A Constructivist-Based Approach to Teaching Database Analysis and Design

Thomas M Connolly
Carolyn E Begg
School of Computing
University of Paisley
Paisley, PA1 2BE, Scotland
thomas.connolly@paisley.ac.uk carolyn.begg@paisley.ac.uk

ABSTRACT

The study of database systems is typically core in undergraduate and postgraduate courses related to computer science and information systems. However, there are parts of this curriculum that learners find difficult, in particular, the abstract and complex domain of database analysis and design, an area that is critical to the development of modern information systems that meet the demands of users in an efficient and effective way. In addition, there is some evidence that companies believe the database analysis and design skills of both new graduate recruits and some of their existing IT staff are insufficient to cope with the complexities encountered in developing such systems. This paper reflects on these difficulties and describes a teaching approach motivated by principles found in the constructivist epistemology to help overcome these difficulties and help provide the learner with the knowledge and higher-order skills necessary to understand and perform database analysis and design effectively as a professional practitioner. The paper presents some preliminary results of this work that seems to suggest that students can learn how to design effective modern information systems when the learning is embedded in problem-solving contexts that are relevant in the real-world.

Keywords: Database analysis and design, constructivism, reflective practitioner, constructivist learning environments, project-based learning

1. INTRODUCTION

The database is now the underlying framework of the information system and has fundamentally changed the way many companies and individuals work. This is reflected within tertiary education where databases form a core area of study in undergraduate and postgraduate courses related to computer science and information systems, and typically at least an elective on other data-intensive courses (ACM/IEEE, 2001; EUCIP, 2003). The core studies, typically, are based on the relational data model, SQL, data modeling, relational database analysis and design and, increasingly, object-relational concepts. This supports industry where the object-relational DBMS is the dominant data-processing software currently in use. With more than 30 years since Codd proposed the relational data model in his seminal paper (1970), the core relational theory is a mature and established area now in relation to other parts of the computing curriculum. However, there are parts of this curriculum that learners find difficult, in particular, database analysis and design. In addition, a recent European survey found that the skill companies considered to be most lacking in both new IT graduate recruits and in some of their existing IT staff was database design (database tuning and database administration were second and third, respectively) and that this was affecting their competitive ability (Connolly, 2005).

Mohtashami and Scher (2000) note that pedagogical strategies for teaching database analysis and design traditionally follow a similar modality to that of other technical courses in computing science or information systems. A significant amount of technical knowledge must be imparted by the teacher and the learners becoming mainly passive listeners. This is the objectivist model of learning, which views learning as the passive transmission of knowledge from the teacher to the learner, heavily criticized for stimulating surface learning and knowledge reproduction. In contrast, the central tenet of the constructivist view is that learning is an active process where new knowledge is constructed based on the learner's prior knowledge, the social context, and the problem to be solved. In this paper, we describe a teaching approach that we have used motivated by principles found in the constructivist epistemology to help provide the learner with the knowledge and higher-order skills necessary to understand and perform database analysis and design effectively as a professional practitioner. In the following section, we outline the high-level pedagogical aims of our database modules and consider some of the difficulties that arise in achieving these aims. In the subsequent section, we examine related work on constructivism and constructivist learning environments. In Section 4, we put forward our own guidelines for an appropriate constructivist environment and discuss how we have applied these guidelines to our teaching of two database modules. In Section 5, we present some early findings from our approach followed by some conclusions and directions for future work.

2. PEDAGOGICAL AIMS AND ANALYSIS

Our database modules, each taught over a 15-week semester and based on a nominal 150 student effort hours, have the following educational aims:

- Develop a sound understanding of the principles and underpinning theory related to the study of database systems.
- Assist the development of an engineering/business ethos in the student that emphasizes fitness for purpose as the guiding principle in the design, development, and assessment of database systems and their components.
- Enable the student to take a disciplined approach to problem definition, and to the specification, design, implementation, and maintenance of database systems.
- Develop critical, analytical, and problem-solving skills and the transferable skills to prepare the student for employment.
- Assist the student to develop the skills required for both autonomous practice and team-working.
- Enable the student to engage in lifelong learning, study, and enquiry, and to appreciate the value of education to society.
- Create awareness of the continuing development of database technologies and applications and the need for continued study, reflection, and development throughout a career as a database professional.
- Develop an appreciation of the legal, professional, and ethical principles pertinent to the practicing database professional to enable the student to identify appropriate practices and to support their continuing professional development.

Our modules have a vocational orientation and we expect our graduates to become professional database practitioners typically in a multi-disciplinary environment. These aims are typical of database modules at many universities yet, students do have difficulty achieving these aims and, as noted earlier, research indicates that graduates may not be providing companies with the level of database analysis and design skills they require. In this section, we examine the difficulties that arise in teaching database analysis and design.

Previous approaches to educating database designers and, more generally, software designers model scientific and engineering methodologies, whose focus is on process and repeatability. In general, this approach is based on a normative professional education curriculum, in which students first study basic science, then the relevant applied science, so that learning may be viewed as a progression to expertise through task analysis, strategy selection, try-out, and repetition (Armarego, 2002). While students tend to cope well using this approach with many of the theoretical and practical components of the core database curriculum, for example, understanding the properties of the relational data model, the basics of SQL, and using a relational DBMS, one area that tends to be problematic is the abstract and complex domain of database analysis and design. For the purposes of this paper, we use the term database analysis and design to encompass system definition, requirements collection and analysis, conceptual database design, logical database design, and physical database design. A comparable problem has been identified with object-oriented analysis and design, which is also highly abstract (Hadjerrouit, 1999; Yazici *et al.*, 2001), requirements engineering (Bubenko, 1995), and software design and testing (Budgen, 1995). Waks (2001, pp. 39) argues that "the crisis of professions arises because real-life problems do not present themselves neatly as cases to which scientific generalizations apply", which becomes an issue of espoused theory (the words we use to convey what we do or what we would like others to think we do) versus theory-in-use (theory that is implicit in what we do as practitioners) (Argyis and Schön, 1974).

While databases have become so essential to organizations, Kroenke (2004, pp. xi) states "unfortunately, increased popularity has not meant increased competency. Many students (as well as professionals) have been deceived by the simplicity of creating small databases using products such as Microsoft Access. With this background, they believe they know sufficient database technology to create databases that have more complicated structure and greater processing complexity. The result is often a mess: databases are hard to use, barely meet system requirements, and are difficult to redesign." While this is true, we would contest that there are other issues such as:

- the skills to work in a project team; the skills to apply appropriate fact-finding techniques to elicit requirements from the client (both "soft", people-oriented skills);
- the skills to conceptualize a design from a set of requirements ("soft", analytical and problem-solving skills);
- the skills to map a conceptual model to a logical/physical design ("hard", technical skills);
- the skills to reflect and review intermediate designs, particularly where information complexity is present (a combination of "soft" and "hard" skills).

These are different skills from learning SQL, knowing the components of an ER model, or being able to recite the properties of the relational model or the advantages/disadvantages of relational systems. Students often have considerable difficulty comprehending implementation-independent issues and analyzing problems where there is no single, simple, well-known, or correct solution. They have difficulty handling ambiguity and vagueness, which can arise during knowledge elicitation. They can also display an inability to translate classroom examples to other domains with analogous scenarios, betraying a lack of transferable analytical and problem-solving skills. These problems can lead to confusion, a lack of self-confidence, and a lack of motivation to continue. In teaching database methods we are, as Postman and Weingartner (1971, pp. 56) state, "trying to help students to become more efficient problem solvers", avoiding "the right answer that only serves to terminate further thought" and reach a position where the student "must learn to depend on himself as a thinker".

Software engineering (and therein database analysis and design) has been described as a "wicked problem", characterized by incomplete, contradictory and changing requirements, and solutions that are often difficult to recognize as such because of complex interdependencies (DeGrace and Hulet Stahl, 1998). According to Armarego (2002), there is an educational dilemma in teaching such problems in software engineering because:

- complexity is added rather than reduced with increased understanding of the problem;
- metacognitive strategies (ie., strategies that students can use to guide their own comprehension by analyzing how they are learning) are fundamental to the process;
- a rich background of knowledge and intuition are needed for effective problem-solving;
- a breadth of experience is necessary so that similarities and differences with past strategies are used to deal with new situations.

As pointed out by Eaglestone and Baptista Nunes (2004) there are a number of other factors that have impacted database teaching, such as the introduction of semesters (as opposed to the previous three term structure), increasing student numbers through the drive for mass higher education, additional administrative processes of assessment and student progress monitoring, compliance with institutional and national policies, and the fact that institutions devote a disproportionate weight to research as opposed to teaching. To compound these factors, it is not unusual for academics within Higher Education to have no formal training in teaching and learning. In addition, to provide more flexible modes of study and capture new markets, tertiary education is providing more modules and courses in an online format, resulting in students who are geographical dispersed and have diverse backgrounds (Connolly *et al.*, 2006).

Figure 1 is a representation of the types of knowledge and skills required to undertake a database analysis and design project and the associated problems. The types of knowledge necessary to support such a project are drawn from topics covered in Database Concepts, Database Analysis & Design, Database Implementation, and Business & Management. The skills necessary to support such a project are drawn from Intellectual, People-oriented, Business-oriented, Personal, and Domain-specific skills. Although shown separately, the figure shows a strong association between knowledge & understanding and skills. For example, the knowledge and understanding of ER modeling concepts is supported by developing the necessary skills to identify entities, relationships and attributes and to represent these concepts in an ER diagram and vice versa. Complicating factors associated with undertaking database analysis and design projects are also represented. For example, such projects are often ill-defined with changing requirements. Finally, the figure also represents the distinctive background knowledge that each student brings to the learning environment. The purpose of this figure is to represent the knowledge, the skills, and the complicating factors that make undertaking a database analysis and design project so difficult.

The above discussions suggest an alternative approach to teaching database analysis and design may overcome some of the above difficulties and in the next section we examine one such approach that we have found useful.

3. PREVIOUS WORK ON CONSTRUCTIVIST APPROACHES

3.1 Constructivist Theory

While traditional education has been guided by the paradigm of didactic instruction, which views the learner as passively receiving information, there is now an emphasis on *constructivism* as a philosophical, epistemological, and pedagogical approach to learning. Cognitive constructivism views learning as an active process in which learners construct new ideas or concepts based upon their current/past knowledge. The learner selects and transforms information, constructs hypotheses, and makes decisions, relying on a cognitive structure to do so (Piaget, 1968). In addition, constructivism asserts that people learn more effectively when they are engaged in constructing personally meaningful artifacts. Social constructivism, seen as a variant of cognitive constructivism, emphasizes that human intelligence originates in our culture. Individual cognitive gain occurs first in interaction with other people and in the next phase within the individual (Forman and McPhail, 1993). These two models are not mutually exclusive but merely focus upon different aspects of the learning process. In fact, Illeris (2003) believes that all learning includes three dimensions, namely, the cognitive dimension of knowledge and skills, the emotional dimension of feelings and motivation, and the social dimension of communication and cooperation – "all of which are embedded in a societally situated context".

According to Gance (2002) the main pedagogical components commonly associated with these models are:

- A cognitively engaged learner who actively seeks to explore his environment for new information.
- A pedagogy that often includes a hands-on, dialogic interaction with the learning environment. For example, actually designing a database is preferred to simply being told how to design a database.
- A pedagogy that often requires a learning context that creates a problem-solving situation that is realistic.
- An environment that typically includes a social component often interpreted as actual interaction with other learners and with mentors in the actual context of learning.

The ultimate goal of a constructivist approach is metacognition (the higher order process of reflecting on our own thinking and problem solving processes), which has powerful problem-solving potential. When the learner

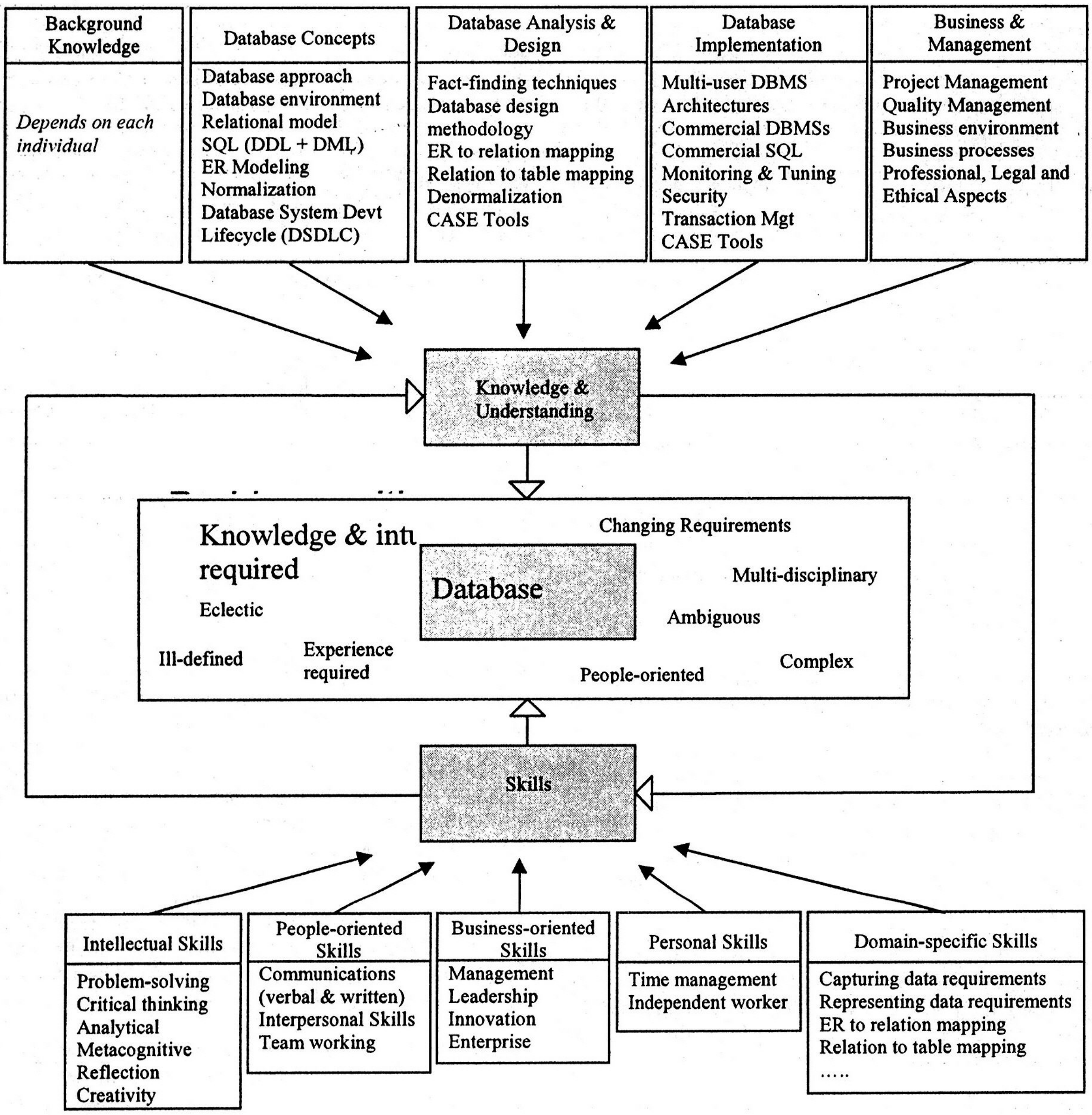

Figure 1 Types of knowledge and skills required to undertake database analysis and design

encounters a problem he can reflect not just on the structure of the problem, but on the structuring of his approaches to the problem and thereby attempt to generate alternative, more productive strategies. Not only is this a useful ability, but the ultimate expression of education - to reflect back on what has been created by the process of education (Boyle, 2000). Cunningham (1991) emphasizes that constructivism provides a clear theory-based approach for design.

Dewey (1958) argued that knowing and doing are intimately connected and that learning occurs in the context of activity when an individual attempts to accomplish some meaningful goal and has to overcome difficulties in the process. Schön (1983, 1987) describes professionals as individuals who make this connection between knowing and doing through reflective practice, suggesting that professionals learn to think in action and learn to do so through their professional experiences. He argues that the primary challenge for designers is how to make sense out of situations that are puzzling, troubling, and uncertain. According to Schön (*ibid*) the following are some of the key problems in teaching design:

- It is learnable but not didactically or discursively teachable: it can be learned only in and through practical operations.

- It is a holistic skill and parts cannot be learned in isolation but by experiencing it in action.
- It depends upon the ability to recognize desirable and undesirable qualities of the discovered world. However, this recognition is not something that can be described to learners, instead it must be learned by doing.
- It is a creative process in which a designer comes to see and do things in new ways. Therefore, no prior description of it can take the place of learning by doing.

For Schön (*ibid*), practitioners (in our case, database designers) have their own particular knowledge codes fully embedded within their practices. They apply tacit knowledge-in-action, and when their problems do not yield to it, they *reflect-in-action*, using the languages specific to their practices. When they evaluate the event afterwards, they *reflect-on-action*, using the language of practice, not the language of science. In this way, professionals enhance their learning and add to their repertoire of experiences, from which they can draw in future problem situations. He believes that it is this ability to reflect both in, and on, action that identifies the effective practitioner from less effective professionals. For Schön (*ibid*) the ideal site of education for reflective practice is the "design studio" where, under the direction of a master practitioner serving as coach, the novice learns the vocabularies of the professional practice in the course of learning its "operational moves". In making the moves, talking about them and even talking about their talk about them (meta-reflection), the novice and coach negotiate the "ladder of reflection".

Shaffer (2004a) proposes a theory of "pedagogical praxis", which links learning and doing within an extended framework of communities of practice (Lave, 1991; Lave and Wenger, 1991). Pedagogical praxis is based on the concept that different professions (for example, lawyers, doctors, database designers) have different epistemologies (epistemic frames) – different ways of knowing, of deciding what is worth knowing, and of adding to the collective body of knowledge and understanding. For a particular community, the epistemic frames define "knowing *where* to begin looking and asking questions, knowing *what* constitutes appropriate evidence to consider or information to assess, knowing *how* to go about gathering that evidence, and knowing *when* to draw a conclusion and/or move on to a different issue" (Shaffer, 2004b, pp. 4). Implementation of pedagogical praxis requires a faithful recreation of the professional community, one that is "thickly authentic"; that is, one where (a) learning is personally meaningful for the learner, (b) learning relates to the real-world outside the classroom, (c) learning provides an opportunity to think in the modes of a particular profession, and (d) learning where the means of assessment reflect the learning process (Shaffer and Resnick, 1999). We would suggest that the term thickly authentic be extended to incorporate: (e) learning using the tools and practices of the modern-day professional.

These arguments suggest that students can only learn about design by doing design, and rely less on overt lecturing and traditional teaching. This approach requires a shift in the roles of both students and teachers, with the student becoming an apprentice, exploring and learning about the problem in the presence of peers (who may know more or less about the topic at hand) and the teacher moving from being the "knowledgeable other" towards becoming a facilitator, who manages the context and setting, and assists students in developing an understanding of the material at hand (Koehler and Mishra, 2005).

3.2 Constructivist Learning Environments and Problem/Project-Based Learning

Many researchers have expressed their hope that constructivism will lead to better educational software and better learning (for example, Brown *et al.*, 1989; Jonassen, 1994). They emphasize the need for open-ended exploratory authentic learning environments in which learners can develop personally meaningful and transferable knowledge and understanding. This has led to the development of guidelines and criteria for the development of a *constructivist learning environment* (CLE) - "a place where learners may work together and support each other as they use a variety of tools and information resources in their guided pursuit of learning goals and problem-solving activities" (Wilson, 1996, pp. 28). See, for example, Cunningham *et al.*, 1993; Grabinger and Dunlap, 1995; Savery and Duffy, 1995; Gance, 2002). According to Ben-Ari (2001) constructivist principles have been more influential in science and mathematics education than in computer science education. However, there are examples of the application of constructivism within computer science from the development of Logo – a programming language for schoolchildren (Papert, 1980), the teaching of programming (for example, Pullen, 2001; Van Gorp and Grisson, 2001), computer graphics (Taxén, 2003), CASE tools (Fowler *et al.*, 2001), object-oriented design (for example, Hadjerrouit, 1999; Yazici *et al.*, 2001), communication skills in computer science (Gruba and Søndergaard, 2001), to collaborative learning using the Web (for example, Cook and Boyle, 2000; Hadjerrouit, 2003; Connolly *et al.*, 2005).

The *problem-based learning model* encompasses these principles. This model started out in the 1960s in medical education in the USA and Canada where groups of students were presented with a problem in the form of a patient with particular symptoms (Biggs, 1999). The students' task is to diagnose the patient's condition and be able to justify the diagnosis and recommend treatment. In diagnosing the condition, the students have to discuss the symptoms, generate hypotheses based on whatever knowledge and experience they have and identify learning issues. At the end of each session, the students reflect verbally on their current hypotheses and each student assumes responsibility for investigating one of more of the identified learning issues through self-directed learning.

A second authentic, constructivist approach to learning is *project-based learning (PBL)*. Esch (1998) offers two helpful continua for distinguishing between problem-based and project-based learning:

- *The extent to which the end product is the organizing center of the project.* At one end of this continuum, end products are elaborate and shape the production process and, at the other end, end products are simpler and more summative, such as a group's report on their research findings. The former case typifies project-based learning, where the end product drives the planning, production, and evaluation process and the latter, where the inquiry and research is the primary focus of the learning process, typifies problem-based learning.

- *The extent to which a problem is the organizing center of the project.* In this case, at one end are projects in which it is implicitly assumed that any number of problems will arise and students will require problem-solving skills to overcome them and, at the other end, are projects that begin with a clearly articulated problem and require a set of conclusions and/or solution. Again, the former example typifies project-based learning and the latter typifies problem-based learning.

In both problem-based and project-based learning, the teacher (facilitator) is available for consultation and plays a significant role in modeling the metacognitive thinking associated with the problem-solving processes. These reflect a *cognitive apprenticeship* environment (Collins *et al.*, 1990) with *coaching* and *scaffolding* (the support provided by an expert as the novice attempts a task such as offering hints, reminders, and feedback) provided to support the learner in developing metacognitive skills. As these skills develop, the scaffolding is gradually removed. The intention is to force learners to assume as much of the task on their own, as soon as possible. The cognitive apprenticeship model also advocates:

- *modeling*, which involves an expert (the teacher) performing a task so that the learner can observe and build a conceptual model of the processes required to accomplish it;
- *articulation* (either verbal or written) to encourage learners to communicate their knowledge and thinking;
- *exploration*, to push learners into a mode of problem-solving on their own;
- *reflection*, as previously discussed.

A similar concept to articulation that has been cited as an important element is *debriefing*, which provides the opportunity for learners to consolidate their experience and assess the value of the knowledge they have obtained in terms of its theoretical and practical application to situations that exist in reality (Kriz, 2003).

4. GUIDING PRINCIPLES FOR THE LEARNING ENVIRONMENT

As a consolidation of the above research work, we put forward our own principles for the learning environment as follows:

1. Allow learners to choose a (thickly) authentic project grounded in professional practice. The project should be sufficiently complex to develop analytical and problem-solving skills. It should also be both personally meaningful (to facilitate intrinsic motivation) and relate to the real-world outside the classroom. The latter implies the project should be group-based (although it may be challenging for the team to find a project that is personally meaningful to all team members).
2. Encourage learners to take responsibility (ownership) for learning and to be aware of the knowledge construction process.
3. Allow learners to develop their own processes to reach a solution.
4. Provide learners with the opportunity to experience and appreciate other perspectives (this may come about as part of the next principle).
5. Provide opportunities for interaction and collaboration (learner-learner, learner-teacher, or learner-system).
6. For group-based work, there must be "group goals" and "individual accountability" for effective collaborative learning (Slavin, 1989).
7. Ensure that the learning environment motivates, engages, and challenges the learner. The environment should support the preference of the learners (Connolly *et. al.*, 2006).
8. Provide feedback mechanisms to enable learners to be fully aware of their progress.
9. Provide support mechanisms for learners using coaching and scaffolding (which should gradually be removed).
10. Be flexible to support different learning styles.
11. Encourage learners, and provide mechanisms for learners, to articulate knowledge and thinking throughout the project.
12. Encourage learners, and provide mechanisms for learners, to reflect on their activities both during the project and after completion of the project. This reflection should be both group-based and individual-based.
13. Provide opportunities for debriefing at the end of the project.
14. Provide an integrated assessment (in our case, the instrument of assessment is the project itself, which can be assessed in a variety of ways).

4.1 Applying These Principles to Our Teaching

We have used the above constructivist principles built around the cognitive apprenticeship model and project-based learning to teach two modules in our undergraduate/postgraduate courses: Business Database Systems (BDS), a core or optional third year module in all our undergraduate courses, and Fundamentals of Database Systems (FDBS), a core module in our MSc Information Technology course, a conversion course for non-computing graduates. In both cases, the students were reasonably experienced learners, although in the latter case, not experienced computing students. We have taken a similar approach for both modules (one online and one using a more traditional face-to-face approach).

4.1.1 Fundamentals of Database Systems (FDBS): The FDBS module runs in a traditional face-to-face mode for full-time and part-time groups and since session 2001/2 in a fully online format for a part-time group. Since session 2002/3, we have used a constructivist learning environment for the online group. The online group typically consists of

15-25 students, all from similar professional backgrounds. Scaffolding is provided through the teacher (facilitator) as well as through the creation of visualizations for a number of database concepts (eg. ER modeling, normalization, mapping an ER model to relations) and lower-level online units covering the relevant module material. When the students encounter problems they can drill down to the relevant material or use the higher-level visualizations. The prescribed textbook for this module is Database Systems (Connolly and Begg, 2005) and so ER modeling is taught using the UML class diagram notation in the first instance with other notations introduced later in the module. The use of UML provides a seamless integration with the other analysis, design, and systems development modules in our courses. However, when students are working with companies they are allowed to use the notation that the company may dictate. In the early stages, asynchronous online tutorials are run to discuss worked examples covering activities that groups would have to undertake as part of database analysis and design. It is important that students fully understand these examples and can apply the principles in the different contexts they will find themselves in.

The students self-select themselves into groups of size 3-4 and each group chooses a project that is of interest to all group members. These projects are generally from Small and Medium-sized Enterprises (SMEs) in the West of Scotland, which has the added advantage of benefiting these businesses and thereby the local economy. The facilitator provides background advice to ensure that a group does not take on a project that is too large or complex or alternatively too trivial. Students are encouraged to keep sufficiently detailed and formal records of their work and, in particular, the decisions made with supporting justifications. They are also encouraged to frequently reflect on these decisions and the processes that led to the decisions both as a group and as individuals. Each group/individual is given scope to use whatever tools they feel most appropriate and most comfortable with (to support cognitive preference). The FirstClass Virtual Learning Environment (VLE) is used for the online material and this system provides email facilities and discussions boards, both public (ie. available to the facilitator) and private (a students-only discussion area). Interestingly, while groups initially use these basic facilities, they also develop their own wikis and blogs, while using voice-over-IP tools like Skype and mobiles/instant messaging for more urgent communication. Groups use laptops and PDAs for recording meetings with the clients and the facilitator.

Support is provided by the facilitator as and when necessary but this is only in an advisory capacity: groups are not provided with solutions or partial solutions but are instead directed to where appropriate information can be found. This reinforces the principles of constructivism and emphasizes to the students that they are acting as professional database design consultants and have to act in this capacity. Debriefing is conducted at the end for all parties (facilitators, students, and clients) to reflect on the learning outcomes and to reflect on issues that had arisen in the performance of the projects. We discuss some of these issues in the next section.

4.1.2 Business Database Systems (BDS): The BDS module has a pre-requisite requirement that students have previously undertaken an introductory database module. The BDS students are experienced learners in terms of time spent in tertiary education and in their knowledge of database systems. The coursework set for the introductory database module uses the traditional small systems solution case or project-based teaching case approach (which we refer hereafter to simply as the "case study" approach for conciseness) and the fact that students are taking the higher level module is evidence that they were able to demonstrate appropriate knowledge and skills in the database systems domain using this approach.

Since session 2000/1, we have used the project-based approach for the BDS module. This approach was introduced because it was felt that having an assessment based simply on a more complex case study would be inadequate to extend the students' knowledge and skills towards that required of a professional database designer. In recent years the number of students taking the BDS module has steadily grown reaching 180 students in the session 2004/2005, spread across two semesters. Now, the facilitation of the projects is undertaken by a group of faculty, although the increasing numbers are testing the scalability of the project-based learning approach, as we discuss later.

In comparison to the online delivery method used in FDBS, BDS is delivered using traditional face-to-face teaching methods made up of lectures, tutorials, and laboratory sessions. However, all teaching resources are also available online using the Blackboard VLE. The BDS module introduces new concepts and extends concepts presented in the introductory database module. New material includes fact-finding techniques, physical database design, monitoring and tuning the operational database, transaction management, concurrency control, backup and recovery, and extended treatment of ER modeling, normalization, SQL, relational DBMSs, CASE tools, and the database system development lifecycle. Again, ER modeling is taught using UML for the reasons cited above for the FDBS module. Some tutorials are used to ensure that material presented in the introductory database module is well understood so that further learning occurs on top of a solid foundation of student appreciation and understanding.

The project-based learning approach used in the BDS module is also in the form of a group coursework. The coursework runs as described above for the FDBS module. We have not performed empirical analysis of the approach in the teaching of the BDS module, and so the results discussed in the following section for the BDS module are based on anecdotal evidence, which has been accumulated over several years of practice by the authors.

5. RESULTS

This section presents some preliminary findings from using the project-based learning approach to teach database analysis and design in the FDBS and BDS modules. A quantitative analysis of students' performance in the FDBS module is presented in Connolly *et al.* (2006). The paper compares the performance of 977 students divided into three groups, one of which used the constructivist project-based approach albeit through online delivery. The evidence supports our view that the constructivist approach can improve student learning. The results were not fully conclusive because the effect could have been entirely attributable to online delivery rather than the project-based approach and further quantitative research is required. However, the qualitative analysis of student and faculty feedback from both the FDBS and BDS modules that we undertook in parallel provides some interesting results to further support our view as we now discuss.

The student feedback was obtained from end-of-module questionnaires and faculty feedback from interviews. Generally, student feedback was extremely positive, all students reporting that they had enjoyed the experience. They were able to compare this approach with the more traditional case study approach that they had encountered in their previous studies and had felt that the project-based approach with learning *in situ* had provided a better, more motivating, more engaging method to learn about database analysis and design. They also appreciated that this approach gave them relevant work experience that could help their employment prospects on completion of the course. The students were also very receptive to the concept of a reflective journal and, while it was sometimes difficult to find the time to maintain it, many reported that they had benefited from this approach and would keep a reflective journal for the remainder of their studies and into employment. On the negative side, most students reported that the workload was significantly higher than in other modules. They also found time-management was an issue, particularly as they had no real feeling at the outset for scope and complexity of the projects they had selected (many were led by their enthusiasm for working as a professional consultant). All were in agreement that the approach should be extended to other modules, but rather than having a project per module, they suggested that one assessment-based integrative project that extended over a number of modules would be an extremely powerful approach to teaching and learning. This was something faculty had discussed on several occasions but had never progressed the idea for resource reasons rather than pedagogic reasons.

Faculty were also enthusiastic of this approach and felt the students had learned more than with the case study approach, particularly in areas not traditionally covered in the database modules (application of fact-finding techniques, and people-oriented and business-oriented skills). It was important that sufficient guidance was given during the project, particularly in the early stages when the groups were selecting projects (as noted above, student enthusiasm had to be tempered with realistic expectations). At the same time, as students were now working in an environment that had not been purpose-built for their effective learning, care had to be taken to ensure students were not overwhelmed with all the complexities that a real-world project can present, otherwise their initial enthusiasm quickly dissipated. The students needed quite a lot of guidance with both group and personal reflection initially until they found tools they were comfortable with (eg. wikis, blogs).

Typically each faculty member handled between 4-6 project groups compared to sometimes as many as 20 groups with the case study approach. Nevertheless, faculty found that their workload was significantly higher than with traditional approaches and that it was necessary to develop in-depth knowledge of each industrial project to be able to support the students effectively. This gave rise to grave concerns over scalability and faculty felt that they could not have coped with any further project groups.

Faculty observed that students generally underestimated the time required to undertake the project and the facilitator needed to discuss the similarities and differences between case study assessments and project-based assessments in terms of student effort. For example, some students underestimated the time spent securing a company's involvement in their project and establishing that relationship cannot always be rushed to fit a timescale that suits the students and meets the demands of faculty. It was also important that the facilitator identified the gaps in the students' knowledge and skills and directed them to appropriate sources to enable them to undertake the project. Failing to do this in a timely manner, led some students to lose confidence and meant they simplified and converted the project into a form of case study that they could cope with. However, this should and can be avoided with sufficient support from the facilitator to encourage students to accept the realities and complexities of project-based learning as a positive aspect of their work. It is the students' ability to cope with and manage the project within this environment that is being assessed and therefore it is necessary that they do not ignore or smooth over the problems of working with a real company.

While assessments based on case studies for database analysis and design usually present a simplified and contrived set of requirements that the students then analyze and solve, our project-based learning approach requires that the students must first capture the requirements for the new database. Capturing requirements means that the students use fact-finding techniques that may be known in theory but not practiced. Students must be guided carefully through this early stage of the lifecycle as the true complexity of real database requirements becomes apparent. Therefore, while case study assessments cover requirements analysis through to physical database design and possibly thereafter to implementation, project-based assessments extend the coverage of the database system development lifecycle from the systems definition stage through to implementation. It is therefore clear that the skills required to undertake project-based learning differs to that of the case study approach.

As the success of this approach is dependent on the support of industry to put forward and assist with project-based learning, faculty emphasized that the facilitator must carefully guide students in their relationship with the company while ensuring that students achieve the specified learning outcomes. This sometimes required significant diplomacy from the facilitator when the academic objectives did not fully match the commercial objectives. It is important that faculty explain to companies at the outset what constitutes reasonable expectations for parameters such as project size, project complexity, and overall timescales. However, in most cases, both students and companies benefited from the relationship and this is why project-based learning has been well supported by local companies over the last few years.

Occasionally, faculty encountered problems with group dynamics, for example, autonomous students tend to prefer to work individually, there can be lack of group cohesion, dominant group members, insecure group members, and free-riders (referred to in group dynamics research as "diffusion of responsibility"). To highlight that these can occur in industry and need to be overcome, students were encouraged to tackle these problems as a group and only in extreme cases did faculty intervene to facilitate a solution acceptable to all.

There was agreement among faculty that the project-based learning approach was pedagogically sound for postgraduate courses and for third/fourth years of undergraduate courses, but were reluctant to use this approach in first or second year, on the grounds that students may not be sufficiently mature learners and may not have developed the necessary discipline and time-management skills required. Further, it was generally felt that the facilitator had to have a fairly extensive knowledge of Computing/IT and a solid foundation in business concepts to be able to handle the variety of projects that students selected with project-based learning.

6. CONCLUSIONS AND FUTURE DIRECTIONS

This paper has examined some of the issues surrounding the teaching of database analysis and design and has described a teaching approach motivated by principles found in the constructivist epistemology to help overcome these issues and provide the learner with the skills necessary to understand and perform database analysis and design effectively. The constructivist approach used is based on the cognitive apprenticeship model and project-based learning.

The approach used points toward learning about design by *doing* design, and relying less on overt lecturing and traditional teaching. Design is learned by becoming a practitioner, albeit for the duration of the module, not merely by learning about practice. In brief, students should engage in challenging problems that reflect real-world complexity. The problems should be authentic and ill-structured; that is, they should not have one predetermined, foregone solution but rather be open to multiple interpretations and multiple 'right answers'. Students should engage in actively working on solving the problem in collaborative groups to reflect the social nature of learning.

This approach requires a shift in the roles of both students and faculty. The student becomes a cognitive apprentice, exploring and learning about the problem in the presence of peers. Faculty shifts from overt lecturing to becoming a facilitator who assists students in developing an understanding of the professional practice of database analysis and design.

The paper presents some preliminary results of this work that suggests the approach can be used successfully to teach students how to design effective modern information systems. The qualitative findings show that students and faculty reacted extremely positively to the approach and found it more motivating and engaging than the more traditional case study approach. However, both students and faculty found the workload higher than with more traditional teaching methods and that scalability was an issue. Faculty also felt that this approach required mature learners and may not be entirely appropriate for first and second year undergraduates.

There are two directions for our future research:

1. *Extended quantitative analysis.* Further empirical data will be gathered to test whether there are any observable differences between the project-based approach and the more traditional case study approach.
2. *Use of ePortfolios.* A strategic decision has been made to implement personal development planning across the University starting this session (session 2005/6). As part of this work, we have procured the full BlackBoard Content Management System, which incorporates an ePortfolio tool. We are currently investigating the use of ePortfolios within the FDBS module as a form of reflective journal.

7. REFERENCES

ACM/IEEE. (2001), ACM/IEEE Computing Curricula. December 15 2001. http://www.computer.org/education/cc2001/ (last access date 23 July 2004).

Argyris, C. and D.A. Schön (1974), Theory in practice: Increasing professional effectiveness. Jossey Bass, San Francisco.

Armarego, J. (2002), "Advanced Software Design: A Case in Problem-Based Learning." Proceedings of the 15th Conference on Software Engineering Education and Training, 25–27 February 2002, Covington, Kentucky, USA, pp. 44–54.

Ben-Ari, M. (2001), "Constructivism in Computer Science Education." Journal of Computers in Mathematics and Science Teaching, Vol. 20, No. 1, pp. 45–73.

Boyle, T. (2000), "Constructivism: A suitable pedagogy for Information and Computing Science?" Proceedings of 1st Annual Conference of the LTSN Centre for Information and Computer Sciences, Heriot-Watt, Edinburgh, August 2000.

Brown, J.S., A. Collins, and P. Duguid (1989), "Situated cognition and the culture of learning." Educational Researcher, Vol. 18, No. 1, pp. 32–42.

Bubenko, J. (1995), "Challenges in Requirements Engineering: keynote address." Second IEEE International Symposium on Requirements Engineering, York, England.

Budgen, D. (1995), "Is teaching software design a 'wicked' problem too?" Proceedings of the 8th SEI Conference on Software Engineering Education, New Orleans (La), pp. 239-254.

Codd, E.F. (1970), "A relational model of data for large shared data banks." Comm. ACM, Vol. 13, No. 6, pp. 377–387.

Collins, A. (1991), "The role of computer technology in restructuring schools." Phi Delta Kappan, 73, pp. 28–36.

Collins, A., J.S. Brown, and A. Holum (1991), "Cognitive Apprenticeship: Making Thinking Visible." American Educator, Vol. 15, No. 3, pp. 6–11, 38–46.

Collins, A., J.S. Brown, and S.E. Newman (1989), "Cognitive Apprenticeship: Teaching the Craft of reading, writing, and mathematics." In L. Resnick (ed.), Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser. Hillsdale, NJ: Lawrence Erlbaum.

Connolly, T.M. (2005), "Database Technology Professional European Survey Analysis." International Conference on Interactive Technology in Education, 20-22 April, Hämeelinna, Finland.

Connolly, T.M. and C.E. Begg (2005), Database Systems: A practical approach to design, implementation, and management. Addison Wesley Longman, London.

Connolly, T.M., E. MacArthur, M.H. Stansfield, and E. McLellan (2006), "A Quasi-Experimental Study of Three Online Learning Courses in Computing." Journal of Computers and Education (accepted for publication).

Connolly, T.M., M.H. Stansfield, and E. McLellan (2005), "An Online Games-Based Collaborative Learning Environment to Teach Database Analysis and Design." Proceedings of 4th IASTED International Conference on Web-Based Education, Grindelwald, Switzerland, February 2005.

Cook, J. and T. Boyle (2000), "Effective delivery of on-campus networked learning: reflections on two Case-Studies." Proceedings of the 2nd International Conference on Networked Learning, April 17 - 19, University of Lancaster, UK.

Cunningham, D.J. (1991), "Assessing constructions and constructing assessments: a dialogue." Educational Technology, Vol. 31, No. 5, pp. 13–17.

Cunningham, D., T.M. Duffy, and R. Knuth (1993), "Textbooks of the future." In Hypertext: A psychological perspective, C. McKnight ed. Ellis Horwood Publishing: London, pp. 19–49.

Deci, E.L. and R.M. Ryan (1985), Intrinsic motivation and self-determination in human behavior. Plenum, New York.

DeGrace, P. and L. Hulet Stahl (1998), Wicked Problems, Righteous Solutions: A Catalog of Modern Engineering Paradigms, Prentice Hall.

Dewey, J. (1958), Art as experience. Capricorn Books, New York.

Eaglestone, B. and J.M. Baptista Nunes (2004), "Pragmatics and practicalities of teaching and learning in the quicksand of database syllabuses." Journal of Innovations in Teaching and Learning for Information and Computer Sciences.

Esch, C. (1998), "Project-Based and Problem-Based: The same or different?" Available at: http://pblmm.k12.ca.us/PBLGuide/PBL&PBL.htm (last access date 11 September 2005).

EUCIP. (2003), EUCIP (European Certification of Informatics Professionals) Core Syllabus. March 2003. http://www.eucip.com/DownloadFiles/Core_Syllabus_March_2003.pdf (last access date on 23 July 2004).

Fowler, L., J. Armarego, and M. Allen (2001), "CASE Tools: Constructivism and its Application to Learning and Usability of Software Engineering Tools." Computer Science Education, Vol. 11, No. 3, pp. 261–272.

Forman, E. and J. McPhail (1993), "Vygotskian perspectives on children's collaborative problem-solving activities." In Contexts for learning. Sociocultural dynamics in children's development, E.A. Forman, N. Minick, and C. Addison Stone, eds. Oxford University Press, Oxford.

Gance, S. (2002), "Are constructivism and computer-based learning environments incompatible?" Journal of the Association for History and Computing, Vol. V, No. 1, May 2002.

Grabinger, R.S. and J.C. Dunlap (1995), "Rich environments for active learning: a definition." Association for Learning Technology Journal, Vol. 3, No. 2, pp. 5–34.

Gruba, P. and H. Søndergaard (2001), "A constructivist approach to communication skills instruction in computer science." Computer Science Education, Vol. 11, No. 3, pp. 203–219.

Hadjerrouit, S. (1999), "A Constructivist Approach to Object-Oriented Design and Programming." Proceedings of 4th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE'99), Cracow, Poland, SIGCSE Bulletin, Vol. 31, No. 3, pp. 171–174.

Hadjerrouit, S. (2003), "Toward a Constructivist Approach to E-Learning in Software Engineering." Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Norfolk, VA, pp. 507–514.

Illeris, K. (2003), "Towards a contemporary and comprehensive theory of learning." International Journal of Lifelong Learning, Vol. 22, No. 4, pp 396-406.

Jonassen, D.H. (1994), "Thinking technology: Toward a constructivist design model." Educational Technology, Vol. 34, No. 3, pp. 34–37.

Koehler, M.J. and P. Mishra (2005), "Teachers Learning Technology by Design." Journal of Computing in Teacher Education, Vol. 21, No. 3, Spring 2005.

Kriz, W.C. (2003), "Creating effective learning environments and learning organizations through gaming simulation design." Simulation and Gaming, Vol. 34, No. 4, pp. 495–511.

Kroenke, D. (2004), Database Processing: Fundamentals, Design, and Implementation, 9th edn, Pearson Prentice Hall, Upper Saddle River, New Jersey.

Lave, J. (1991), "Situating learning in communities of practice." Washington, DC: American Psychological Association.

Lave, J. and E. Wenger (1991), Situated learning: Legitimate peripheral participation, Cambridge University Press, Cambridge, England.

Lederman, L.C. and F. Kato (1995), "Debriefing the debriefing process." In Simulation and gaming across disciplines and cultures: ISAGA at a watershed, D. Crookall and K. Arai, eds. Sage, Thousand Oaks, CA.

Malone, T. (1981), "Towards a theory of intrinsically motivating instruction." Cognitive Science, No, 4, pp. 333–369.

Mohtashami, M. and J.M. Scher (2000), "Application of Bloom's Cognitive Domain Taxonomy to Database Design." Proceedings of ISECON (Information Systems Educators Conference) 2000, Philadelphia.

Papert, S. (1980), Mindstorms: children, computers and powerful ideas, Basic Books.

Parker, D. and A. Gemino (2001), "Inside online learning: Comparing conceptual and technique learning performance in place-based and ALN formats." Journal of Asynchronous Learning Networks, Vol. 5, No. 2, pp. 64–74.

Piaget, J. (1968), Six Psychological Studies. Vintage Books, New York.

Postman, N. and C. Weingartner (1971), Teaching as a Subversive Activity, Penguin:London.

Pullen, M. (2001), "The Network Workbench and Constructivism: Learning Protocols by Programming." Computer Science Education, Vol. 11, No. 3, pp. 189–202.

Reeves, T.C. (1994), "Evaluating What Really Matters in Computer-Based Education." In Computer Education: New Perspectives, M. Wild and D. Kirkpatrick eds., 219-246, Edith Cowan University Press, Perth.

Savery, J.R. and T.M. Duffy (1995), "Problem based learning: An instructional model and its constructivist framework." Educational Technology, 35, pp. 31–38.

Schön, D.A. (1983), The Reflective Practitioner: How Professionals Think in Action. Basic Books, New York.

Schön, D.A. (1987), Educating the Reflective Practitioner: Towards a New Design for Teaching in the Professions. Jossey-Bass Inc., San Fransisco.

Shaffer, D.W. and M. Resnick (1999), "Thick authenticity: New media and authentic learning." Journal of Interactive Learning Research, Vol. 10, No. 2, pp. 195–215.

Shaffer, D.W. (2004a), "Pedagogical Praxis: The Professions as Models for Postindustrial Education." The Teachers College Record, Vol. 106, No. 7, pp. 1401–1421.

Shaffer, D.W. (2004b), "Epistemic frames and islands of expertise: Learning from infusion experiences." International Conference of the Learning Sciences (ICLS), Santa Monica, CA. Available at http://www.education.wisc.edu/edpsych/facstaff/dws/papers/epistemicframesicls04.pdf (last access date 28 July 2005).

Slavin, R.E. (1989), "Research on cooperative learning: An international perspective." Scandinavian Journal of Educational Research, Vol. 33, No. 4, pp. 231–243.

Taxén, G. (2003), "Teaching Computer Graphics Constructively." Proceedings of International Conference on Computer Graphics and Interactive Techniques, San Diego, California, pp. 1–4.

Van Gorp, M.J. and S. Grissom (2001), "An Empirical Evaluation of Using Constructive Classroom Activities to Teach Introductory Programming." Computer Science Education, Vol. 11, No. 3, pp. 247-260.

Waks, L.J. (2001), "Donald Schon's Philosophy of Design and Design Education." International Journal of Technology and Design Education, 11, pp. 37–51.

Wilson, B. (1996), Constructivist learning environments: Case studies in instructional design. Educational Technology Publications, New Jersey.

Yazici, S., T. Boyle, and T. Khan (2001), "Towards a multimedia learning environment for object oriented design." Proceedings of the 2nd Annual Conference of the LTSN Centre for Information and Computer Sciences, London, August 28–30.

AUTHOR BIOGRAPHIES

Thomas M Connolly is a Professor in the School of Computing at the University of Paisley, having managed the Department for several years. He worked for over 15 years in industry as a Manager and Technical Director in international software houses before returning to academia. His specialisms are database systems and online learning. He is co-author with Carolyn Begg of the highly successful academic textbooks Database Systems (now in its 4th edition) and Database Solutions (in its 2nd edition) and soon to be published Business Database Systems.

Carolyn E Begg is a Lecturer in the School of Computing at the University of Paisley. She specializes database design, data warehousing, and data mining, and in the application of database systems in biological research. As noted above, she is co-author with Thomas Connolly of the highly successful academic textbooks Database Systems and Database Solutions.