INTERACT
INTEGRATE
IMPACT

Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE)

Adelaide, Australia
7–10 December 2003

Editors
Geoffrey Crisp, Di Thiele, Ingrid Scholten, Sandra Barker, Judi Baron

ascilite
**Australasian Society For Computers
In Learning In Tertiary Education**

MAPPING ACADEMIC PROGRAMS
WITH PROGRAMMAP

Ian W Roberts, Susan J. Shannon and Antony Radford
School of Architecture, Landscape Architecture and Urban Design
The University of Adelaide, AUSTRALIA
{Ian.W.Roberts, Susan.Shannon, Antony.Radford}@adelaide.edu.au

Abstract

ProgramMap is a web tool that allows curriculum leaders to describe the relationships amongst components of academic programs and between those programs and one or more curriculum frameworks. Information including examples of student work are collected at course-level and aggregated to year-level and program-level as required. Curriculum leaders are led to work within self-declared and externally imposed academic objectives. The design and delivery of integrated programs are thereby fostered and evaluation against strategic objectives enhanced. Compliance with internal and external requirements can be monitored and current and prospective students can discover valuable information about the programs they are undertaking or considering. This paper describes the design and production and implementation of ProgramMap in the School of Architecture, Landscape Architecture and Urban Design at the University of Adelaide.

Keywords

curriculum design, curriculum analysis, planning, evaluation, curriculum framework, integrated curriculum, graduate attributes, course objectives

Introduction

A commitment to quality, a focus on learning outcomes and a continued pursuit of accountability remain major current themes in tertiary education in Australia. Recently, Minister Brendan Nelson has called for a 'renewed emphasis on teaching and learning outcomes' to ensure relevance for students in their future life and careers (Nelson 2003 p11). Further pressure is applied to curriculum leaders institutionally with the articulation of internal academic planning objectives (UofA 2001).

Particularly in professionally accredited Faculties, external expectations oblige academic programs to comply with multiple sets of requirements. For instance, in the *School of Architecture, Landscape Architecture and Urban Design*, the education policies and accreditation requirements of the *Royal*

Figure 1: Student construction models displayed in ProgramMap (Bachelor of Design Studies - Construction I in Semester II 2002) by David Gregory, Tsang Pui-Chi and Mark Hinchcliff respectively.

Australian Institute of Architects (RAIA 2000) and the *Australian Institute of Landscape Architects* (AILA 2002) represent separate, overlapping curriculum frameworks to which our programs must comply. In addition, statements of graduate attributes endorsed by the School and University must also be accommodated (Boumelha 2001; SALAUD 2001). Although deriving from existing policies, external accreditation procedures add further complexity for the School (RAIA 1997). Finally, the reasonable expectations of students to know in advance the nature and circumstances of learning activities and work to be submitted add more demands.

The School recognised that a tool configured with necessary curriculum frameworks would facilitate the design, comprehension and evaluation of our programs with concomitant benefits for all stakeholders. An investigation of commercially available products revealed none that would meet our requirements. Recent entrants to the field, including Myoporum Pty Ltd, with their Graduate Attributes Program are limited to single frameworks, require programs to be described in excessive detail and do not allow convenient aggregation to program level.

ProgramMap was conceived as a tool that would allow academic programs to be efficiently described in one or more curriculum frameworks and to be represented in approachable and meaningful ways for a range of users (Radford, Shannon & Roberts 2001). Improved interaction between program planning, delivery and evaluation would facilitate the development of integrated programs and allow for better learning outcomes.

Support was provided from the Office of the Deputy Vice-Chancellor (Education) to develop ProgramMap-a tool that could be loaded with one or more curriculum frameworks, descriptions of academic programs within one, some or all of those frameworks and then provide attractive and comprehensible representations of that information (Radford, Shannon *et al*. 2001).

Methodology

ProgramMap captures information from curriculum leaders about the intent of their programs and courses. As programs are developed and delivered in response to a range of existing curriculum frameworks, curriculum leaders were surveyed for information describing the extent to which their program components emphasised or focussed on the various dimensions of those frameworks.

Our three undergraduate academic programs were described: Bachelor of Design Studies (generic, entry-level program); Bachelor of Architecture; and Bachelor of Landscape Architecture.

We needed to decide at what level of detail we wished to describe these programs. The academic course was ultimately selected as an appropriate curriculum object on which to base our program descriptions. The course was considered appropriate for a number of reasons including:

1. Its clear definition across the institution (title, scheduling details, enrolment restrictions, points value, etc) and the availability in enterprise data systems of these course descriptors;

2. Its role in the enrolment of students;

3. The manageable number of courses that constitute most academic programs.

ProgramMap has been designed to preserve the option of integration of ProgramMap with our enterprise-wide online learning system and thereby improve the level of awareness of learning objectives in that system.

Data collection
Data was collected in structured interviews using a specially designed survey instrument. The survey methodology was validated using a small number of participants. Refinements indicated during validation were implemented before general use. Fifteen staff provided course information. Interviews typically took between 30-40 minutes for the first course and 15-20 minutes for subsequent courses.

The structured interviews ensured consistency and a high level of response. For some courses where responsibility was shared, data was collected from several respondents. Under these circumstances, ProgramMap calculates an arithmetic mean of active survey responses related to the same course instance. By associating responses to a particular course instance (offering) and to a particular respondent interview, ProgramMap allows for future changes in the program to be recorded and tracked.

Completed survey forms were converted to spreadsheets and the resulting data was loaded into ProgramMap using delimited text files and a data loading script. The original forms were retained and photocopies returned to the respondents. The data entry effort was minimal and data loading completed quickly and efficiently.

Due to the novel nature of the process and reporting style, respondents were offered an opportunity to revise their initial data. Only a relatively few, minor changes were sought and made to the original data supplied.

Survey respondents were encouraged to supply examples of student work that would help to illustrate their courses (Figure 1). The School sought a non-exclusive licence to use digital representations of their work from nominated students. The School undertook to observe the students' moral rights as creators of the material. Students were generally happy to have their work promoted and only licensed work is published within ProgramMap.

Curriculum frameworks

For the purposes of ProgramMap, a curriculum framework is a self-contained systematic curriculum description comprising a finite number of orthogonal (independent) dimensions and associated nomenclature. ProgramMap accommodates multi-level and multi-dimensional descriptions of curriculum frameworks. These can include, but are not limited to: learning outcomes (target graduate attributes, learning objectives); lists of competencies to be achieved, learning process descriptions (learning locale, learning activity, learning mode, outputs, etc).

Framework dimensions can be numeric (proportion of effort, time, assessment fraction) or textual (enumerated course objectives). Numeric dimensions are aggregated to year and program level using unit-weighted calculations. Textual dimensions are usually only reported at course level but may be aggregated by list concatenation to program level.

ProgramMap reports strictly within framework dimensions with no attempt to indicate or imply comparative weightings or scales between dimensions. ProgramMap has the potential for two-tiered dimension descriptions to allow for frameworks that provide or seek additional detail. When representing program analyses, ProgramMap collapses detail sub-dimensions into broader categories to improve readability. Detail information remains available in course level views.

Framework sources

Data from the following curriculum frameworks were loaded into ProgramMap:
- The University of Adelaide academic programs-University of Adelaide Calendar (UofA 2003)
- School of Architecture, Landscape Architecture and Urban Design Graduate attributes (SALAUD 2001)
- Bloom's Taxonomy (Cognitive Domain)-as described the Education Coalition (Lane 2001)
- Royal Australian Institute of Architecture Education Policy 2000 (RAIA 2000)
- Royal Australian Institute of Architecture National Visiting Panel Categories 2003 (RAIA 1997)
- Australian Institute of Landscape Architecture Education Policy 2002 (AILA 2002)
- *ad hoc* set of program descriptors including: Course Goals, Learning activities, Learning locale, Learning mode (group | individual), Student outputs (Roberts, Shannon & Radford 2003)

In addition to applying internal and generic curriculum frameworks, respondents describing courses within our generalist introductory program (Bachelor of Design Studies) were obliged to provide data within both the RAIA and AILA Education Policy frameworks. Courses in the Bachelor of Architecture

and Bachelor of Landscape Architecture programs were only described against their corresponding frameworks.

Framework descriptions and cues terms as defined by the framework creator were supplied to respondents to assist in the completion of the survey.

Technology

With the exception of student work examples, all ProgamMap data is stored within a relational database. Common Gateway Interface (CGI) scripts extract and render information in HTML as required. Student work examples can constitute single or collections of digital resources and can be stored within the ProgramMap system or held on external web sites. Collections of digital resources can be rendered into HTML dynamically if no index markup is supplied.

ProgramMap stores all representational information within the database allowing alternative "skins" to be applied within the application as appropriate to the provenance of the programs. Basic rights management is also incorporated to record approval (from students) for use of their work as exemplars.

No software licence fees were incurred in the development and deployment of ProgramMap as it is based exclusively on public domain software running on a Linux operating system. Standard relational database functionality is provided by PostgreSQL v6.5.3. The CGI is harnessed for data extraction and representation using the Perl5 extension for PostgreSQL (Pg). Perl scripts render all data representations in HTML v4 to ensure wide compatibility.
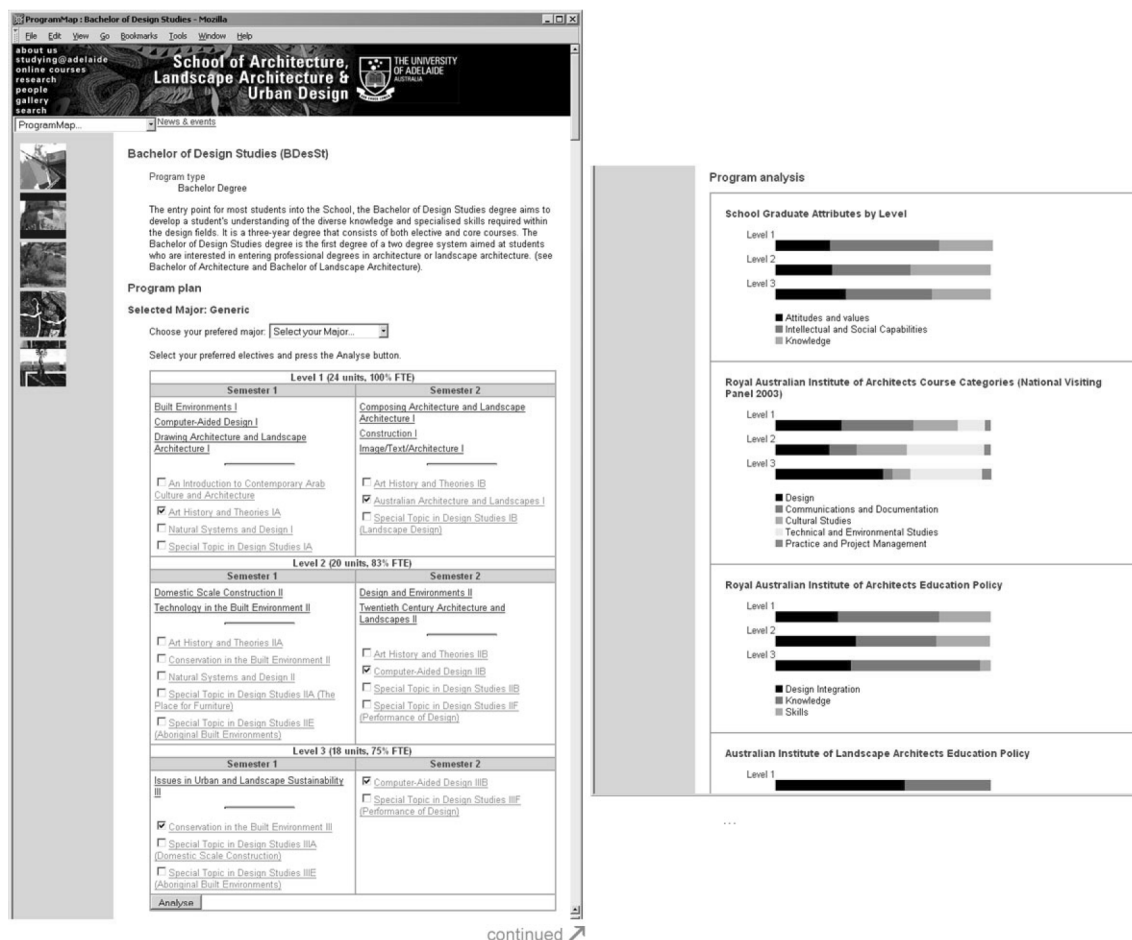
Figure 2: Bachelor of Design Studies program including subset of program analysis

Results

ProgramMap was designed, developed and deployed during the first phase of the project. ProgramMap marries course and program descriptors from enterprise data systems with information from curriculum leaders about the intent of their courses. Numerical, textual and student exemplar data were collected for all courses within the academic programs selected for the project (3 undergraduate academic programs, 41 courses, 68 exemplars total across 9 courses) (Radford, Shannon, Roberts & Jones 2003).

ProgramMap allows curriculum leaders to efficiently report aspects of their courses within multiple curriculum frameworks each of which comprises a number of dimensions along which the programs are described and outcomes expected and measured.

Mapping academic programs
ProgramMap contains information about the program plan (level and semester availability); program majors (where applicable); enrolment restrictions; study majors and core and elective courses. By default, ProgramMap presents a program-level analysis based on the core courses within the most generic major option available (Figure 2).

If necessary, users are presented with options to choose from available majors. ProgramMap responds by indicating changes in core and elective courses as appropriate. The user is then able to choose elective courses to complete a program for ProgramMap to analyse. A complete representation of a program analysis is only provided once a user has indicated a major and chosen from the available electives.

As a part of its analysis, ProgramMap indicates at each year level the proportion of a full-time load that is being included. ProgramMap's program analysis aggregates information about core and selected electives. The contribution of a course to the program analysis is proportional to the unit value of that course.

While the circumstances of the pilot ensured full description of all courses within the programs under consideration, ProgramMap has been designed to accommodate incompleteness (courses with enterprise data system information but no framework information available) by normalising the available data.

Mapping courses
Courses listed within the program analysis appear as links to separate, detailed descriptions of each course. The same course information as used in program analysis is included in the course descriptions but more detail is provided (Figure 3). Examples of student work may be included (Figure 1). Course descriptions, learning objectives, enrolment restrictions and scheduling information are combined with representations of the relationship of the course to relevant curriculum frameworks.

Examples of student work associated with particular courses were selected by the survey respondents and permission sought from the students to publish their work. Appropriate digital representations of the work were created from the work or obtained directly from the students. Student work can be represented by single digital resources or collections of files if necessary. There is no restriction on the file formats that can be included but the ability of end-users to download and view the material must be considered.

ProgramMap will automatically render collections of digital files as simple HTML files within a consistent graphical interface if no index file is available. ProgramMap can also accommodate links to external sites containing example work.

Visual data representations
Scripts were developed to represent proportional composition using coloured, horizontal, stacked bar charts rendered in HTML. These charts are functionally equivalent to circular pie charts but use much less space on the screen or page and can be rendered using capabilities native to all standard web browsers.

The use of colour allows for a great deal of information to be represented in a small area and at a glance. However, the project team was keen to ensure that access to the information was not denied to people with visual impairment so data labels and values, and alternative text attributes were provided to deliver equivalent information in textual form and thereby ensure accessibility (W3C 1999). An additional benefit for all users is that precise data can be read directly from the graphs.

Discussion

ProgramMap has been developed efficiently to collect information at an appropriate and convenient level of detail and to aggregate and represent that information for our range of academic programs.

The 2003 pilot of ProgramMap within the School of Architecture, Landscape Architecture and Urban Design has been positively received by students (prospective and enrolled), staff and external accreditation authorities. While detailed evaluation will be reported elsewhere (Shannon, Roberts & Radford 2003, paper submitted), users have reported that ProgramMap provided relevant information and presented it in an attractive and accessible manner. Prospective students indicated a strong interest in information describing learning activities, group/individual work, the types of work that would need to be produced, etc. The utility of ProgramMap during a recent RAIA course accreditation was clear.

University senior learning and teaching management have also recognised the value of this project and provided additional funds to further develop the application and to trial its broader adoption within the University.
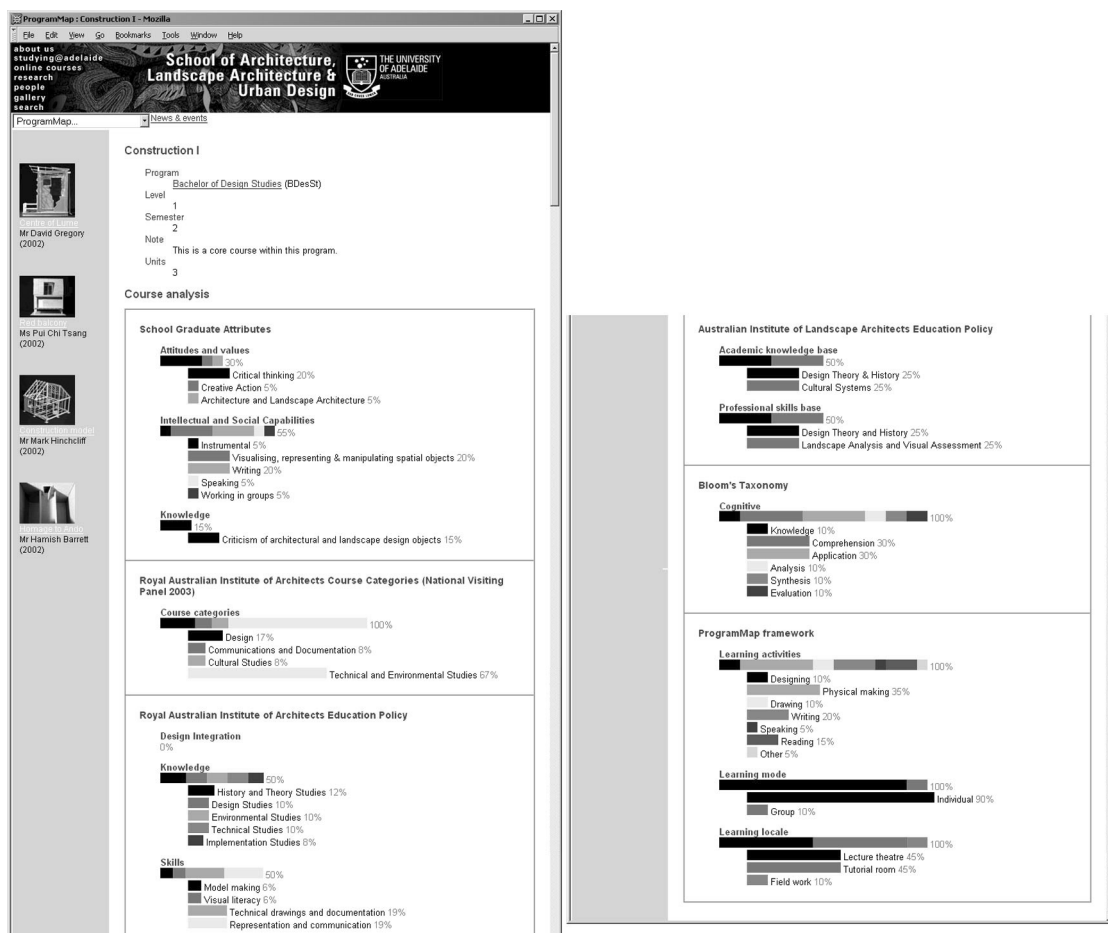
Figure 3: Construction I course (within BDesSt program)

Integrated curricula

The School remains committed to presenting an integrated curriculum because of the context and support it provides for effective and deep learning (Prosser & Trigwell 1999; Radford, Shannon *et al.* 2001). While this integration helps us support effective student learning, it leaves us with the challenge of ensuring that the diverse aspects of the curriculum (communication, design, construction, professional practice, etc) are appropriately covered to the satisfaction of ourselves, university authorities and external accreditation bodies.

ProgramMap has proved an effective, efficient and engaging way to record and report important information about our academic programs and assists us to reflect upon those programs.

In addition to meeting the needs of curriculum leaders, School management and program evaluators, ProgramMap was developed to be a tool that current and prospective students could use to understand the relationship between various program components and the roles that they play within the overall program. ProgramMap has become a curriculum knowledgebase accommodating and representing highly-dimensioned data, program descriptions and examples of student work.

Further development

Data entry: While the structured interview allowed us to accumulate a complete and highly consistent data set, current developments of ProgramMap will allow curriculum leaders to interact directly with the database and allow online data entry and updates. It is likely that curriculum frameworks will be loaded using XML or delimited text files.

Other discipline areas: The project team is keen to demonstrate the value of ProgramMap in other discipline areas. We are fortunate to work in an area that is visually rich providing ample scope for student work examples. We also benefit because our programs are reasonably completely described. Students complete a common set of core courses choose from a finite set of electives offered within the School. Schools delivering more generic programs such as those leading to Bachelors of Arts or Science need to deal with a much larger set of possible courses permutations with few if any core components. However, our obligation to two external accreditation bodies (in addition to internal requirements) generates substantial complexity in the design and evaluation of our programs that will not be experienced elsewhere. We have been directed and provided with resources to populate ProgramMap with information for programs from another school within the University. That work is in progress.

Display refinement: Feedback from users indicates that ProgramMap presents too much information at once. Work is underway to present users with the ability filter the information as required.

Alternate voices: While this project was explicitly focussed on collecting information from curriculum leaders on the intent of their courses it has been suggested that the tool could be used to capture student experience of the programs as well-information that is currently collected via Student Evaluation of Learning and Teaching surveys. The team is considering how this information might be collected and reported within ProgramMap.

Challenging frameworks: Survey participants reported various degrees of difficulty indicating the relationship of their courses within some framework. The project team considers that these areas of difficulty represent the best prospects for future investigation as conceivable sources of the problem include:

- Inappropriately designed or poorly articulated curriculum frameworks-indicating a need for revision of the frameworks;

- Inadequate implementation of the curriculum frameworks within ProgramMap-indicating a need for further development or redesign or the tool;

- Poor alignment between academic program components and curriculum frameworks-indicating a need for revision of those components of the program.

Handing incompleteness: Designed to accommodate incompleteness in course descriptions, ProgramMap currently does not indicate the proportion of the program that has **not** been described. Feedback has indicated that this information ought to be provided. Alternative ways of displaying this information are under consideration.

Student portfolios: Integration of ProgramMap information with a student's enrolment details and examples of their work is seen as a creative and feasible approach to provide students with an informative student portfolio. The role that the University's learner management system might play in this is under investigation.
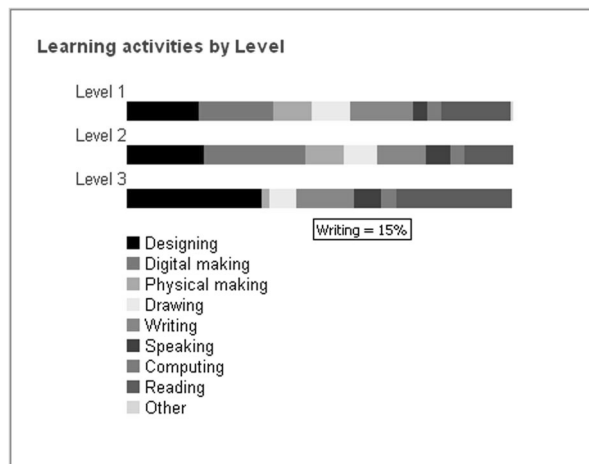

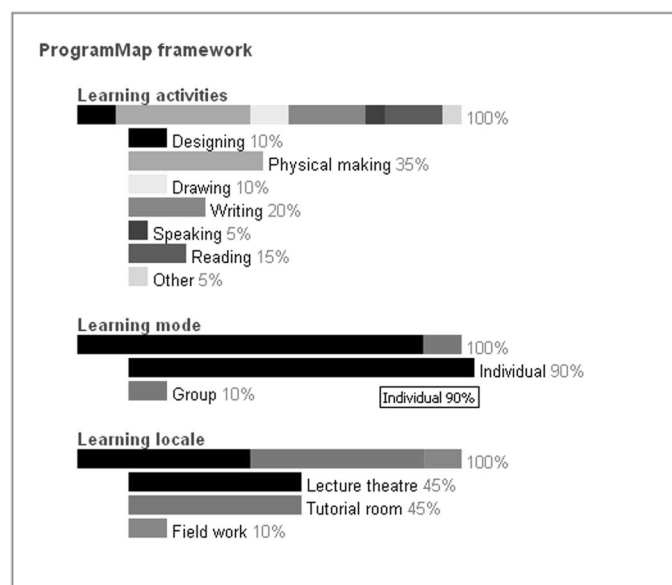*Figure 4: Program level proportional bar charts*

Figure 5: Course level proportional bar charts

References

AILA (2002). Landscape Architecture Education Policy 2002, Australian Institute of Landscape Architects.
    http://www.aila.org.au/ailaweb/education/2002%20policy%20doc/AILA%20educ%20Policy.pdf.
Boumelha, P. (2001). Graduate Attributes, Adelaide University.
    http://www.adelaide.edu.au/DVC/students/graduate_attributes.html.

Lane, C. (2001). Blooms Taxonomy, The Education Coalition.
    http://www.tecweb.org/eddevel/edtech/blooms.html.

Nelson, B. (2003). Our Universities -Backing Australia's Future. Canberra, Australia, Commonwealth of
    Australia: 48 http://www.backingaustraliasfuture.gov.au/policy_paper/policy_paper.pdf.

Prosser, M. & K. Trigwell (1999). Understanding Learning and Teaching: The Experience in Higher
    Education, Open University Press, 325 Chestnut Street, Philadelphia, PA 19106.

Radford, A., S. Shannon & I. Roberts (2001). ProgramMap: Program and Course mapping to scaffold
    learning, The University of Adelaide.
    http://www.adelaide.edu.au/DVCE/quality/grants/doc/architecture_matchingIT.doc.

Radford, T., S. Shannon, I. Roberts & D. Jones (2003). Program and Course Mapping to Scaffold
    Learning: Final Report. Adelaide, School of Architecture, Landscape Architecture and Urban Design,
    The University of Adelaide: 25
    http://www.arch.adelaide.edu.au/~iroberts/papers/programMap.finalrpt.2003.pdf.

RAIA (1997). Australian Architecture Course Recognition & Accreditation Procedure, The Royal
    Australian Institute of Architects.
    http://www.architecture.com.au/i-cms_file?page=/1/23/198/Websiteaustarchcourserecogaccredprocbo
    oklet2-200201.pdf.

RAIA (2000). The Royal Australian Institute of Architects Education Policy, Royal Australian Institute of
    Architects.
    http://www.architecture.com.au/i-cms_file?page=/1/23/542/543/endorsed_policy_2000.PDF.

Roberts, I., S. Shannon & A. Radford (2003). ProgramMap, The University of Adelaide.
    http://www.arch.adelaide.edu.au.

SALAUD (2001). Graduate Attributes, School of Architecture, Landscape Architecture and Urban
    Design, the University of Adelaide.
    http://www.arch.adelaide.edu.au/study/gradattributes.shtml.

Shannon, S., I. Roberts & A. Radford (2003). ProgramMap: a tool for web-based mapping of graduate
    attributes. ANZAScA 2003, University of Sydney, Australian & New Zealand Architectural Science
    Association.

UofA (2001). Quality Assurance at Adelaide University: Frameworks, Issues, Tasks, 2001. Adelaide,
    Academic Board, The University of Adelaide: 9
    http://www.adelaide.edu.au/DVCE/quality/doc/fwork_acboard.pdf.

UofA (2003). 2003 University Calendar, The University of Adelaide.
    http://www.adelaide.edu.au/calendar/.

W3C (1999). Web Content Accessibility Guidelines 1.0 -W3C Recommendation 5-May-1999, W3C
    Worldwide Web Consortium. 2000 http://www.w3.org/TR/WAI-WEBCONTENT/.