report					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
module_code	varchar(10)		PRI		
academic_year	int(1)		PRI		
semester	varchar(15)		PRI		
module_report	blob	YES			
tutor_signature	blob	YES			
report_date	date	YES			

MOD_module_tutors					
Field	Type	Null	Key	Default	Extra
module_code	varchar(10)		PRI		
tutor_code	varchar(10)		PRI		
tutor_role	varchar(100)		PRI		

MOD_student_modules					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
module_code	varchar(10)		PRI		
academic_year	int(1)		PRI		
semester	varchar(15)		PRI		

REFLECT_skills_development					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
student_number	varchar(9)		PRI		
set_date	date		PRI	0000-00-00	
completed_date	date	YES		0000-00-00	
introduction	varchar(255)		PRI		
question_number	char(2)		PRI		
question_text	varchar(255)	YES			
question_feedback	blob	YES			

STA_address					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(30)		PRI		
add_desc	varchar(80)		PRI		
address1	varchar(255)				
address2	varchar(255)	YES			
address3	varchar(255)	YES			
phone	varchar(40)	YES			
fax	varchar(40)	YES			
mobile	varchar(40)	YES			

STA_availability					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(10)		PRI		
day	varchar(20)		PRI		
time_from	varchar(5)		PRI		
time_to	varchar(5)		PRI		
location	varchar(255)				

STA_availability_tutorial					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(10)		PRI		
date	varchar(20)		PRI	0000-00-00	
time_from	varchar(5)		PRI		
time_to	varchar(5)		PRI		
location	varchar(255)				

STA_department					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(10)		PRI		
department_code	varchar(10)	YES			

STA_email					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(10)		PRI		
tutor_email	varchar(150)	YES			

STA_favourite					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(50)	YES			
favourite_location	varchar(255)	YES			
favourite_title	varchar(50)	YES			

STA_name_details					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(10)		PRI		
tutor_surname	varchar(50)	YES			
tutor_forenames	varchar(150)	YES			
tutor_title	varchar(10)	YES			

STA_notebook					
Field	Type	Null	Key	Default	Extra
tutor_code	varchar(10)		PRI		
student_number	varchar(9)		PRI		
academic_year	int(1)				
date	date		PRI	0000-00-00	
time	varchar(5)		PRI		
category	varchar(255)		PRI		
notes	blob	YES			
last_update	date			0000-00-00	

STU_achievements					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
date	date		PRI	0000-00-00	
achievement	varchar(255)		PRI		
details	blob				

STU_address					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
address_type	varchar(15)		PRI		
address1	varchar(255)				
address2	varchar(255)	YES			
address3	varchar(255)	YES			
address4	varchar(255)	YES			
postcode	varchar(10)	YES			
country	varchar(30)	YES			
telephone	varchar(40)	YES			
fax	varchar(40)	YES			

STU_course					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
course_code	varchar(10)	YES			
academic_year	int(1)	YES			
ft_pt	char(2)	YES			
student_group	varchar(15)	YES			

STU_department					
Field	Type	Null	Key	Default	Extra
student_number	varchar(9)		PRI		
department_code	varchar(10)	YES			

STU_email					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
student_email	varchar(150)	YES			

STU_favourite					
Field	Type	Null	Key	Default	Extra
student_number	varchar(50)	YES			
favourite_location	varchar(255)	YES			
favourite_title	varchar(50)	YES			

STU_general_information					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
information_type	varchar(255)		PRI		
details	blob	YES			

STU_modules					
Field	Type	Null	Key	Default	Extra
student_number	varchar(9)		PRI		
module_code	varchar(10)		PRI		
academic_year	int(1)		PRI		
semester	varchar(15)		PRI		

STU_modules_planned					
Field	Type	Null	Key	Default	Extra
student_number	varchar(9)		PRI		
module_code	varchar(10)		PRI		
academic_year	int(1)		PRI		
semester	varchar(15)		PRI		

STU_name_details					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
student_surname	varchar(50)	YES			
student_forenames	varchar(100)	YES			
student_title	varchar(10)	YES			

STU_notebook					
Field	Type	Null	Key	Default	Extra
student_number	varchar(9)		PRI		
academic_year	int(1)				
date	date		PRI	0000-00-00	
time	varchar(5)		PRI		
category	varchar(255)		PRI		
notes	blob	YES			
last_update	date			0000-00-00	

STU_photograph					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
photograph_data	blob	YES			

STU_previous_courses					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
subject	varchar(150)		PRI		
course_type	varchar(30)	YES			
grade	varchar(10)	YES			
date	date		PRI	0000-00-00	

STU_tutor					
Field	Type	Null	Key	Default	Extra
student_number	int(9)		PRI		
tutor_role	varchar(255)		PRI		
tutor_code	varchar(10)		PRI		
academic_year	int(1)		PRI		

TIME_general					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
academic_year	char(1)		PRI		
start_date	date		PRI	0000-00-00	
end_date	date		PRI	0000-00-00	
time	varchar(5)		PRI	00:00	
appt_length	varchar(5)	YES			
category	varchar(255)		PRI		
date_text	varchar(255)				
event	varchar(255)		PRI		
preparation	varchar(255)	YES			
what_happens	blob	YES			
private	char(1)	YES			

TOOL_general					
Field	Type	Null	Key	Default	Extra
user_category	varchar(40)		PRI		
tool_category	varchar(40)		PRI		
tool	varchar(200)		PRI		
tool_description	blob				
tool_location	varchar(255)				
tool_icon_location	varchar(255)	YES			
list_order	char(3)				

TOOL_introduction					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
user_category	varchar(40)		PRI		
user_sub_category	varchar(100)		PRI		
identifier	varchar(9)		PRI		
tool_category	varchar(40)		PRI		
tool	varchar(200)		PRI		
introduction	blob	YES			

TOOL_name					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
user_category	varchar(40)		PRI		
user_sub_category	varchar(100)		PRI		
identifier	varchar(9)		PRI		
tool_category	varchar(40)		PRI		
tool	varchar(200)		PRI		
new_tool_name	varchar(200)	YES			
link_text	blob	YES			

TOOL_rubric					
Field	Type	Null	Key	Default	Extra
level	varchar(40)		PRI		
name	varchar(100)		PRI		
user_category	varchar(40)		PRI		
user_sub_category	varchar(100)		PRI		
identifier	varchar(9)		PRI		
tool_category	varchar(40)		PRI		
tool	varchar(200)		PRI		
rubric	blob	YES			

TRANS_record					
Field	Type	Null	Key	Default	Extra
student_number	varchar(9)		PRI		
course_code	varchar(10)		PRI		
academic_year	int(1)		PRI		
module_code	varchar(10)		PRI		
component	varchar(150)		PRI		
sub_component	varchar(255)		PRI		
grade	varchar(10)				
module_credits	varchar(4)				
qualification	varchar(255)				
qual_title	varchar(255)				
classification	varchar(20)				
date_of_award	date			0000-00-00	
date_of_issue	date			0000-00-00	
apl_apel	varchar(255)				
apl_apel_tot_credits	varchar(255)				

3.3 Appendix 3: Object Definitions

3.3.1 Student

Name id Student’s full name and unique identifier.

Name basic Student’s full name.

Name surname Student’s family name (capitalised), followed by given name, followed by title in brackets.

Id Student’s unique identifier.

Email Student’s email address.

Address Student’s home and term-time address information.

Dept The unique institutional code for the student’s department.

Tutor All tutors with whom the student is associated, academic, pastoral, senior, etc. (held as academic year and tutor code).

Current tutor Student’s current pastoral tutor (held as tutor code).

Course code The unique institutional code for the student’s course.

Full course The name of course and its code.

Academic year The student’s current academic year.

Interested year The academic year which is being shown in the *Internet-PAR*. Since the *Internet-PAR* can show past or future course information it needs to record which year it is currently viewing. If this value has not been set, it will default to the student’s current academic year.

Modules All of the modules on which the student has been enrolled, past and present (held as: year, semester, module code, details …).

Current modules Modules in the current academic year on which the student is enrolled (held in the same format as modules).

name_basic	Miss Susan Mary Graham
name_id	Miss Susan Mary Graham (100000010)
name_surname	GRAHAM, Susan, Mary (Miss)
id	100000010
email	S.M.Graham@ncl.ac.uk
address	[home, 7 Tudor Gardens, …], [term, Flat 45, …]
dept	D001
tutor	[1, 54321a12], [1, 54321j12], …
current_tutor	54321j12
course_code	Q309
full_course	English Language And Literature (N100)
academic_year	1
interested year	1
modules	[1, 1, ELL119, ...], [1,2,FRE800, …], …
current_modules	[1, 1, ELL119, ...], […], …

3.3.2 Tutor

Name id Tutor’s full name and unique identifier.

Name basic Tutor’s full name.

Name surname Tutor’s family name (capitalised), followed by given name(s), followed by title in brackets.

Id Tutor’s unique identifier.

Password Tutor’s password (held only in the working memory of the computer for the duration of the login to *Internet-PARs*).

Email Tutor’s email address.

Dept The unique institutional code for the tutor’s department.

Modules All of the modules with which the tutor is involved (held as: year, semester, module code, details …).

Tutees All tutees with whom the tutor is associated as an academic, pastoral, or senior tutor (held as the unique student identifier).

Title Tutor’s title.

Fname Tutor’s given name(s).

Sname Tutor’s family name.

name_id	Mr Paul Drummond (9507K054)
name_basic	Mr Paul Drummond
name_surname	DRUMMOND, Paul (Mr)
id	9507K054
password	junk
email	P.Drummond@ncl.ac.uk
dept	D100
modules	[1, 1, ELL119, ...], [1,2,FRE800, …], …
tutees	[100000001],[100000012],[100000034],…
title	Mr
fname	Paul
sname	Drummond

3.3.3 Administrator

Name id Administrator’s full name and unique identifier.

Name basic Administrator’s full name.

Name surname Administrator’s family name (capitalised), followed by given name, followed by title in brackets.

Id Administrator’s unique identifier (may be the same as their tutor identifier).

Email Administrator’s email address.

name_id	Mr Paul Drummond (npd12)
name_basic	Mr Paul Drummond
name_surname	DRUMMOND, Paul (Mr)
id	npd12
email	P.Drummond@ncl.ac.uk

3.3.4 Course

Course code The unique institutional code for the student’s course.

name_basic The name of course.

name The name of course and its code.

length The length of the course in years.

students_total The total number of students registered on the course.

students_1 The number of students registered on year one of the course.

students_2 The number of students registered on year one of the course.

students_n The number of students registered on year n of the course.

course_code	A106
name_basic	MBBS Medicine
name	MBBS Medicine (A106)
length	3
students_total	700
students_1	190
students_2	189
students_3	194

3.3.5 Department

dept_code The unique institutional code for the department.

name_basic The name of department.

name The name of the department and its code.

students_total The total number of students registered with this department.

students_1 The student number of all students in academic year one of their course.

students_2 The student number of all students in academic year two.

students_n The student number of all students in academic year n.

stages The maximum length in years of any course supported by this department.

```
dept_code       5170
name_basic      Management Studies
name    Management Studies (5170)
students_total  700
students_1      [912345678,912436583,932127645,…]
students_2      [912309678,916396583,988827645,…]
...
stages   4
```

3.3.6 Meeting

"last" Scheduled meetings occurring in the last academic year.

"next" Scheduled meetings occurring in the next academic year.

scheduled_meetings Scheduled meetings occurring in this academic year.

agenda Agenda items for all meetings.

requested_meetings Requested meetings.

action_planning Action planning notes for all meetings.

"last"	[meeting_date, tutor_code, academic_year, meeting_title], [meeting_date, tutor_code, …]
"next"	[meeting_date, tutor_code, academic_year, meeting_title], [meeting_date, tutor_code, …]
scheduled_meetings	[meeting_date, tutor_code, academic_year, meeting_title], [meeting_date, tutor_code, …]
agenda	[start_date, end_date, booked_date, time, agenda_type, agenda_item, preparation, agenda_item_subject, agenda_item_description, agenda_item_notes], [start_date, end_date, …]
requested_meetings	[date, time, tutor, subject], [date, time, …]
action_planning	[start_date, end_date, action_item, action_subject, action_description, action_notes, tutor_signature], [start_date, end_date, …]

3.3.7 Module

module_code The unique institutional code for the department.

name_basic The name of department.

name The name of the department and its code.

credits The credit value of the module.

assessment The assessment method for the module.

tutors The tutor code and role of all tutors involved in the module.

students A list of all students studying the module.

students_total The total number of students studying the module.

module_code	ACC302
name_basic	Accounting 101
name	Accounting 101 (ACC302)
credits	20
assessment	Exam
tutors	[97f362, academic], [7832g4, senior]
students	[912309678,916396583,988827645,…]
students_total	700

3.3.8 Tool

name Default institutional name for the tool.

name_glossary Aftermath of a GLOSSARY request on the above.

description Brief description of the tool to be used in links.

introduction Introduction to the tool.

rubric Rubric to contextualise the use of the tool.

component List of components and their access status, e.g. [component, access status].

icon Icon location.

location Location of tool, e.g. /cgi-bin/student/tutorials/tutorial.pl

name	Academic
name_glossary	Academic Record
description	View your academic record including ….
introduction	This tool allows you to …
rubric	Using your record you should be able to …
component	[modules, 1], [photograph, 0]
icon	/pictures/icons/student.gif
location	/cgi-bin/student/tutorials/tutorial.pl

3.4 Appendix 4: Data Consent

3.4.1 Newcastle Faculty of Medicine Data Protection Consent Form

The Medical Faculty administrative systems are supported by 'personal data' files. Personal data is processed for administration and support services in the Medical Faculty in strict accordance with the current Data Protection Act, by registering 'personal data' with the University's Data Protection Registrar. Examples of 'personal data' include:

UCAS application form
Registration details
Student photo
Vaccination records
Results of Assessment

The Medical Faculty will hold this information for the duration of your studies, and in an archived form for an indefinite period, allowing the Medical Faculty to provide authenticated confirmation of your qualifications to your future employers.

Information on the Data Protection Act 1984 and the Data Protection Act 1998 can be found on the World Wide Web at:

http://www.legislation.hmso.gov.uk/acts/acts1984/1984035.htm
http://www.hmso.gov.uk/acts/acts1998/19980029.htm

I consent to the use of my personal data for the purposes for which it is registered under the Data Protection Act 1998.

I understand and consent to the above.

Signed: ....

Date: ....

3.4.2 Newcastle University Data Protection Statement

UNIVERSITY OF NEWCASTLE UPON TYNE

USE OF STUDENTS' PERSONAL INFORMATION

The University needs to collect and retain certain types of data about its current and past students in order that it can properly fulful its functions as a provider of education and a degree-awarding body. In seeking to comply with the Data Protection Act 1998 and its basis of informed consent, this statement describes below the variety of purposes for which the information collected is used. The data are collected from your UCAS or University application form and at Registration, and are supplemented by information gathered during your period of study at the University relating to your academic progress (eg. examination marks).

Administration

- Students' personal data are used for many administrative purposes within the University including registration with the University Computing Service, Library and Careers Service, allocation of University accommodation, collection of fees and distribution of loan and grant cheques. The data are also used by academic departments which often display students' names on departmental noticeboards for the purpose of identifying class lists etc.
- All students need a Smart Card in order to use the University Library, Sports Centre, some Computing Service facilities and the students' Union. The computer system used to issue Smart Cards stores information about you that is needed for the production of the cards including your name and student number and a digital photograph.
- A photograph provided by you is attached to your personal file held in the Records Section of the Student Progress Office. This is used for identification purposes (for example, where students seek documents or advice from the Student Progress Office and do not have their Smart Card available).

Computer Systems

- Once you have registered with the Computing Service, your e-mail address may be found by staff and students through the University web server.
- Departments are increasingly making use of computer mediated communication tools for purposes such as listing or e-mailing students by degree programme. An on-line managed learning environment system is being developed within the University to help tutors and their students interact and record achievement. Where students have agreed to the use of their photograph in this system, the photograph is accessible only to the student concerned and to University staff.

Disclosure of Information to Outside Bodies

- Information about individual students may be disclosed by the University to outside bodies at the request of individual students themselves.
- The University will comply with requests from sponsors, Local Education Authorities, the Student Loans Company, Council Tax Registration Officers and Immigration authorities to confirm students' attendance, periods of registration etc.
- The University informs schools/colleges whose past students obtain first or second class degrees of the results of those students.
- In some cases the University may pass on information about individual students to debt collection agencies where attempts to recover debts to the University by internal processes have failed.

'Sensitive' Data

- The University collects information from students relating to ethnic origin and disability in order that it can carry out equal opportunities monitoring, offer advice and support to students with disabilities and manage its duties under the Disability Discrimination Act.
- The University is required to obtain information about past criminal convictions before offering a place on some programmes of study.

Alumni Association

- At the end of your studies you automatically become a member of the University's Alumni Association and a summary of your address and course details will be added to the University's alumni database.
- The Alumni Office keeps data on former students so that it can contact them with information about events, benefits and services for graduates and other matters of interest. The Alumni Office will contact you after graduation with full details. You will at that stage be given the chance to opt out of the alumni database, should you wish.

3.4.3 Newcastle student advisory leaflet - Personal Data and You

Next two pages illustrate a leaflet designed (two vertical folds) for distribution to Newcastle University students.

Personal Data and You

Introduction

In order to record your time and experiences at University, collect fees from your sponsors, and enable you to graduate, the University of Newcastle needs to process and maintain records which contain personal data about you. This includes the data which you provided during registration, such as:

- student number (UCAS number)
- title (e.g. Mr, Ms)
- given names and last name
- date of birth
- home department
- personal tutor(s)
- course (code and name)
- modules of study (code and name)
- address (local and permanent)
- photograph

The University adds a transcript of your grades and other relevant academic information during your course. Faculties and Departments 'process' subsets of your personal data in order to assign you a tutor, to study groups and administer the assessment and examinations process. This leaflet outlines the purpose of data processing, and your rights under the Data Protection Act 1984 and the Data Protection Act 1998 (the Acts).

Data Protection Acts 1984 and 1998

The Acts require the University to register descriptions of all electronic (and paper from 2007) personal data processing with the Registrar. All staff in the University who have a contract of employment are bound to respect the privacy and confidentiality of personal data and to 'register' use they make of personal data (see *http://www.ncl.ac.uk/internal/data.protection.html*). The Data Protection Act 1984 is being amended by the Data Protection Act 1998, principally to increase the rights of data subjects (you).

Your Rights

You have a right to ask who is processing electronic data about you, and to request sight of that data. You may request information in writing and for a small fee (currently £10.00) by completing a data information request form and returning it to the Registrar's Office, 6 Kensington Terrace. You will be provided with information about who holds records of personal data which might contain reference to you, and how the data is processed within 40 days.

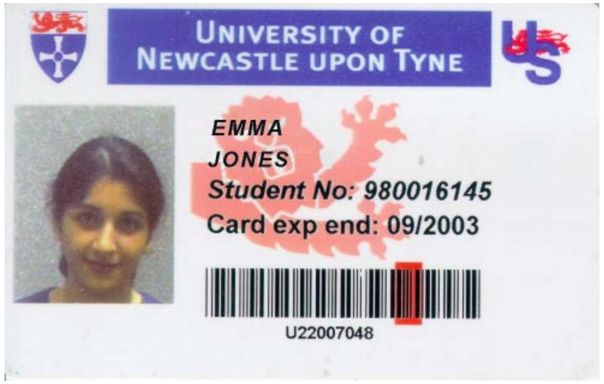
Accommodation, Library, Computing, Counselling, Careers, Alumni, Clubs and Societies

Some areas process additional data in order to provide access to services, and to administer your membership, interests and affiliations. This is intended to be of benefit to you. The University recommends that these areas hold only as much personal data as is absolutely necessary, and destroy out-of-date data.

For example, the University Computing Service holds a small subset of information about you in order to enable you to register for a computer user identifier and an email address, and record finance for printing. Your email address and your computer login identifier are available to members of the University (staff and students) by performing a mailname search on the University web server.

Smart Cards and Access

Smart cards contain a subset of your personal data which is used to authorise your access to University buildings and other services. Smart cards house personal data either on the card's magnetic strip, or in a central database which is read when the card is swiped. Smart cards include a photograph of you in order to prevent fraudulent use.

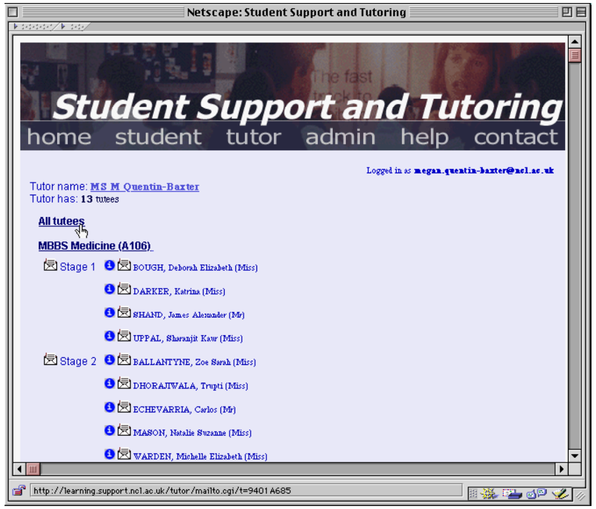


EMMA
JONES
Student No: 980016145
Card exp end: 09/2003
U22007048

Fig. 1: Smart cards identify you in order to give you access to University buildings and other benefits

Student Support and Tutoring (Computer Mediated Communication)

A subset of your registration data combined with your email address allows the University to run computer mediated communication tools, such as listing or emailing all students taking a particular module or degree programme.


Netscape: Student Support and Tutoring
Student Support and Tutoring
home student tutor admin help contact

Logged in as megan.quentin-baxter@ncl.ac.uk

Tutor name: MS M Quentin-Baxter
Tutor has: 13 tutees

All tutees

MBBS Medicine (A106)

Stage 1 BOUGH, Deborah Elizabeth (Miss)
DARKER, Katrina (Miss)
SHAND, James Alexander (Mr)
UPPAL, Sharanjit Kaur (Miss)
Stage 2 BALLANTYNE, Zoe Sarah (Miss)
DHORAJIWALA, Trupti (Miss)
ECHEVARRIA, Carlos (Mr)
MASON, Natalie Suzanne (Miss)
WARDEN, Michelle Elizabeth (Miss)

http://learning.support.ncl.ac.uk/tutor/mailto.cgi/t=9401A685

Fig. 2: Computer mediated communication tools ensure that students with email addresses who are registered on degree programmes and modules are copied into messages from staff and other students.

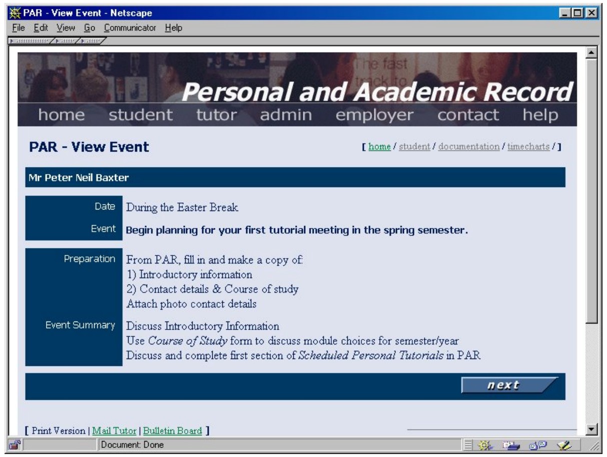
Support for Tutors

Tutors have access to shared communication facilities for booking times for personal tutorials and identifying agendas for discussion. The facilities may also contain the latest information about in-course assessment results, providing tutors with information to enable them to recognise achievement, and early warning of students in need of support.

Personal and Academic Records

Personal and academic records (PARs) are being developed to support you throughout your studies and to assist you to recognise and record achievement. Logging into the system provides you with access to your personal and academic information as well as timecharts and agendas for meetings with your tutor (*Fig. 3*). You can submit evaluations for modules and courses and record agreed outcomes and action plans resulting from tutorial meetings.

Tools are being developed to help you to recognise and evaluate a range of learning experiences, both within your course and in extra-curricular activities. These tools, in combination with quality time spent in face-to-face meetings with tutors, enable you to build up a reflective record throughout your course of study. From this resource you can construct and evidence a professional CV with ease, and interface with IT facilities in careers services.


PAR - View Event - Netscape
File Edit View Go Communicator Help
Personal and Academic Record
home student tutor admin employer contact help
[home / student / documentation / timecharts /]
next
[Print Version | Mail Tutor | Bulletin Board]
Document: Done

PAR - View Event

Mr Peter Neil Baxter

Date: During the Easter Break

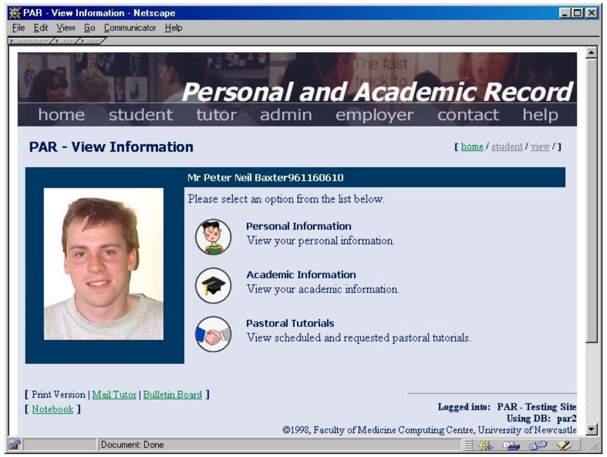
Event: **Begin planning for your first tutorial meeting in the spring semester.**

Preparation: From PAR, fill in and make a copy of:
1) Introductory information
2) Contact details & Course of study
Attach photo contact details

Event Summary: Discuss Introductory Information
Use *Course of Study* form to discuss module choices for semester/year
Discuss and complete first section of *Scheduled Personal Tutorials* in PAR

Fig. 3: Personalised timecharts inform students of events in the tutorial process giving details of any preparation required

Your personal data can be used to dynamically provide access to relevant learning resources and support services. In the future, preferences will allow you to control access to your personal data. These tools are secured by ensuring that you must login (you should never share or allow someone to learn your login or password).


PAR - View Information - Netscape
File Edit View Go Communicator Help
Personal and Academic Record
home student tutor admin employer contact help
[home / student / view /]
[Print Version | Mail Tutor | Bulletin Board]
[Notebook]
Document: Done

PAR - View Information

Mr Peter Neil Baxter961160610

Please select an option from the list below.

Personal Information
View your personal information.

Academic Information
View your academic information.

Pastoral Tutorials
View scheduled and requested pastoral tutorials.

Logged into: PAR - Testing Site
Using DB: par2
©1998, Faculty of Medicine Computing Centre, University of Newcastle

Fig. 4: Through robust authentication each user has access only to their own records. All information provided is personalised and the display looks and feels like the user's own web space

Research

Where personal data is collected for research purposes the University aims to ensure that staff and students are aware of their obligation to register it with the Data Protection Officer, and to anonymise personal data wherever possible.

Consent

Although you are not *required* to consent to the University processing your personal data, it would be impossible for the University to recognise your studies and reward you without doing so. In addition, a number of benefits arise such as access to services, communication, information and support. Staff and students in the University are working together in order to comply with the law, and to keep you informed of issues relating to data protection

3.5 Appendix 5: Evaluation of the use of the software

3.5.1 Breakdown by discipline of staff and students involved in piloting

Nottingham Internal Evaluation

Nottingham evaluation strategy

There have been two stages within the strategy.

1. It was established from the outset that the Nottingham evaluators would have a development brief, liasing frequently between
   - departments/schools piloting the PAR
   - piloting departments/schools and the Newcastle team.

Rapid collection and logging of feedback from Nottingham “users became the main focus for the Nottingham Project team discussions and in turn informed regular discussions with the IT development team at Newcastle. Modifications of the IT tool were thus made in response to user feedback. On the Nottingham campus liaison took the form of

- scheduled meetings which brought together all the departmental contacts ("administrators")
- scheduled discussions between a Project team representative and staff and students in each pilot department
- informal contact in response to departmental demand
- informal contact initiated by the Project team in response to emerging issues, both local and national

2. In the latter stages of the Project, after consultation with the Users, the team agreed to provide a quantitative and qualitative data for the final report to the DfEE as follows:
   i. Numbers of piloting staff and students within each department/school as a percentage of the total number of staff/students in the relevant cohort; information about the existence or absence of any pre-existing PAR before the department/school joined the pilot.
   ii. Reactions from students to the experience of using the PAR; this would be derived from a detailed questionnaire and from discussions with small groups of students.
   iii. Reactions collected from discussions with administrators and other piloting tutors.

Main findings

(i) *Quantitative information*

Department	Staff	Students	Previous PAR?
School of Biological Sciences (Animal Sci)	4/50 [8%]	37/350 [11% of Y1/2 Single Hons]	Yes
SOBS (Botanical Sci)	7/22 [34%]	73/230 [40% of Y1/2 Single Hons]	Yes
Art History	2/4.5 [50%]	22/52 [42% of Y1 Single Hons]	No
Modern European Studies, Archaeology and History	1/1 [100%]	12/12 [100% of Y1 Joint Hons]	No
Economics	3/33 [9%]	35/380 [9% of Y1/2 Single Hons]	Yes – early stages
Education PGCE	2/20 [10%]	12/300 [4% of postgrads on 1 year course]	Yes – comprehensive Record of Prof. Dev.
Mathematics with Physics	2/2 [100%]	17/17 [100% of Y1 on the combined course]	Yes – but separate PAR for each dept.
Environmental Engineering	12/12 [100%]	45/45 [100% of Y1]	No
Physiotherapy	11/11 [100%]	40/40 [100% of Y1]	Yes
Medicine	4/100 [4%]	32/400 [8% of Y1/2	No

(ii) *Qualitative information*

i. ***Student reactions to the PAR***

The information which follows is derived from discussions held with students by members of the Project team during Autumn 1998

- an analysis of a questionnaire [See Appendix 3.5.2] completed by Physiotherapy students (100% returned) and Medicine students (25% returned) in February 2000
- discussions with small groups of students from Economics, SOBS and Education in February 2000.

After some discussion of the scope of a PAR and a demonstration of the IT-PAR in its early stages, students were asked to consider what such a record might provide to enhance their experience of learning and their personal development. Their *suggestions* were

- published tutorial agendas
- the facility to add their own items
- the facility to request extra personal tutorials
- encouragement to reflect on subject-specific skills
- an opportunity to provide material for reference writing
- feedback on assignments, modules, courses
- improved continuity between successive tutors
- information on degree pathways and implications of module choices
- group feedback from lecturers

Their *concerns* were

- security
- inadequate IT access/reliability of the network
- training for use
- on going support

Results from the questionnaire indicate how far the design of the IT-PAR has met students' needs.

How helpful was the way in which students were introduced to the PAR?

Students commented positively on

- proper group tutorials or "lessons" on its scope and uses
- step-by-step logical explanations
- clear instructions/demonstration
- staff and peer support

Although they were not specifically asked to do so, students (especially the Physiotherapists) also commented here on what they immediately identified as positive aspects of the PAR itself, suggesting that staff had placed considerable emphasis on developing students' awareness of its wide-ranging potential. For example

- recording achievements and progress
- a permanent record of thoughts
- easy to re-visit what you have written
- capacity to handle updating
- could improve personal organisation/help you keep up to date
- long-term advantages
- improving IT skills and awareness
- drawing attention to the School's IT provision
- supporting the tutorial process
- continual access to information for tutor and student
- everything stored in one place; no papers to lose
- easy to follow format
- encourages you to think about objectives and to plan (eg to become involved in the local community)
- keeping in touch with your tutor

Could the introduction have been improved in any way?

The main problems for Physiotherapists were

- access to the system seemed fiddly; students wanted to get into the PAR with their email password - too much information in weeks 1 and 2 of a new course
- unreliable computer systems
- some needed more time to practise after the demonstration
- some wanted the introduction delivered to smaller groups/needed a "mentor"
- some felt that their very basic IT skills were over-estimated
- although staff explanations were clear, some students would have liked a written sheet of instructions to take away; this could also have been used to record their password.
- some wanted the system to remind them to use it after a tutorial
- knowing exactly what was appropriate to include - provision of a "model" suggested

Half the group reported complete satisfaction with their induction into the PAR.

The main problems for Medicine students were

- password difficulties
- logging in
- not having access from the university's home page (they conceded this had had been rectified)
- some wanted more information on the exact uses of the PAR
- one would have liked a "proper tutorial" on it; it emerged later in the questionnaire that this student had "hardly used it"

Students were asked to consider which of the various functions of the PAR would be most important to them - and to gauge their current success.

The most popular functions were

- making it easy to make tutorial appointments
- viewing tutorial agendas beforehand
- requesting extra tutorials
- providing the department with introductory information
- offering two-way feedback between student and tutor

There was also relative enthusiasm for

- adding their own agenda items
- preparing for tutorials
- recording outcomes
- discussing module choice (less relevant for Medicine students)

About half the students in each discipline saw the PAR's potential as a means of following up outcomes from tutorial discussions.

Suggestions for improving the PAR's current functions were

- make it clearer for first-time users (perhaps with the help of a page of instructions/account of its functions)
- make it more accessible (more reliable computer network; email link and password)

Students' comments on how successfully these functions were working at present have been passed on to the Newcastle team.

Other functions students would like within the PAR were

- examination feedback and results

- class averages
- lecturer feedback on assignments with suggestions for improvements
- tutor feedback on general progress
- guidelines for future use
- hospital visit agendas (Medicine)
- access to the Network Learning Environment (Medicine)
- dates and times when tutors were on site
- a Favourites facility for noting useful websites
- peer/lecturer help (bulletin board)
- closer link with email to enable them to receive messages relating to teaching arrangements
- a CV building facility

Students were asked to consider how the PAR might relate in other ways to the experience of studying at university

They were positive about its capacity to

- show how the tutorial system would support them over the length of the course
- help them to use it actively
- give them an overview of their degrees
- and of their whole university experience
- provide support (from fellow students, lecturers, resources)
- encourage target setting

They were relatively positive about its capacity to

- define the roles and responsibilities of tutors and tutees
- help improve their own learning
- enable them to audit their subject-specific skills

There were relatively few students who were interested in its capacity to help them audit or develop key skills.

Suggestions for improvements were

- CV prompts
- Encouragement to check progress on targets

The Internet-PAR has three main aims.

Students were

a. generally positive about it as a formal record of academic progress

b. evenly divided on its usefulness as a reflective record of personal development and skills

c. rather unconvinced on its usefulness as a reflective record of additional skills and competencies. developed by extra-curricular work or activity

The relatively low interest in (c) was more marked among Medicine students.

To be more useful, they suggested the PAR should offer

- more emphasis on analysing weaknesses and setting targets
- a subject specific checklist to help assess coverage
- prompts to encourage regular self assessment
- clarification on how to seek support for developing communication skills

- more focus on personal development
- community action links (Physiotherapy)

Fewer than half the students had previous experience of recording and reviewing progress. Asked to identify this, they mentioned

- NRA
- 6th form/college reviewing and target setting
- preparation for UCAS
- NHS workplace personal development forms (Physiotherapy)

The main differences noted between such examples and the PAR were

- IT (but no comment on how this might affect the process of reviewing)
- the PAR has more features and is more complicated
- filling in paper forms seemed simpler and quicker
- the PAR is more process based

Students were asked to assess the impact of the PAR on their skills in using and attitudes towards IT

Skills

- large numbers of Physiotherapy students commented on a positive improvement and
- some welcomed having to use IT for a key element so early on
- half the Medicine students felt it had had little or no impact on their IT skills and
- a quarter of them reported an improvement.

Attitudes

- Medicine students divided evenly on whether use of the PAR had made their attitude to IT more positive
- there was evidence to suggest that their frustrations over inadequate numbers of computers/problems with the network/slow PCs were adversely affecting their attitudes
- there was some frustration, too, among Physiotherapy students, but fewer reports of difficulties
- many commented favourably on how the PAR had demonstrated an effective application of IT and
- broadened their outlook
- some found the use of IT time consuming and felt they got little back for their efforts
- others commented that their attitude to IT was directly related to their success or failure in using the system

Students were asked to comment on ease of operation and the visual impact of the PAR

- about 80% found it easy or not too difficult to navigate their way around the system
- over half commented favourably on the icons.

These findings were confirmed by discussions with small focus groups.

ii. ***Staff reactions to the PAR***

It has been evident that, in spite of goodwill, enthusiasm and willingness to take part in discussions to inform the design process, staff have had rather less time than students to persevere with the system when it has failed to work straightforwardly in the early stages. There has been little incentive to continue trying to use it to arrange tutorial appointments when it is easier and quicker to do this by phone, notice board or email. This has made it difficult to gather detailed responses to the *use* of the

PAR within the time-scale of the Project.

Scheduled discussions with representatives from each piloting school/department (administrators) did however prove most useful throughout the Project in ensuring that the IT tool was designed to meet a wide range of needs.

3.5.2 Questionnaire and student responses to Internet-PARs

Student questionnaire

Course code:

Degree:

Year.

[Male] [Female]

Mature student? [Yes] [No]

Full-time student? [Yes] [No]

You are taking part in a trial for an Internet-based Personal and Academic Record (PAR), which has been designed to support students' personal and academic progress during their degree Your initial experiences within this pilot will be helpful to us in modifying and developing the IT support aspect of personal tutoring.

1. **Think about how you were initially introduced to the Internet-PAR.**

   What were the positive aspects of this part of the process?

   Is there anything that you would like to have happened differently?

2. How *important* to students do you think each of the functions are?
How successful is the Internet-PAR system at the moment in meeting these needs?
(Please insert score in the table, <u>using 4 as high and 1 as low)</u>

	Importance	Success
Making it easy to make tutorial appointments		
Viewing the agenda for personal tutorials		
Adding agenda items for personal tutorials		
Preparing for tutorials		
Recording the outcomes of personal tutorials		
Following up outcomes		
Recording the discussions regarding module choice		
Requesting extra personal tutorials		
Providing two-way feedback between student and personal tutor		
Providing the department with relevant introductory information about yourself		

What other functions would you like the Internet-PAR to offer?

Do you have any suggestions for improving the functions it currently offers?

3. The Internet-PAR will relate in other ways to students' experience of studying at university.

How would you rate its potential usefulness in the following areas?
(*Please score* *using 4 as high and 1 as low)*

- Providing information about the scope of the personal tutor system within your department/school
  1 2 3 4

- Letting you know when meetings would take place and what they would cover
  1 2 3 4

- Preparing you to take an active part in these meetings
  1 2 3 4

- Defining the roles and responsibilities of tutors and tutees
  1 2 3 4

- Encouraging you to improve your own learning
  1 2 3 4

- Enabling you to set useful and realistic targets
  1 2 3 4

- Helping you to take stock of the skills you are acquiring - subject specific
  1 2 3 4

- Helping you to take stock of the skills you are acquiring - general (e.g. working with others; ICT)
  1 2 3 4

- Putting you in touch with other people who might help you (staff, students, others)
  1 2 3 4

- Drawing your attention to useful learning resources
  1 2 3 4

- Encouraging you to take an overview of your degree programme
  1 2 3 4

- Encouraging you to take an overview of your university experience as a whole
  1 2 3 4

Do you have any suggestions for improving on any of the above?

4. The aims of the Internet-PARs are to provide

a) a formal record of academic progress (degree classification and transcript of modules)

b) a reflective record of personal development - and academic skills

c) a reflective record of additional skills developed whilst at University e.g. work experience, study abroad, community service or Student Union activities.

From your experience so far, how well is the system overall currently meeting these aims?

(Please score using 4 as high and 1 as low)

a) A formal record of academic progress (transcript of modules)

1 2 3 4

What suggestions do you have for improving it?

b) A reflective record of personal development and academic skills

1 2 3 4

What suggestions do you have for improving it?

c) A reflective record of additional skills developed whilst at University

1 2 3 4

What suggestions do you have for improving it?

5. Does this process (as outlined in question 4 above) resemble anything you have previously experienced (whether IT based or not)?
If so, in what ways does it differ?

6. What impact do you think your involvement with the Internet-PAR is having on:

- your IT skills?

- your attitude towards IT?

7. Finally, we would welcome some feedback on the practicalities of using the IT PAR.

Approximately how long did it take you the first time you used it?

Has this differed with further use?

How easy/difficult is computer access?

Where do you use the computer?

How easy/difficult is it to "navigate" around the system?

What is your reaction to the visual icons?

Thank you very much for your time.

3.6 Appendix 6: List of Universities linked with PADSHE and Internet-PARs projects

3.6.1 *Universities committed to institution-wide implementation:*

University of Newcastle	University policy led by	
	Prof Madeleine Atkins	Pro Vice Chancellor
University of Nottingham	University policy led by	
	Prof Stephen Bailey	Pro Vice Chancellor

3.6.2 *Other PADSHE universities, interested in institution-wide implementation:*

University of Birmingham	Prof Frances Young	Pro Vice Chancellor
	Philip Denner	Academic Secretary
City University	Prof Chris Haines	Pro Vice Chancellor
University of Liverpool		
John Moores	Dr Phil Margham	Pro Vice Chancellor
	Dr Steve Jackson	Assistant to the Provost
University of Northumbria	Prof Monica Shaw	Dean of Social Sciences
University of Reading	Prof Michael Fulford	Pro Vice Chancellor
	Roger Clark	Registrar

3.6.3 *Other universities interested in institution-wide implementation:*

University of Leeds	Dr Jim Parry	Dean for Students

3.6.4 *Further universities where senior manager is aware of / interested in the projects:*

Belfast, Queen's	Prof Angela Wilcox	Pro Vice Chancellor
	(May 98 she was Enterprise Coordinator)	
Dundee, Abertay	Dr David Button	Registrar
Durham	Dr BS Gower	Pro Vice Chancellor
Hertfordshire	Prof Thomas Hanahoe	Pro Vice Chancellor
Manchester	Prof Philip Gummett	Pro Vice Chancellor
Middlesex	Prof Ken Goulding	Deputy Vice Chancellor
Nottingham Trent	David Samson	Registrar
Sheffield	Prof PA Jones	Pro Vice Chancellor
Southampton	Prof Roy Farrar	Deputy Vice Chancellor
Stirling	Prof Sally Brown	Deputy Principal
Wolverhampton	Jon Baldwin	Registrar

3.6.5 *Further universities where a central service department is aware of / interested in the project:*

Bath	Dr John Harris	Senior Assistant Registrar
Bradford	Dr Tony Pickles	
Cambridge	Jayne Williams	Administrative Officer
Coventry	Lynn Martin	Educational Development Unit
Essex	Terry Barry	

Greenwich	Henry Hill	Assistant Registrar
Hull	Marilyn Howarth	Information Manager
UfI	Dr Ian Goodwin	Qualifications & Assessment
	Dr Brian Sutton	Director of ICT
Kent	Dr Jeremy Overden	Senior Assistant Registrar
Leicester	Annie Grant	Director, Educational Development
Liverpool	Dr Janet Strivens	Student Services
	Dr G Walker	Senior Assistant Registrar
London, University College	Maureen Donelan	HE Research & Development Unit
Manchester	Dr Catherine O'Connell	Enterprise Centre
	Maddie Smith	
UMIST	Rosemary Warner	Staff Development
Middlesex	Chris Osborne	Quality Assurance & Audit Service
	Barbara Workman	
Oxford	Catherine Godman	Senior Assistant Registrar
Oxford Brookes	Keith Cooper	Head of Student Services
	Stuart Brown	Head of MIS Student Records
Plymouth	June Harwood	Educational Development
	Graham Fice	Deputy Registrar
Portsmouth	Joy Moloney	Staff Development
	Jackie Moore	
	Pat Morris	
	Ian Kendall	
Sheffield Hallam	Rosie Bingham	
Southampton Institute	Andrew Comrie	Student Development
	Michael Marrinan	
Staffordshire	Siobhan Holland	
Sunderland	Linda White,	
	Lynne Humphries,	
	Anne Burlinson,	
	Mike Wyvill	Quality Unit
Sussex	Ben Ball	Teaching & Learning Development
	Lilla Funge	
	Anne Johnson	Assistant Registrar
Ulster	Joan Mulholland	Edducational Development Unit
	Sam Bailie	
UWE	Rod Coleman	Modular Scheme Director
	Paul Cox	Careers Service
Wales, Newport	Gareth Davies	BTec Licence Officer
Warwick	Dave Hall	Senior Assistant Registrar
	Cheryl Turner	Student Services
Charles Sturt (NSW, Australia)	Mike Rebbechi	Executive Director of IT

3.6.6 Further universities where an academic school is aware of / interested in / using the project:

UEA	Clare Jones	
Edinburgh	BD Barron	Vice Dean, Faculty of Arts
Edinburgh, Napier	Jenny Westwood	
Kingston	Katrina Luker	
	Sheila Browning	
	Demetra Katsifli	
	Marion Webb	
Leeds Metropolitan	Dr Jago Morrison	School of Cultural Studies
LSE	Liz Barnett, Mike Tiley	
Loughborough	Alan Maddocks	Dept of Civil & Building Eng.ing
Manchester Metropolitan	Pauline Hearn	

Plymouth	Brian Chalkley	Dept of Geography
	M Powdenley	
Salford	Iain Jones	
	Carena Eaton	
Sheffield	Paul Fennell	
Staffordshire	Prof RW Painter	Dean of Law
Wales, Cardiff	Carol Nicholas	
Wales, Swansea	Frank Slater	
Wolverhampton	Steve Reynolds	Applied Sciences

Internet-PARs was demonstrated at a meeting of Medical School Secretaries and Registrars held at Newcastle University, January 1999, to senior academics and administrators from the following medical schools/institutions:

> Belfast, Bristol, Cambridge, Dundee, Edinburgh, Glasgow, Institute of Cancer Research, Leeds, Nottingham, Royal Free, Sheffield, Southampton, St Bart's & Royal London, St George's, and UCL.

Internet-PARs has been specifically demonstrated to a number of senior medical educators:

> Prof John Hamilton, Dean of Medical Studies, University of Durham
> Dr Jeannette Murphy, CHIME, UCL
> Prof David Newble, Sheffield University
> Prof Bert Jan Vermeer, Dean of Medical School, Leiden (with several colleagues)

3.6.7 University Colleges which have participated in PADSHE events:

Bolton Institute	Sue Burkinshaw
Bretton Hall College	Rebecca Grattan
	Annabelle Hardcastle
Chester College	Wendy England
Edge Hill College of HE	Joy Bowes
London College of Arts	Alex Lumley
Ripon and York St John	Dr Tim Middleton
Suffolk	Marc O'Day, School of Arts
Winchester King Alfred's	Ann Henderson
	Ann Keating
Worcester	John Peters, History

3.6.8 FE Colleges

Bradford and Ilkley	Howard Stones	
Gateshead College	Muriel Callaghan	
Nottingham, Bilborough College	Dorothy Mountford	Vice Principal
	Martin Slattery	Principal
Nottingham, New College	Keith Dennis	Vice Principal

3.6.9 Other organisations

CVCP contact:	David Young	Policy Adviser
SCoP contact:	Patricia Ambrose	Executive Secretary
QAA contact	Dr Norman Jackson	Development Directorate
UCAS	Jill Johnson	
Northern Grid for Learning	John Harrold	Development Consultant

3.6.10 CPD links

BAe Systems	Dr Andrew Pember	Manager Learning Networks
RIBA	Pamela Edwards	

3.6.11 Companies – software and publishing

ANCOMM	David and John West	Directors
Harcourt Health Sciences, Europe	Ellen Green	Commissioning Editor
Leighton Internet and Multimedia	Paul Callaghan	Managing Director
TNL	Kevin Curran	Chief Executive

3.7 Appendix 7 – Evaluation report

July 2000

HIGHER EDUCATION DEVELOPMENT PROJECT

AN INTERNET-BASED COMUNICATIONS AND INFORMATION TECHNOLOGY (C & IT) SYSTEM FOR THE DEVELOPMENT OF PERSONAL AND ACADEMIC RECORDS FOR HE (INTERNET - PAR)

**Led by the University of Newcastle
in partnership with the University of Nottingham**

FINAL EVALUATION REPORT

Introduction and Methodology

This evaluation work has taken place over five days from June 1999 - March 2000. Activities have included attendance at two Steering Committee meetings at the University of Newcastle and two visits to the University of Nottingham to meet with staff and students involved in the pilot work. On the last visit on 1 March, discussions took place with Angela Smallwood, the Nottingham project director; the two internal evaluators, Gill Murray and Cliff Litton; and two focus groups - one of students, the other of the academic staff from eight of the pilot departments. These were Medicine, Art History, History, Economics, Biological Sciences, Engineering, Physiotherapy and Education.

It should be pointed out that I was the external evaluator for a previous University of Nottingham PADSHE project and therefore familiar with the rationale and foundation work for the development of an internet PAR.

Wherever possible, I have used a facilitative approach. My role as the external evaluator for the Nottingham site was to attend to the process of using this PAR and monitor the pedagogical implications. Evaluation of the technical aspects of the development was to be done at Newcastle.

Results in the Context of Achievement of Project Aim and Objectives

Aim

It had been envisaged in the original contract that three departments would be involved at Nottingham in the first year. Following modifications to the prototype shell, other departments where an IT-based process was favoured would be invited to participate within the two universities. Eventually, eight departments across all disciplines became involved at Nottingham. Piloting did not take place at Newcastle.

The main deliverable - the internet PAR - is still being further developed. At this stage, an interactive shell has been produced and many technical problems have been overcome. At the moment, a system for arranging the time and agenda for tutorials is completed. Student input on the outcomes of the tutorial and use of the PAR as a reflective tool requires further modification. The design has been under constant review by the end-users involved in piloting. They have been vigilant in reporting to the design team on accessibility, ease of navigation and interface appeal. The response to this feedback by the team at Newcastle has been satisfactory. The staff at Newcastle intend to resolve any outstanding technical problems so that the next cohort of students will be able to access a robust product as envisaged in the original project proposal.

The level of interest from other HEIs is very high. It is anticipated that the product will be sufficiently developed to be further modified for subject-specific requirements or ‘owned’ by the HE sector.

Objectives

With reference to the anticipated objectives of the project proposal, providing the necessary technical amendments to the PAR are completed, the pilot groups should be able to complete a reflective record of learning and achievement for the final semester. I am aware of how important it is that both staff and students in these pilot groups are able to proceed confidently into the next academic year with a commitment to the internet PAR.

It would have been helpful to have a clear line-management policy in place from the outset. I suspect a lack of clarity in this is an inevitable consequence of projects organised on a consortium basis. Perhaps the steering group has a major role to play in being pro-active in this matter.

Lessons Learnt From This Project

The academic staff, project director and internal evaluators have met regularly to monitor progress at every stage of development. For the remainder of this report, I should like to outline the major lessons learnt, many of which were discussed in the staff focus group. These should offer guidance to those academic staff involved in dissemination and to others new to the implementation of an internet PAR.

- Despite the expectation that internet communication between the design team and the pilot group would alleviate any problems or delays in piloting, this has not, in fact, been the case. On reflection, it would have been helpful to have at least one student group at Newcastle, and therefore have face-to-face contact between the designers and academic staff. Opportunities for in-depth discussion about the technical problems encountered on a daily basis would have been beneficial to all concerned.

- Although the academic staff of the leading pilot departments were tolerant and resilient to the technical problems, they were concerned that their less committed colleagues would require a fully operational product before using such a PAR. However, those involved at the development stage are convinced that their contribution has been essential for ownership of the system for future dissemination work at Nottingham.

- Anecdotally, there was a belief that the students are more receptive to using an internet PAR than many academic staff. It was recommended that wider-scale implementation should include appropriate induction sessions and backup support for tutors.

- Variations between different pilot groups across departments in overcoming the frustration resulting from technical setbacks may reflect the culture of the discipline. Willingness to tolerate ambiguity may relate to expectations of 'perfect systems' in, for instance, pure science. For those planning implementation in the HEIs, this might be a factor to consider in the initial choice of pilot departments. It may also be beneficial to restrict the number of groups involved to three or four departments. There was a view that it would be difficult to sustain commitment to the development if more staff and students were exposed too early to frequent changes in the medium for arranging and documenting the outcomes of tutorials through an IT-based process.

- There is an unresolved issue around the degree of flexibility of the tool. So far, opinions are divided between the need for flexibility in order to customise it for a given department and the need to have a tool which is sufficiently standardised to be evaluated. Some tutors commented that too much flexibility could hamper the pilot work because of the number of decisions which have to be made. I see this as another facet of the irreconcilable nature of development work undertaken by several groups, all of whom already have a heavy workload.

Summary

Despite the technical difficulties intrinsic to a development project such as this, there is agreement between key staff that a great deal has been learned. Furthermore, the learning can be used by those involved in future projects of this nature where the primary medium of communication is electronic. Transferability of learning from the development of the previous Nottingham PAR has been limited, but the academic staff agree that, pedagogically, their previous experience was important in order to clarify the aims and essential processes of a PAR in Higher Education.

To date, it has been possible to design an attractive interface which can record academic progress and arrangements for tutorial meetings between tutor and student. The next challenge will be to consolidate development of a reflective record of academic, personal and skill development.

Signed **Marjorie G Allen**
Education Adviser to the Postgraduate Dean
The Department for NHS Postgraduate Medical and Dental Education (Yorkshire)
Willow Terrace Road
University of Leeds
Leeds LS2
Date **31/07/2000**

3.8 Appendix 8 – CD-ROM of software

A CD-ROM has been prepared for distribution as part of the Final Project Report which has been attached to the back cover of some copies of the report.

This CD-ROM provides copies of all of the scripts developed in the project along with a demonstration dataset and basic information relating to the development environment for the *Internet-PARs* project. It does not represent an installation version of the software and it is strongly recommended that anyone interested in further developing the *Internet-PARs* concept described in the report should contact the IT Development Team at the Faculty of Medicine Computing Centre at the University of Newcastle upon Tyne for more detailed instructions on the setup and operation of the scripts contained on the CD-ROM.

If your copy of the report does not contain a CD-ROM and you would be interested in replicating the development environment and testing / further developing the scripts in your own institution, please contact the Development Team (*Internet-PARs* Project, Faculty of Medicine Computing Centre, The Medical School, Framlington Place, Newcastle upon Tyne, NE2 4HH) and we will arrange for a CD-ROM to be sent to you.