Concise paper

ICE: Integrated Computing Education

- An individual integrated computing teaching experience

Min Jie Hu
Tairawhiti Polytechnic
Gisborne, NZ
min@tairawhiti.ac.nz

ABSTRACT

Each NACCQ programme consists of a set of modules in order to build up enough credits for a certificate or diploma. However, it is hard for students to find out the relationship between these modules they study. Thus, they are likely to forget the modules after they have completed. Sometimes, they seem like novices when asked question that you expect them to already understand from their previous programme of study. The author has opportunities to teach a number of computing modules including different computer languages, database, application software and computer hardware at different levels. It is a challenge for one person to continuously teach a number of modules to the same class with smooth transitions and linkages from one module to another. This paper attempts to explore the relationship among all the modules the author delivered at DipICT level 6 programme and to find a way to integrate computing teaching of this programme. The author is also pleased to share this experience and findings with other educators and expects to see more educators enjoying the Integrated Computing Education (ICE).

Keywords:
integrated computing education (ICE), computer language, database and application software

1. INTRODUCTION

Each NACCQ (2002) programme consists of a set of modules in order to build up enough credits for a certificate or diploma. However, it is hard for students to find out the relationship between these modules. Thus, they are likely to forget the modules after they have completed. Sometimes, they seem like novices when asked a certain question that you expect them to already understand from their previous programme study.

For the DipICT Level 6 programme, the author teaches DA600, SP610, DB600 and FG600 sequentially. Parallel to this, the author also teaches PR 610. It is a challenge for one person to continuously teach a number of modules to the same class with smooth transitions and linkages from one module to another.

The following research starts from analysing the benefits of a currently integrated teaching approach. It explores the linkages and relationships between the modules delivered at Tairawhiti Polytechnic and finally, to find a way of integrating computer teaching.

2. INTEGRATED TEACHING

Integrated teaching is not a new method in education. When engineering education started its reform, it emphasised integrated interdisciplinary applications, hands-on learning, development of communications and teamwork skills. While the integrated teaching is successful for most engineering majors, it applies equally to computing discipline.

2.1 Advantages and Disadvantages of Integrated Teaching

In 1988, Drexel University (Quinn, 1995) initiated its first-year and second-year engineering curricular reform as "E4" (Enhanced Educational Experience for Engineers). They offered the same

integrated courses for all students regardless of their major. All the presentations, assignments, tests and exams are integrated. In 1992, nine other universities took part in the Gateway Engineering Education Coalition under Drexel's leadership. Seven other engineering education coalitions were also established.

Parker *et al.* (1995) attempted their improvement of engineering education in terms of curriculum integration, active learning, and technology enabled education. Froyd and Rogers (1997) also started their Integrated, First-Year Curriculum in Science, Engineering, and Mathematics in 1988. It bridges gaps between disciplines and also develops a positive and flexible learning environment.

Recently, Fortier *et al.* (2002) reported that traditional classroom and teaching pedagogy is slowly being replaced with more active, team-work, cooperative learning, and hands on approaches. They developed their teaching programme called IMPULSE (Integrated Math, Physics and Undergraduate Laboratory Science and Engineering).

Why is integrated engineering teaching so popular? Parker *et al.* (1995) expected the integrated course to motivate student's parallel interest in different subjects. Richardson *et al.* (1996) and Al-Holou *et al.* (1998) found that integrated course promotes students' broadly understanding when they learn related topics simultaneously. They believed that it could help students to visualise and understand links between different disciplines. It is also better for retention of material.

However, Al-Holou *et al.* (1998) also found that faculty often expressed reluctance to work with other faculty, especially across departmental boundaries. Some faculties even believe it is the students' responsibility to make connections between the topics they are studying.

2.2 Integrated Teaching in Computing

When engineering process of "analyse, design, build and test" was introduced to computing discipline, it led to the generation of Software Engineering. Now, the integrated teaching, after it was successful for most engineering majors, is extended to computing education.

Cordes *et al.* (1997) reformed their first-year computer courses by teaching Software, Hardware, and Discrete Mathematics concurrently so that related topics are introduced together. The coordination of topics illustrates how relationships are exploited to give students a better understanding. They apply a "just-in-time" approach to deliver concepts between different subjects in the same day or next day.

Furthermore, Cordes *et al.* (1998) associated Discrete Mathematics with Digital Logic, Hardware and Programming. They tried to provide students a "big picture" of the entire discipline to be able to comprehend easily. Finally, they found that students' motivation was increased and the quality of programming assignments was significantly higher. Similarly, Belleville (2004) integrates Discrete Mathematics with Hardware for a better understanding the interplay between them.

Following the way of integrated engineering teaching, Yu (1997) compacted a number of foundation courses into one integrated course, which contains material from C, System Development, Hardware and Data Communication. The experience confirms that an integrated approach to teaching and understanding computer topics is definitely viable and advantageous. Griffin *et al.* (2004) also designed a new paper involving all engineering and computer departments. The integrated paper includes Problem Solving, Ethics, Graphics and OOP. A positive reaction was given from both students and faculty.

In achieving the goal of cross-functional integration of material, educators also developed integrated computing teaching to other related disciplines. Pendegraft *et al.* (2000) have developed an integrated information systems and business course. In particular, integrated projects encourage student appreciation of business problem solving with information systems. Deek (2002) disseminated a "computing and composition" approach to integrate first-year college programming and English composition courses. CACHE Corporation (2004) is also looking for integrated teaching computer programming, software selection, mathematical modelling, process and product design throughout the chemical engineering curriculum.

3. FINDINGS

The following finding is based on the integrated teaching experience from DipICT level 6 programme, with software Office 2000, SQL Server 2000 and VB.NET 2003.

3.1 Linkage between Data Analysis and MS Office Applications (DA600 and SP610)

Students have learned MS Access in their introductory study. But a lot of them are not quite sure of what kinds of tables and fields are necessary in the database. A sample report is introduced to let students find out how many tables are needed when they start module DA600. While E-R Modelling is helping students to establish the main idea of tables, normalisation supports them in sorting out which fields must be included in each table. Thus, a consolidated data model is created.

In module SP610, an integrated simulated project (practical assignment) implements the data model. It needs students to convert data from one program to another including MS Access, Excel, Internet Explorer, Outlook, PowerPoint and Word. Finally, students are required to create Online Help in CHTML (complied HTML).

The integrated project not only links the database theory to practice but also covers most basic applications students have studied. It motivates students to apply different programs together for their MS Office level application.

3.2 Linkage between different Database Management Systems (SP 610 and DB600)

MS Access is basically good at PC application. For multi-users network or Internet applications, we have to use other DBMS. In the integrated application, SQL Server is used as a main database to hold tables, views and stored procedures on the server computer, while Access is applied for user's interface to keeps Forms and Reports on the client computers. Students use Access Form to enter data to SQL Sever and then retrieve data back to Access Report. It also fills the gap when, so far, there is no Form and Report available for our SQL Server system, although Microsoft has just released the Report system.

The integrated application is smoothly moving from Access to SQL Server, from PC to Client-Server network environment. Students use their already known concepts to understand new knowledge (such as Query to View, QBE to SQL, stand-alone to multi-user environment). It provides retention of students' knowledge and stimulates their motivation for learning.

3.3 Linkage between Programming and the above modules (PR610 with DA600, SP610 and DB600)

Students study programming at the same time when they learn the above integrated applications. After understanding data modelling, students use their programming skills to retrieve data from tables, views and stored procedures. They also create reports from programming language rather than from DBMS. Similarly, students also apply data modelling to understand their programming on related files.

Programming on GUI design motivates students' interest in developing applications. GUI also links the On-line Help created in SP610. Since VS.NET makes it easy to transfer project from windows to web, students are even more interested in integrated web development to web programming and web database application.

In a word, students learned to use computer language for front-end GUI design and use database behind the application. Integrated teaching programming, database and web application stimulates students to develop further applications on both windows and web.

3.4 Relationship between Programming and 4GL(PR610 and FG600)

Students study SQL through both Access and SQL Server. After creating stored procedures with parameters in a server database by both DBMS and VB.NET, students send argument data from their programming applications to retrieve data from database. They also run applications to transfer Access database to SQL Sever.

Integrated teaching programming and 4GL clarifies the difference and presents the linkage

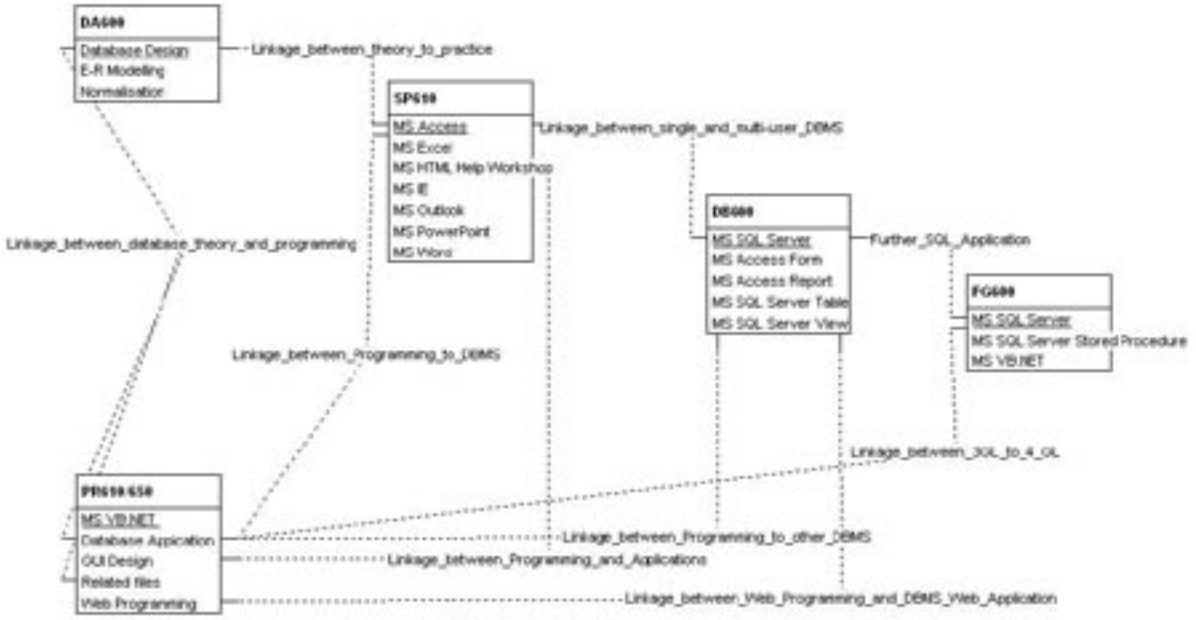

Figure 1. Linkage among modules within integrated teaching

between 3GL and 4GL. It continues further linkage from previous modules of programming and database.

4. CONCLUSION

Database application is always popular as a highlighted thread through the software teaching. It is helpful to find out the linkage from database theory to practice, database application to other program application, programming to database. Figure 1 shows the relationship of the above findings.

As you can see from the figure, students start to learn database design (DA600) and then practice (SP610) at Office application level. After integrated application, they move to a multi-user network database (DB600) environment. And then, they carry on to study more about SQL as 4GL (FG600). At the same time, programming (PR610) links all the other modules at a deep level with language coding. It basically uses another way to implement database design (DA600). It also works together with different database applications (SP610 and DB600). Moreover, programming for related files links to data modelling (DA600) and programming for GUI links to integrated applications (SP610). Finally, programming interacts to 4GL (FG600).

Integrated teaching links different modules together and let students to repeat using different packages together. It helps students to understand all modules they learned and to master packages they used. It is also easy for students to transfer data from one programme to another when consecutively using a series of compatible software in different modules, e.g. Office, SQL Server, and Visual Studio.NET. Integrated teaching provides students with the whole consequential ideas of what they studied rather than individual separate points they have learned.

Integrated teaching so far only relates to limited topics and modules. Fortunately, NACCQ programme has divided all the subjects into modules, which basically includes several small topics. The teaching style of delivering each module at most ITPs is a mini lecture together with hands-on lab for a small class. Team work and group tasks have already been widely accepted and applied in the ITP environment. This makes it easy for us to take the advantage of the success of integrated teaching and to reorganise our teaching modules. Especially, it is easier and more flexible for one lecturer/tutor to integrate all her/his own teaching modules. She/he only needs to review all her/his own teaching modules, find the linkage between them, and reorganise them for integrated teaching. By individual integrated teaching, there is no worry about issues across departments or faculty such as topics, time schedule, computer room usage etc. After individual

teaching modules have been integrated, it is then time to move to the integration between staff and department. Meanwhile, more integrated module development by NACCQ team from different ITPs would be very helpful for both individual and group integrated teaching development.

The author expects to see more integrated teaching of different computing topics by both individuals and groups of educators. Further research would discover more topic coordination. This certainly involved more staff and needed their cooperation to discover the internal linkage of different topics for the development of integrated computing education. ICE will be then widely accepted, popular and enjoyed by most educators.

REFERENCES

Al-Holou *et al.* (1998) "First-year integrated curricula across engineering education coalitions", 1998 Frontiers in education conference

Belleville, P. (2004) "Integrating introductory Discrete Mathematics and logic design", University of British Columbia, Canada. www.cs.ubc.ca/wccce/program04/Papers/patrice/cpsc121-full.html

CACHE Corporation (2004) "The White paper, Computing through the curriculum: An integrated approach for chemical engineering", www.che.utexas.edu/cache/Edgar-computingthrough.html

Cordes, D. *et al.* (1997) "An integrated first-year curriculum for computer science and computer engineering", 1997 Frontiers in education conference

Cordes, D. *et al.* (1998) "Teaching an integrated first-year computing curriculum: Lessons learned", 1998 American Society for Engineering Education Annual Conference

Deek, F., Friedman, R. and Kimmel, H. (2002) "The computing and composition as an integrated subject in the secondary school curriculum", Proceedings of the 2002 American society for engineering education annual conference & exposition

Fortier, P. *et al.* (2002) "Technology supported learning applied to an innovative, integrated curriculum for first-year engineering majors", 35th Annual Hawaii International Conference on System Sciences, IEEE

Froyd, J. and Rogers, G. (1997) "Evolution and evaluation of an integrated, first-year curriculum", 1997 Frontiers in education conference

Griffin Jr., H. *et al.* (2004) "Work in progress: A freshman course for engineering and computer science students", 34th ASEE/IEEE Frontiers in education conference

NACCQ (2002) "Blue Book, New Zealand polytechnic qualification in information and communication technology", National advisory committee on computing qualification, 9th edition, NZ

Parker, J. *et al.* (1995) "Engineering design in the freshman year at the university of Alabama - Foundation coalition program", 1995 Frontiers in education conference

Pendegraft, N., Stone, R. and Dyers, R. (2000) "Using information systems as a unifying influence in an integrated business curriculum", Journal of information systems education, pp. 61-66, Issue 1-2, Volume 11

Quinn, R. (1995) "Implementing large scale curricular changes - The Drexel experience", 1995 Frontiers in education conference

Richardson *et al.* (1996) "The foundation coalition freshman year: Lessons Learned", 1996 Frontiers in education conference

Yu, B. (1997) "An integrated approach in teaching computing courses", British Columbia Institute of Technology, Canada. www.cs.ubc.ca/wccce/program97/yu/yu.html