Journal of Information Systems Education

Volume 34
Issue 2
Spring 2023

Teaching Tip

Teaching Expert Systems in a Postgraduate Accounting Degree Program

Pak-Lok Poon

Recommended Citation: Poon, P.-L. (2023). Teaching Tip: Teaching Expert Systems in a Postgraduate Accounting Degree Program. *Journal of Information Systems Education*, 34(2), 131-141.

Article Link: https://jise.org/Volume34/n2/JISE2023v34n2pp131-141.html

Received: May 26, 2022
Revised: August 4, 2022
Accepted: September 2, 2022
Published: June 15, 2023

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 34(2), 131-141, Spring 2023

Teaching Tip
Teaching Expert Systems in a Postgraduate Accounting Degree Program

Pak-Lok Poon
School of Engineering and Technology
Central Queensland University
Melbourne 3000, Australia
p.poon@cqu.edu.au

ABSTRACT

Teaching the concepts of expert systems to accounting students is often challenging, not only because these students are not equipped with the necessary technical knowledge to comprehend the concepts taught, but also due to the other various constraints (e.g., limited teaching hours and a diverse background of the students' undergraduate disciplines) imposed on teaching and learning. A further difficulty is how to design a hands-on project so that, on the one hand, learning can be reinforced and, on the other hand, students are able to complete the projects with their limited technical knowledge and background. This article shares our experience in teaching an expert system component and designing a corresponding project in one of the first information technology courses for accounting students at the master's level. A subsequent student feedback evaluation exercise was performed, showing that students were able to understand the expert system concepts and that they considered the project interesting and useful.

Keywords: Expert systems, Hands-on project, Course development, Student engagement, Accounting education

1. INTRODUCTION

Nowadays, information technologies (IT), including expert systems (ES), have diffused into many sectors (including accounting and finance) of our society (Abouloula et al., 2019; Al Janabi, 2019; Bresnahan & Yin, 2017). Thus, today's accounting professionals need to possess an adequate knowledge of ES so that they are able to: (a) work as professional and effective ES users, (b) understand the technical jargons and terminology commonly used by ES developers for better communication, and (c) participate in and make contributions to an ES development project when they become members of the project team (Baldwin et al., 2006; Changchit & Holsapple, 2004; Pan & Seow, 2016; Wagner et al., 2002). To respond to this demand, accounting schools in many universities have incorporated some IT courses (with an ES component) into their accounting degree programs to equip graduates with the necessary technical skills and knowledge (Brown et al., 1995; Jackson & Cherrington, 2001; Lymer, 1994; Sangster, 1991; Stancheva-Todorova & Bogdanova, 2021). This article discusses the experience and insight into teaching ES concepts and designing a corresponding hands-on group project for accounting students at the master's level in a university at which the author previously taught.

2. BACKGROUND

This article refers to a university located in Hong Kong, which we call "UNIV" to make it anonymous. In UNIV, its Accounting School has been offering coursework-based accounting degree programs at both bachelor's and master's levels for many years. This article focuses on the School's Master of Accounting (MAcc), which is a "conversion" degree program that allows people without an undergraduate accounting qualification to change their career tracks to accounting (although some applicants with an undergraduate accounting qualification will also be admitted to the program).

This MAcc has a one-semester core course "Contemporary Issues in IT" (hereafter simply referred to as "CI-IT"), which covers various kinds of IT concepts and applications (e.g., software development, end user computing, ES, decision support systems, and enterprise resource planning systems). The main aim of CI-IT is to equip students with the necessary IT knowledge so that they can work as professional IT *users* and are able to communicate effectively with IT practitioners by understanding the commonly used technical jargon and terminology (rather than being trained to become competent IT practitioners such as software developers).

CI-IT has only one class (in the form of a three-hour seminar) per week in the offering semester. Over the years, the class size has been 40–50 students, of which about 80% are enrolled in full-time mode and 20% in part-time mode. Because most part-time students are working full time during their studies, CI-IT (and all the other MAcc courses) has been offered in the evening to accommodate working professionals.

3. CHALLENGES ON ES TEACHING AND ASSESSMENT

From this section onwards, the discussion is based on the author's last teaching of CI-IT at UNIV. In CI-IT, the following challenges were encountered in designing and using a group project for the ES component:

- ***Problem 1 (Student Background):*** There was a diverse background of the students' undergraduate disciplines. The class of CI-IT had 42 students whose undergraduate disciplines are shown in Table 1. The diverse background created a great difficulty in determining the right level and number of technical details of ES to teach. Teaching too much technical detail would have created learning problems to students with a non-IT background (e.g., accounting/finance, business administration, marketing, and HRM). On the other hand, teaching too little or superficial technical detail would have left students with an IT background bored and unsatisfied.

Undergraduate Disciplines	Number of Students
Accounting/finance	13
Business administration/ commerce/management	12
IT (e.g., computer science & information systems)	7
Marketing	4
Engineering	3
Education	2
Human resource management (HRM)	1

Table 1. Students' Undergraduate Disciplines

- ***Problem 2 (Topic Selection):*** Many aspects of ES could possibly be taught, e.g., planning, management, selection, design, implementation, and use. Since only two consecutive weeks (six teaching hours in total) were allocated to teach ES, it was impracticable to cover every aspect of ES in detail. Thus, a decision had to be made about what aspects of ES to teach and emphasize.
- ***Problem 3 (Project Design):*** Using project-based learning (PBL) has several merits (Ellis, 2021; Guo et al., 2020; Miller & Krajcik, 2019; Thomas, 2000). For example, PBL projects: (i) focus on questions or problems, thereby driving students to encounter (and struggle with) the central concepts and principles of a discipline; and (ii) involve students in a constructive investigation (Thomas, 2000). In view of these merits, CI-IT included an ES group project to complement and reinforce students' learning. Designing this hands-on group project and its associated assessment scheme involved the following issues:
    a) ***Group Heterogeneity:*** Some groups comprised both students with IT and non-IT backgrounds. In these groups, students with an IT background might prefer their projects to address more on the "technical" aspects (e.g., designing and implementing a "working" system) of an ES. However, those students with a non-IT background might prefer projects that emphasized the "non-technical" aspects (e.g., planning, management, and selection) of an ES.
    b) ***Project Emphasis:*** As mentioned in Problem 2 (Topic Selection), six teaching hours were insufficient to cover every aspect of an ES in detail. If some aspects of ES were only touched on during teaching, then students should not be expected to demonstrate sophisticated knowledge of those aspects in their projects. Thus, a question arose: What aspects should be selected for emphasis in and assessment of the group project?
- ***Problem 4 (Application Domain):*** To achieve good learning outcomes, a project should be designed to engage students in solving *real-world* problems. However, it was challenging to choose an appropriate application domain for the ES project. Because of the diverse background of the students (in terms of their undergraduate qualifications and working experiences), regardless of which application domain (e.g., accounting, finance, engineering, and HRM) was chosen, students would have lacked exposure to it. Those students might lose interest in the group project, which would lead to poor learning outcomes.

In the previous discussion, Problems 1 (Student Background) and 2 (Topic Selection) were related to teaching, whereas Problems 3a (Group Heterogeneity), 3b (Project Emphasis), and 4 (Application Domain) were related to the ES group project.

4. OUR TEACHING, PROJECT DESIGN, AND ASSESSMENT APPROACH

To address the aforementioned problems, the following approach of teaching, project design, and assessment was adopted:

4.1 Teaching Approach

This section is concerned with addressing Problem 2 (Topic Selection) stated in Section 3. Since CI-IT was a course for postgraduate accounting (not IT) students, the ES component was designed to equip students (as ES *users*) with the necessary knowledge of the following two aspects:

a) ***Communication:*** To communicate effectively with ES developers by understanding their technical jargon and terminology.
b) ***Contribution:*** To participate in and make contributions to an ES project if they were part of the project team.

To make the teaching of ES (especially its development) more comprehensible (particularly for those students with a non-IT background), the *ES Development Life Cycle* discussed by Turban et al. (2011) was used as the framework to guide teaching. Turban et al. (2011) argued that developing an ES largely follows a life cycle of activities, like the traditional software development life cycle (SDLC) for conventional systems (Sommerville, 2015). By using an ES development life cycle framework to guide teaching, we supported our students in developing a holistic view of all the phases and activities that an ES project would go through, from initial project inception to postimplementation maintenance and evaluation. Students also understood how these phases and activities were interconnected. Not only did this knowledge help the students understand their roles and contributions (as an ES *user*) at different stages of a real-life ES development project, it also provided a useful clue to work on the ES group project (to be discussed in Section 4.2).

In Turban's ES Development Life Cycle (Turban et al., 2011), an ES project goes through six major phases with 27 associated tasks (see Appendix A). To cover the most major lifecycle concepts in the first three-hour seminar, some relatively minor tasks were omitted without affecting achieving purposes (a) and (b) above. Accordingly, a cut-down version of the ES Development Life Cycle (with only 19 associated tasks) was used in teaching (see Appendix B). Due to page constraints, it is not possible to discuss all 19 associated tasks in this article. Readers are referred to Turban's textbook (Turban et al., 2011) for the details.

The second three-hour seminar was used to cover other aspects of ES outside development (recall that the ES component in CI-IT was covered in two three-hour seminars). These non-development aspects include, for example, ES history, different ES categories (e.g., interpretation, prediction, diagnosis, monitoring, and repair), types of knowledge (declarative knowledge, procedural knowledge, and meta-knowledge), inferencing (forward chaining and backward chaining), explanation capability, and some commercial ESs used in different application domains (e.g., taxation). In addition, the differences between an ES and a knowledge-based system were highlighted.

Our experience was that six-hour teaching was sufficient to achieve purposes (a) and (b) as stated in the beginning of this section.

4.2 Project Design

Appendix C shows the ES group project handout. This group project accounted for 25% of the total assessment of CI-IT (the remaining 75% was allocated to other IT (non-ES) components), with the following *project-specific* learning outcomes:

- (LO1) To understand the situations under which an ES (rather than a "conventional" non-ES system) should be developed.
- (LO2) To understand the managerial issues associated with an ES development project.
- (LO3) To understand the design/implementation issues and concepts associated with an ES development project.

Each group was required to make a presentation in class at the end of the project during weeks 10 and 11. For fairness, every group had to submit presentation slides at the end of Week 9. After a submission was made, further changes to the presentation slides were not allowed. Since the cut-down version of Turban's ES Development Life Cycle (see Appendix B) was taught in Week 3, each group had about six weeks to work on the project.

Overall, the project was designed in accord with the two purposes (a) and (b) stated in Section 4.1. Thus, the group project did not aim to prepare students to be competent ES developers. With this in mind, students were told of the following:

1) Each group comprised four or five students, and students were allowed to form groups of their own choice.
2) Each group selected two or three members to make a 40-minute presentation in the class on behalf of the group. After the presentation, there was a 15- to 20-minute discussion, chaired by the presentation group.
3) Each group could freely choose the project's application domain, even if it were outside the accounting/finance area. In the class, nine project groups were formed. Their self-chosen application domains are listed in Table 2. Among the nine self-chosen application domains, only one was related to accounting/finance. A major reason was that CI-IT was offered in Semester 1 of Year 1 (recall that MAcc was a "conversion" program so not many students were admitted with an undergraduate accounting or finance qualification; see Table 1). Thus, at the time of studying CI-IT, most students do not have sufficient accounting/finance knowledge to enable them to choose this application domain for their projects.

Domain Categories	Application Domains
Finance	Wealth management
Property	Dream home selection
	Housing rental
Education	Overseas study consultancy
Healthcare	Skin care treatment
	Obesity management
	Nutrition advice
Leisure & hobby	Weekend holiday planning
	Professional camera selection

Table 2. Different Application Domains of Students' ES Projects

In general, a group tended to choose a particular application domain when some group members considered themselves "knowledgeable" on that domain since domain knowledge was essential for defining the relevant production rules in the knowledge base as discussed in item 4(b) later. For example, all the students in the group that chose skin care treatment as the application domain were females who had been using skin care products regularly. The arrangement of self-chosen application domains alleviated Problem 4 (Application Domain) in Section 3, where students in a group might not have the relevant knowledge of an instructor-assigned application domain. Another merit of this arrangement was that students found the ES projects with their chosen application domains more interesting and, hence, were inclined to devote more effort to them.
4) The project should be undertaken largely by following the cut-down version of the ES Development Life Cycle in Appendix B. This gave students a general idea about what aspects or elements their projects should cover. Note that some tasks (e.g., Tasks 3.1 and 4.1) in Appendix B are relatively more "technical" and they are primarily the responsibility of the ES developers (not ES users). After considering purposes (a) and (b) stated in Section 4.1 and that students with a non-IT background might feel uncomfortable or incompetent

working on these "technical" tasks (see Problem 3a (Group Heterogeneity) in Section 3), we advised students that they did not need to develop a "working" ES in the project. Thus, for example:

a) In Task 3.1, students only needed to design: (i) a series of input screens showing the dialogue questions (and their sequences) between a human user and the ES, and (ii) a series of output screens showing the possible advice or recommendations generated by the ES. Students could use any drawing software or tools (whether primitive or sophisticated) to design and produce these input and output screens. Students did not need to develop and implement the "background" programs or system routines to link all these screens together to make the system work.

b) The project assumed that a *rule-based* ES was to be developed with the support of an ES shell. An *ES shell* is a set of toolkits for developing ESs. The shell consists of some built-in ES components (e.g., an inference engine and a user interface) but with an *empty* knowledge base (Dunaway, 2022). Given a shell, in Task 4.1, students could focus on defining some production rules that were relevant to their chosen application domains, rather than working on "technical" tasks (e.g., implementing the inference logics) that were not related to the main learning outcomes of the project.

   In a rule-based ES, a production rule has the syntax: "IF <a condition or premise> THEN <an action or conclusion>" (Note that a rule can have multiple conditions or premises joined by the keywords AND, OR, or a combination of both). Hence, with some basic training, even students with a non-IT background should be able to define a set of production rules to capture the expert knowledge of the chosen application domains. Obviously, a "working" rule-based ES must be supported by a well-built knowledge base with a fairly comprehensive set of rules. Since students were not required to produce a "working" ES but rather to demonstrate their understanding of the key concepts of production rules, students needed only to define at least 30 production rules in the knowledge base.

Problem 3a (Group Heterogeneity) in Section 3 also mentions that few students with an IT background might prefer to address the more "technical" aspects of an ES in their projects. In view of this, any group was allowed to develop and implement a "working" ES (supported by a "comprehensive" knowledge base with more than 30 production rules) if the group wished to. Overall, by using the cut-over version of the ES Development Life Cycle for teaching and project execution, as well as making the development of a "working" ES and a "comprehensive" knowledge base optional, Problems 1 (Student Background), 2 (Topic Selection), 3a (Group Heterogeneity), and 3b (Project Emphasis) in Section 3 were largely alleviated.

5) An indispensable feature of a well-designed ES is the *explanation capability*, which is similar to the reasoning ability of human experts (Ye & Johnson, 1995). This explanation capability enables the user to ask the ES how a particular conclusion is reached and why a specific fact (or question) is needed (or asked). For example, the ES can generate a list of "fired" production rules to support a system recommendation. This capability has been shown to improve performance and aid the user with a better understanding of the application domain as well as generating a more positive user perception of the ES (Darlington, 2011). During their presentations, students needed to demonstrate the explanation capability of their ES by showing a list of "fired" rules used for deriving a system recommendation. These "fired" rules should be part of the minimum set of 30 production rules defined in the knowledge base.

4.3 Expected Issues to Address in Different Tasks of the ES Development Life Cycle

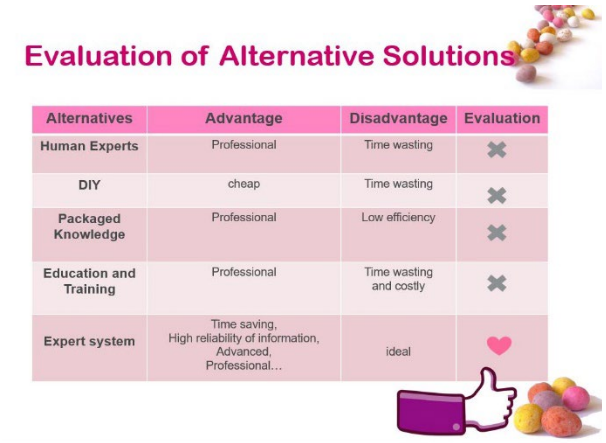
Although groups (particularly those comprised students with an IT background) were allowed to develop and implement a "working" ES, all students were reminded that they should spend sufficient effort on addressing the *managerial-related* (or *user-related*) issues in the ES Development Life Cycle (corresponding to LO2 in Section 4.2). This was because most of these students would enter the accounting profession after completing the MAcc degree. Thus, the most important thing they should learn from the project was how to participate in an ES project as a system user (not a developer). Here are some examples of the issues that students should address in the project:

Problem definition (Task 1.1 in Appendix B): Each group should identify the problems or real needs that motivated the development of an ES. For example, for an ES project related to obesity management (see Table 2), the group argued that developing this ES was worthwhile because of the time problem and the lack of sufficient expert knowledge. Without this ES, obese people must spend time and money to consult a doctor or to visit a weight control clinic.

Figure 1 shows the problem definition of a dream-home selection ES from another group (see Table 2):

Figure 1. Problem Definition of a Dream-Home Selection ES

Evaluation of alternative solutions (Task 1.3 in Appendix B): This task specifically addressed LO1 mentioned in Section 4.2. In some cases, even with a sound problem definition, developing an ES may not be the best solution. Other possible solutions to alleviate (or satisfy) the identified problems (or needs) are human-expert recruitment, education and training, packaged knowledge, conventional software, and public knowledge on the Internet. In view of this, each group should analyze and explain why developing an ES was the best solution. Figure 2 shows an example of this analysis for skin care treatment.



Alternatives	Advantage	Disadvantage	Evaluation
Human Experts	Professional	Time wasting	✖
DIY	cheap	Time wasting	✖
Packaged Knowledge	Professional	Low efficiency	✖
Education and Training	Professional	Time wasting and costly	✖
Expert system	Time saving, High reliability of information, Advanced, Professional…	ideal	♥

Figure 2. Evaluating Alternative Solutions for Skin Care Treatment

Verification of an ES approach (Task 1.4 in Appendix B): Same as Task 1.3 above, Task 1.4 also specifically addressed LO1 mentioned in Section 4.2. Developing an ES should only be considered when certain conditions are met, such as: (a) the task requires only cognitive skills, (b) the task mainly involves symbolic (not numeric) data processing, (c) nonoptimal results generated by the ES can be tolerated, and (d) there is at least one genuine human expert who is willing to cooperate, because human experts must be available to help the project team identify the expert knowledge to be captured as production rules in the knowledge base.

Building a small prototype (Task 3.1 in Appendix B) and completing the knowledge base (Task 4.1 in Appendix B): These tasks specifically addressed LO3 in Section 4.2. Recall that implementing a "working" ES was not a mandatory requirement for the project. Nevertheless, each group still had to design some input and output screens to illustrate the flow of dialogue between an ES and the human user. Also, at least 30 production rules should be defined. Figure 3 shows two sample input screens of a housing rental ES (see Table 2). In each screen, the production rule associated with the question is also shown.

User acceptance (Task 5.1 in Appendix B): Same as Tasks 3.1 and 4.1 above, Task 5.1 also specifically addressed LO3 mentioned in Section 4.2. It would be a project failure if the user did not accept the developed ES. Thus, each group had to explain what measures were taken to increase user acceptance. Figure 4 shows the measures taken to improve user acceptance of a dream-home selection ES.

4.4 Project Assessment

Students were reminded that each group member needed to make sufficient contribution to the project. Project assessment was made in accordance with the criteria listed in Appendix D. Every student in a given group received the same mark except that, when a student did not contribute to or participate in the project satisfactorily, then their mark would be downward adjusted accordingly (which did not happen in my last teaching).

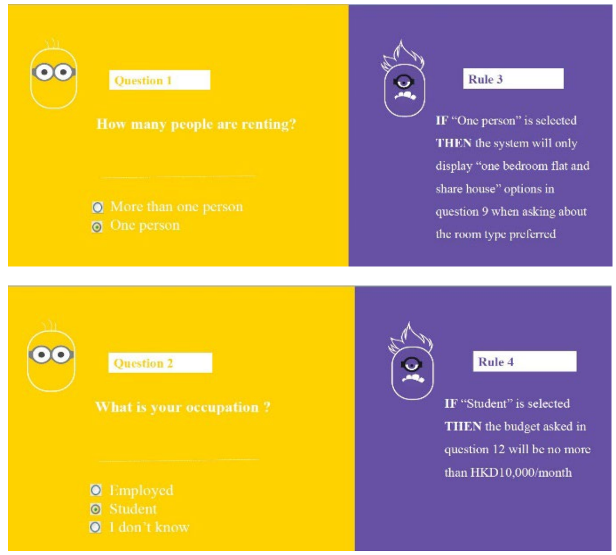


Figure 3. Sample Input Screens and Associated Production Rules for a Housing Rental ES

Figure 4. Measures to Improve User Acceptance of a Dream-Home Selection ES

For the nine project groups, their scores (out of a total of 25; in ascending order) were 16, 17, 18, 18, 19, 20, 20, 21, 22, respectively. The mean score was 19.0 (equivalent to 76.0 out of 100) and the standard deviation was 1.83. In the Accounting School of UNIV, a mean score of 76.0 (out of 100) was considered very satisfactory.

Near the end of the semester, an anonymous student feedback evaluation exercise was performed. Overall, the feedback evaluation score of CI-IT was 4.4 (out of 5.0; higher the better), indicating that students were very satisfied with the learning journey and outcomes. Note that this feedback evaluation score was for the whole CI-IT course, in which ES was only one component. Table 3 shows that the students' open comments specific to the ES component, which were extracted from the student feedback evaluation exercise.

Students' Comments
• The ES project was very interesting.
• The ES group project was well designed.
• I learnt a lot from the ES group project.
• Throughout the [ES] group project, I learnt a lot.
• Thank you for making the uninteresting ES topic so interesting.
• As a student without an IT background, in the beginning, I was worrying that I might not be able to understand the ES concepts. After the group project, I found that I was able to grasp the overall idea without difficulty.
• Students were encouraged to do an interesting [ES] project, which was helpful for students to understand how an ES works in reality.

Table 3. Students' Comments on the ES Component

5. ADAPTING OUR TEACHING AND PROJECT APPROACH

Section 4 has discussed our approach to teaching the ES component in CI-IT, supported by a corresponding group project. The effectiveness of the approach was also confirmed by a follow-up student feedback evaluation exercise, particularly its students' open comments. It should be kept in mind, however, that the approach was designed in accordance with the specific teaching environment and constraints associated with MAcc and CI-IT at UNIV. When adapting our approach to other environments, the following recommendations should be considered:

- At UNIV, many students with a non-IT background were admitted to MAcc and CI-IT was offered in Semester 1 of Year 1. Thus, at the time when students undertook the ES project, most of them had not learnt sufficient accounting/finance knowledge to enable them to choose this application domain for their projects. To alleviate this problem and to inspire their interests and engagement, students were allowed to choose any domain (even outside accounting/finance) for their projects. However, if teaching ES occurs in a later semester or year, the instructor may consider the appropriateness of restricting the ES project to the accounting/finance domain only.
- In CI-IT, only six hours were allocated to teach ES. Thus, a cut-down version of Turban's ES Development Life Cycle (see Appendix B) was used as a framework to guide teaching. If, however, the teaching hours for ES are longer, the instructor may consider reverting to the "original" Turban's ES Development Life Cycle and spending more time to discuss each phase or task of the development life cycle in more detail.
- If more time is allowed for teaching ES, students should be introduced to a range of ES applications in different accounting-related or finance-related areas such as financial accounting, management accounting, cost accounting, auditing, taxation, financial statement analysis, financial planning, and financial analysis (Tomás, 1998). Such teaching should preferably be supported by a hands-on lab session.

6. SUMMARY AND CONCLUSION

Due to the popularity of ES in many sectors (including accounting and finance) of our society, accounting students should be taught with ES. The key question is how to teach ES to them to achieve good learning outcomes, which is easier said than done. Teaching the concepts of ES to non-technical accounting students is often problematic. Based on our teaching experience at UNIV, we find that this task is challenging but still achievable, provided that thorough and thoughtful planning is undertaken in designing the teaching approach and the supplementary hands-on project exercise.

In this article, we have highlighted some challenges and problems of teaching ES to accounting students at the master's level at UNIV. We have then discussed in detail how to: (a) teach ES development using a cut-down version of Turban's ES Development Life Cycle (Turban et al., 2011), (b) design a group project to reinforce students' learning, and (c) assess the outcome of the project. A follow-up student feedback evaluation exercise has confirmed the effectiveness of our teaching approach and that students were able to comprehend the technical ES concepts with good learning outcomes.

7. REFERENCES

Abouloula, K., Ali, O.-Y., & Salah-Ddine, K. (2019). Pattern to Build a Robust Trend Indicator for Automated Trading. In N. Metawa, M. Elhoseny, A. E. Hassanien, & M. K. Hassan (Eds.), *Expert Systems in Finance: Smart Financial Applications in Big Data Environments* (pp. 217-235). Taylor & Francis Group.

Al Janabi, M. A. M. (2019). Theoretical & Practical Foundations of Liquidity-Adjusted Value-at-Risk (LVaR). In N. Metawa, M. Elhoseny, A. E. Hassanien, & M. K. Hassan (Eds.), *Expert Systems in Finance: Smart Financial Applications in Big Data Environments* (pp. 1-31). Taylor & Francis Group.

Baldwin, A. A., Brown, C. E., & Trinkle, B. S. (2006). Opportunities for Artificial Intelligence Development in the Accounting Domain: The Case for Auditing. *Intelligent Systems in Accounting, Finance & Management*, 14, 77-86.

Bresnahan, T., & Yin, P.-L. (2017). Adoption of New Information & Communications Technologies in the Workplace Today. In J. Lerner & S. Stern (Eds.), *Innovation Policy & the Economy* (pp. 95-124). Vol. 17. National Bureau of Economic Research.

Brown, C. E., Baldwin-Morgan, A. A., & Sangster, A. (1995). Expert Systems in Accounting Education: A Literature Guide. *Accounting Education*, 4(3), 283-296.

Changchit, C., & Holsapple, C. W. (2004). The Development of an Expert System for Managerial Evaluation of Internal Controls. *Intelligent Systems in Accounting, Finance & Management*, 12, 103-120.

Darlington, K. W. (2011). Designing for Explanation in Health Care Applications of Expert Systems. *SAGE Open*, 1(1). https://journals.sagepub.com/doi/10.1177/2158244011408618

Dunaway, J. (2022, April 22). *What Are Expert System Shells?* EasyTechJunkie. https://www.easytechjunkie.com/what-are-expert-system-shells.htm

Ellis, D. (2021). *The State of Project-Based Learning: Engaging & Empowering Our Students for Future Success.* PBL Works. https://www.pblworks.org/sites/default/files/2021-07/The%20State%20of%20Project%20Based%20Learning.pdf

Guo, P., Saab, N., Post, L. S., & Admiraal, W. (2020). A Review of Project-Based Learning in Higher Education: Student Outcomes & Measures. *International Journal of Educational Research*, 102, Article 101586.

Jackson, R. B., & Cherrington, J. O. (2001). IT Instruction Methodology & Minimum Competency for Accounting Students. *Journal of Information Systems Education*, 12(4), 213-221.

Lymer, A. (1994). The Integration of Expert Systems into the Teaching of Accounting: A Third-Year Option Course Approach. *Accounting Education*, 4(3), 249-258.

Miller, E. C., & Krajcik, J. S. (2019). Promoting Deep Learning Through Project-Based Learning: A Design Problem. *Disciplinary & Interdisciplinary Science Education Research*, 1, Article 7.

Pan, G., & Seow, P.-S. (2016). Preparing Accounting Graduates for Digital Revolution: A Critical Review of Information Technology Competencies & Skills Development. *Journal of Education for Business*, 91(3), 166-175.

Sangster, A. (1991). Expert Systems in the Accounting Curriculum: A Textbook Review. *British Accounting Review*, 23(2), 171-178.

Sommerville, I. (2015). *Software Engineering* (10th ed.). Upper Saddle River, NJ, USA: Pearson.

Stancheva-Todorova, E., & Bogdanova, B. (2021). Enhancing Investors' Decision-making: An Interdisciplinary AI-based Case Study for Accounting Students. In *AIP Conference Proceedings*, 2333, 150007. https://aip.scitation.org/doi/pdf/10.1063/5.0041612

Thomas, J. W. (2000). *A Review of Research on Project-Based Learning.* https://www.asec.purdue.edu/lct/HBCU/documents/AReviewofResearchofProject-BasedLearning.pdf

Tomás, A. S. (1998). Expert Systems Applications in Accounting. In M. A. Vasarhelyi & A. Kogan (Eds.) *Artificial Intelligence in Accounting & Auditing: Towards New Paradigms* (pp. 11-27). Vol. 4. Marcus Wiener.

Turban, E., Aronson, J. E., Liang, T.-P., & Sharda, R. (2011). *Decision Support & Business Intelligence Systems* (9th ed.). New Delhi, India: Pearson.

Wagner, W., Otto, J., & Chung, Q. B. (2002). Knowledge Acquisition for Expert Systems in Accounting & Financial Problem Domains. *Knowledge-Based Systems*, 15(8), 439-447.

Ye, L. R., & Johnson, P. E. (1995). The Impact of Explanation Facilities on User Acceptance of Expert Systems Advice. *MIS Quarterly*, 19(2), 157-172.

AUTHOR BIOGRAPHY

Pak-Lok Poon received his Ph.D. degree in software engineering from The University of Melbourne, Australia. He is an associate professor with the School of Engineering and Technology, Central Queensland University, Australia. His research interests include software testing, requirements engineering and inspection, electronic commerce, and information systems education. He was a guest editor of two special issues of the *Journal of Systems and Software* in 2018 and 2021, respectively. He is also on the editorial board of the *Journal of Organizational and End User Computing*, *Journal of Information Systems Education*, and *International Journal of Information Systems and Supply Chain Management*.

APPENDICES

Appendix A. The Original Version of ES Development Life Cycle (Turban et al., 2011)

Major Phases	Associated Tasks
1. Project initialization	1.1 Problem definition
	1.2 Need assessment
	1.3 Evaluation of alternative solutions
	1.4 Verification of an ES approach
	1.5 Consideration of managerial issues
2. System analysis & design	2.1 Conceptual design & plan
	2.2 Development strategy
	2.3 Sources of knowledge
	2.4 Computing resources
	2.5 Feasibility study
	2.6 Cost-benefit analysis
3. Rapid prototyping	3.1 Building a small prototype
	3.2 Testing, improving, & expanding the prototype
	3.3 Demonstrating & analyzing feasibility
	3.4 Completing design
4. System development	4.1 Completing the knowledge base
	4.2 Testing, evaluation, & improving the knowledge base
	4.3 Planning for integration
5. Implementation	5.1 User acceptance
	5.2 Installation, demonstration, & deployment
	5.3 Orientation & user training
	5.4 Security
	5.5 Documentation
	5.6 Integration & field testing
6. Postimplementation	6.1 Operation
	6.2 Maintenance & upgrades
	6.3 Periodic evaluation

Notes:

- Tasks 1.5, 2.4, 2.6, 3.2, 3.4, 4.3, 5.5, & 5.6 above do not appear in the cut-down version of the ES Development Life Cycle in Appendix B.
- In the above table, Task 2.6 (Cost-Benefit Analysis) is merged with Task 2.5 (Feasibility Study) to form Task 2.4 (Feasibility Study) in Appendix B, because cost-benefit analysis is in fact an economic feasibility study (i.e., one of the various types of feasibility study).

Appendix B. A Cut-Down Version of ES Development Life Cycle (Turban et al., 2011)

Major Phases	Associated Tasks	Technical-related	Managerial/user-related
1. Project initialization	1.1 Problem definition		√
	1.2 Need assessment		√
	1.3 Evaluation of alternative solutions		√
	1.4 Verification of an ES approach	√	√
2. System analysis & design	2.1 Conceptual design & plan	√	
	2.2 Development strategy	√	
	2.3 Sources of knowledge	√	√
	2.4 Feasibility study	√	√
3. Rapid prototyping	3.1 Building a small prototype	√	
	3.2 Demonstrating & analyzing feasibility	√	
4. System development	4.1 Completing the knowledge base	√	
	4.2 Testing, evaluation, & improving the knowledge base	√	
5. Implementation	5.1 User acceptance		√
	5.2 Installation, demonstration, & deployment	√	
	5.3 Orientation & user training	√	√
	5.4 Security	√	
6. Postimplementation	6.1 Operation		√
	6.2 Maintenance & upgrades	√	
	6.3 Periodic evaluation	√	√

Appendix C. ES Group Project Handout

This group project has the following three specific learning outcomes:
- (LO1) To understand the situations under which an ES (rather than a "conventional" non-ES system) should be developed.
- (LO2) To understand the managerial issues associated with an ES development project.
- (LO3) To understand the design/implementation issues and concepts associated with an ES development project.

This project contributes to 25% of the total course assessment. It should be worked by a group of four to five students. You are allowed to form groups of your own choice. Every student in the same group will receive the same mark, except that when a student does not contribute to or participate in the project satisfactorily, then the mark of this student in concern will be downward adjusted accordingly.

Group presentations will spread across Weeks 10 and 11. For fairness, regardless of presentation week, every group must submit the presentation slides by the end of Week 9. Once a submission is made, further changes to the presentation slides are not allowed. For the presentation, each group nominates two to three members to make a 40-minute presentation in the class on behalf of the group. After the presentation, there will be a 15-20 minute discussion session, chaired by the presentation group.

Notes:
- Each group can freely choose the project's application domain, even if it is outside the accounting/finance area.
- The project should largely follow the cut-down version of Turban's ES Development Life Cycle taught in the lecture. Your group should spend sufficient effort on addressing the managerial-related (or user-related) issues in the ES Development Life Cycle.
- This project assumes that the ES is rule-based and is developed with the support of an ES shell. Accordingly, your group should focus on defining some production rules that are relevant to the chosen application domain, rather than sidetracking the effort to work on the technical tasks of implementing the inference logics.
- Developing a "working" ES is optional. Thus, it is not a mandatory requirement that the knowledge base must contain a large set of rules. However, a minimum of 30 production rules should be defined in the knowledge base to show your group's understanding of the concept.
- A series of input and output screens showing the interaction between the ES and human users is required. You can use any drawing software or tools to design and produce these screens.
- Your presentation should demonstrate the explanation capability of the ES by showing a list of "fired" rules used for deriving a system recommendation. These "fired" rules should be part of the minimum set of 30 production rules defined in the knowledge base.

Appendix D. Assessment Scheme of the ES Project

Score (out of 25%)	Criteria and Standards
22–25	• Present arguments that have an element of originality • Demonstrate a strong understanding of all relevant knowledge • Organize the presentation in a professional manner • Handle questions professionally • Demonstrate a strong coordination among group members • Well-prepared PowerPoint slides
18–21	• Present arguments that go beyond the lecture & textbook • Demonstrate a good understanding of all relevant knowledge • Organize the presentation in a structured & logical manner • Handle questions in a logical way • Demonstrate a good coordination among group members • Well-prepared PowerPoint slides
14–17	• Present arguments in a fair manner • Display a basic understanding of the concepts involved • Organize the presentation with a clear structure • Handling some questions • Demonstrate some coordination among group members • Acceptable PowerPoint slides
10–13	• Present arguments that are barely sound • Display a minimum understanding of the concepts involved • Organize the presentation in a marginally acceptable manner • Unable to handle questions • Demonstrate a weak coordination among group members • Marginally acceptable PowerPoint slides
0–9	• Fail to present arguments • Display a poor understanding of the concepts involved • Organize the presentation poorly • Unable & unwilling to handle questions • Demonstrate a weak coordination among group members • Poorly prepared PowerPoint slides

Information Systems & Computing Academic Professionals

Education Special Interest Group

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN: 2574-3872 (Online) 1055-3096 (Print)