Preprint of article published in Journal of Informatics – Education and Research © IAIM Press 2002.

Teaching IS with Simulations: Issues and Experiences

John Grundy

Department of Computer Science, University of Auckland
Private Bag 92019, Auckland, New Zealand

Phone: +64-9-3737-599 ext 8761
Fax: +64-9-3737-453
Email: john-g@cs.auckland.ac.nz

ABSTRACT

Topics of Information Systems lectures and tutorials are dynamic systems, such as business models and development processes, requirements specification formulation, and system architectures and designs. Using simulations in lectures, tutorials and for distance learning can significantly increase students' understanding of topics, teamwork skills and enjoyment of their learning experience. The concept of simulations and some of the ways they can be used during teaching Information Systems are explained and illustrated. Specific issues when using simulations during tutorials, lectures and for distance learning are discussed. One detailed and several brief case study examples of using simulation from the author's teaching experiences are described.

KEYWORDS

IS Education (IA), IS Curriculum (IA01), Computer Science Education (IA03),

INTRODUCTION

There are a variety of ways of presenting information during lectures and tutorials to Information Systems, Software Engineering and Computer Science students:

- Factual descriptions of concepts and technology characteristics
- Use of case study descriptions with examples of concepts and technologies in use
- Worked examples
- Simulations

This paper focuses on the use of simulations. A simulation is where a business model, requirements gathering, architecture and design, or a system technology or application is actively simulated during a lecture, tutorial or during distance learning. An E-commerce business model, for example, might be simulated by students playing the roles of customer, supplier, vendor and bank thus simulating inter-organisational information exchanges. A client-server system architecture might be simulated by students exchanging messages between architecture components. A software application might be simulated on paper by “running” its user interfaces by hand.

The author’s teaching experiences have shown simulations to be very effective teaching tools. They make IS concepts and technologies concrete for students helping them learn about these by playing the part of elements of models, specifications and implementations. They encourage teamwork as students must co-operate to play different roles in a simulation. A key emphasis is on learning in a collegial, mutually-supportive manner. Well-designed simulations are both informative and enjoyable for students and IS faculty. Designing and partaking in simulations helps foster staff-student communication and understanding.

In the following section an outline of the author’s views on using simulations as teaching tools for IS is given along with a summary of some of the key general issues in using this approach for both IS faculty and students. This includes a review of the education literature pertaining to the use of simulations in various disciplines. Particular issues for using simulations in small tutorials, large lectures and distance learning scenarios are addressed in the following sections. In each of these sections some example simulation exercises are described along with the author’s experiences using these simulations in different settings. The paper concludes with a discussion of key issues for IS faculty and students of adopting some simulation-based learning based on the author’s experiences.

SIMULATION AS A TEACHING INSTRUMENT

Simulation is where an IS concept, technology or application is worked through using a concrete example and participants actively play roles in the process. Students and faculty simulate how a business model, development process, information technology or computer application works. Some generic examples are illustrated in Figure 1. Students may play different organisation roles when simulating the interactions in a business model. Typically a case study will be used to provide organisational information to use in the simulation. Students may play roles in the simulated execution of a system architecture, design or software application. Example scenarios of inter-object communication or data input and output are used to drive such execution-based simulations. Students may play roles of IS development team members, simulating the process of system development. Students may also play the roles of supporting documents, events and messages, IS infrastructure components or CASE tools, aiding others during a simulation exercise. The instructor may also play a role themselves, or help co-ordinate a simulation. A whole class may participate, selected members of the class, or a class may be broken into small, con-currently simulating groups. In distance education, simulations may be pre-packaged for “replay”, a live, on-line simulation may be interacted with, or Computer-Supported Co-operative Learning technologies may be used to support synchronous, distributed simulations by dispersed students (this includes the use of on-line slide shows, real-time audio and video, and text-based chat and email).

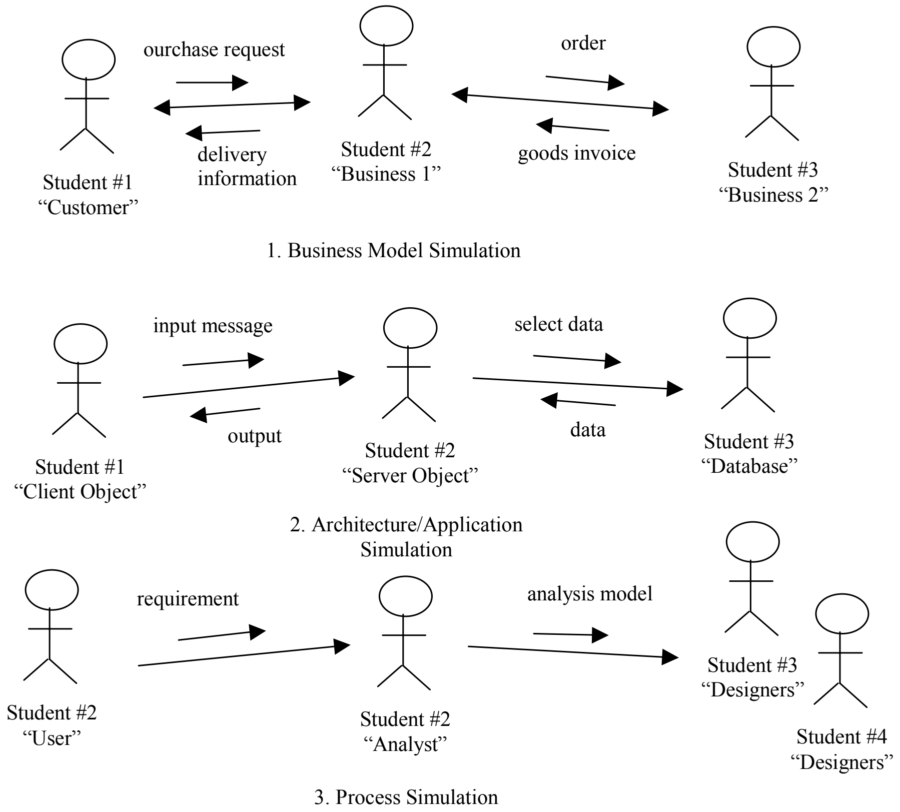


Figure 1. Some models of simulations in IS Education.

Simulation-based learning can be thought of as a subset of the more-familiar "project-based learning" or "problem-based learning" – learning-by-doing (Santoro et al, 2000; Land et al, 2000; Cocker et al 1999; Hasman and Boshuizen, 2001). Project-based learning involves students working as groups or teams tackling realistic problems. It has become popular in many disciplines, including medicine, engineering and IS education. Simulations are discrete problem-solving exercises that typically are fitted within lecture or tutorial constraints to allow students to understand concepts, processes, techniques or technologies in an abstract yet hands-on manner. Students work through a simplified, concrete example of a conceptual, technical or process-based problem to gain an understanding of key issues.

The use of simulation is common in a number of disciplines, particularly Engineering, Computer Science and Medicine (Calazans and Moraes, 2001;Shah and Darzi, 2001; del Rio et al 2001). The use and purpose of simulation exercises in these fields is quite different, but all share a common aim of illuminating complex (or sometimes simple) things by acting-out usually simplified problem scenarios. The use of props during simulation exercises differs markedly depending on the discipline and focus of the work. Engineers may use pen and paper, but often things to represent parts of machinery, power plants, electronic devices and buildings. Medical students may simulate body functions or treatment processes using pseudo-realistic props. Computer Scientists may use real computer componentry or abstract representations of computer parts, data structures, algorithms and designs.

Computer-supported co-operative learning (CSCL) environments and multi-media and hyper media technologies would seem to offer great potential for conducting simulations during distance learning or may help provide a richer set of simulation experiences and documentation aides in a tutorial setting (Newman et al 1997; Despres and George, 2001). However, while these technologies allow for digital presentation and packaging of simulation exercises, it is unclear to us just how effective these might prove in practice. The simulation examples described in this paper do not depend on any particular information technology for delivery (except the ones for distance learning) but it is possible to incorporate additional technology aides, if available, in addition to the ones described.

Some key issues when using simulation-based teaching and learning include:

- How the simulation will be organised in order to make use of previously presented conceptual or practical material
- The organisation of the students and co-ordination of the simulation (for example, the whole class participating, part of class being watched by the rest, or several con-currently simulating groups)
- Involvement of the instructor (either active co-ordinator, participant, passive observer or discussion facilitator after simulation finished)
- Identifying good, realistic information to use to "drive" the simulation (not overly simplistic nor too detailed/complex)
- Timeframe for running simulation (for example, a few minutes, a whole lecture or tutorial, spread across multiple tutorials, or student-determined if distance learning delivery model)
- Use of "props" to make the simulation more tangible and interesting (for example, pieces of paper, example business information scenario or design model diagrams on paper, balloons, envelopes, boxes, string or pens)
- Manner in which the results are recorded (for example, done after completion, done during simulation by each participant, done during simulation by one student or by the instructor, done using paper, a whiteboard or via CSCL tools)

The following sections discuss the use of simulation exercises during tutorials, lectures and distance learning, drawing on the author's experiences in using diverse IS simulations in these settings. One large case study is presented to give the reader a detailed illustration of planning, carrying out and reviewing results of a simulation exercise. Several other simulation examples are briefly described and reviewed to illustrate the diversity of approaches and examples available to the instructor.

SIMULATION IN TUTORIALS

A tutorial setting provides an almost ideal venue for the use of simulations to aid student understanding and learning of theoretical concepts and practical techniques. However, not all tutorials are organised in the same manner, comprise a homogeneous student population nor are hosted in a venue suitable to all types of simulation-based learning.

Some key issues when using simulation in small group tutorial situations are summarised below:

- *Nature of the group*. Typically tutorials comprise relatively small numbers of students in most institutions, usually between 5 and 30 participants. This small size is crucial, as it allows group members to more easily interact, encourages a more free-format discussion between tutorial members, and the group can be hosted in tutorial rooms which are flat and often with moveable desks and chairs.

- *Student involvement*. Most tutorials can be run to easily involve all students. The instructor can usually explicitly ensure each participant makes some contribution, though some students may not be keen to join in due to personal preferences (reserved, lacking self-confidence) or cultural issues.
- *Sharing experiences*. Due to their small size and more "intimate" nature than lectures, it is usually possible and very desirable for students in a tutorial to share experiences during and after conducting a simulation.
- *Venue characteristics*. A simulation run using a typical tutorial room that is small in size with non-fixed chairs and desks can use all of tutorial room, including setting up the room in an "open-plan" style. This can greatly enhance the ability of students to participate in a simulation-based exercise, and often the room characteristics and furniture can be incorporated into the simulation design, for example to influence how the simulation is done.

Below are two case studies of using simulations in a tutorial setting. The first, a groupware system simulation exercise, is described in detail to give the reader a full simulation example for reference.

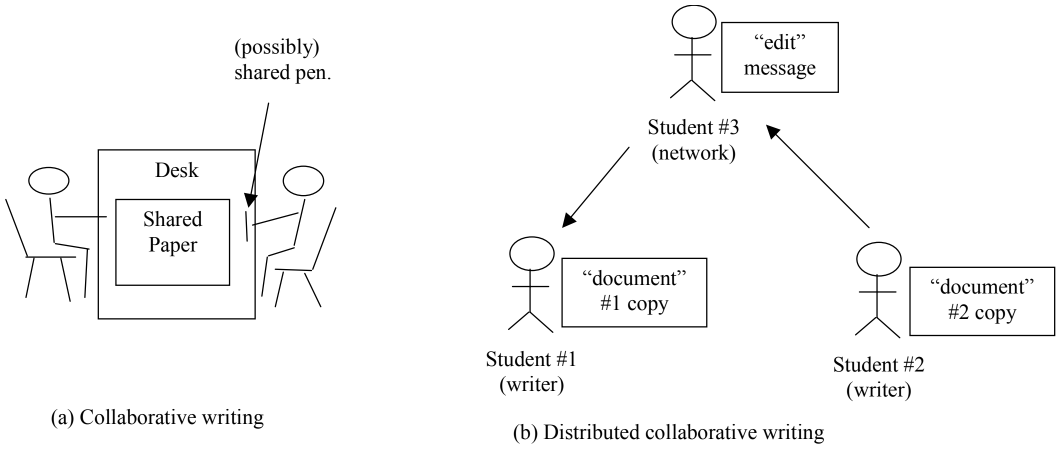
Synchronous Groupware

Groupware provides collaborative work tools for organisational workers (Ellis et al, 1991; Drummond et al, 2001). Groupware is complex, from user, architectural and organisational perspectives. In order to better understand the way groupware works and can be used, several different simulation exercises can be used. A simulation is described below that aims to illustrate key user interaction issues with groupware systems and how key groupware architectural characteristics need to work. The first part of the simulation provides a user view of interaction with groupware applications, the second provides a groupware system architecture perspective for students.

This groupware simulation was designed and run for use within a graduate Human-Computer Interaction course. In this example students work in several small groups of between 3 and 4 people, ideally using an open-plan tutorial room with moveable desks and chairs, allowing students to easily gather around small desks and move about the room. Each group does the same simulation exercise and independently document their progress and write up their results. The props used to aide the simulation exercise include pieces of blank paper, pens and envelopes. The students must co-author a simple document (usually from part of the lecture notes or course textbook) and understand a fictitious groupware tool used to do this from both a user and a system perspective.

In the first part of the exercise the students try and co-author their simple document. Initially one student writes first then stops to let the other student write (this situation is called "turn-taking" in groupware systems, used to avoid con-current editing clashes). One student keeps writing until they are interrupted by the other who then asks to be allowed to write (called "floor control" in groupware systems). A shared pen enforces this turn-taking protocol. Later, both authors will try and write at the same time using different pens (this is called "shared information spaces"). The students must work out a mechanism to avoid over-writing each other's work. Figure 2(a) illustrates this scenario.

In the second part of the simulation the document authors are then separated across the room. Other group members are used to mimic network communications by taking "messages" written on pieces of paper and carrying these in envelopes from one author's groupware client to the other author's client application, which has to incorporate the information into the other copy of the "document". One author can also send a whole document with changes needing to be "merged" by the other author. Figure 2(b) illustrates this scenario. The instructor can make things more challenging by interrupting message flow (simulating network down-time), scrambling the order messages are received (simulating asynchronous message arrival) and having message responses "lost" (simulating lossy networking and requiring re-synchronisation of the shared documents).



(a) Collaborative writing

(b) Distributed collaborative writing

Figure 2. Groupware simulation examples.

Students need to document their simulations. This includes writing down the steps in each simulation done, what happened at each step, what information was exchanged, and any problems that occurred (and how they solved them). Depending on time limitations, these results can be discussed by the tutorial class at various points, at the end of the class or in a subsequent tutorial or lecture. When running this simulation, students come up with some interesting solutions: a third student co-ordinates pen sharing and document "locking" (just like real groupware applications often support automatically); students allocate areas for each student to write in (once including vertical blocks on a page where students write half a line each to be completed by the other student!); and students devise realistic messaging protocols between groupware clients.

Many illuminating things about groupware are discovered by students during this sumulation. The issue of group awareness, co-ordinating tasks, allocating tasks and so on are all manifested, and students must develop both a "social" and "technical" protocol to handle these. Technical issues of "bottle-neck" (students acting as the network running into each other) slow down "response" time. Organisational issues can be highlighted by the instructor pointing out the issues of deploying groupware, training users, and noting problems with information privacy. After doing this simulation students can use commercial groupware tools or build prototype groupware applications, having a sound knowledge of both user interaction and technical issues.

E-commerce Business Model

Another example simulation the author has used extensively is the working of an E-commerce-style business model. This has been used in tutorials for both a second year undergraduate Object-oriented Systems course and a graduate E-commerce System Engineering course. Groups of between 4 and 6 students in a tutorial act as customer, vendor, supplier, banker, and sometimes as network infrastructure. The simulation can be run in open-plan rooms but has also been successful in rooms with fixed seating, sometimes by having one group "act out" the simulation while the rest of class observes or even directs it with the instructors help. Figure 3 illustrates this simulation scenario.

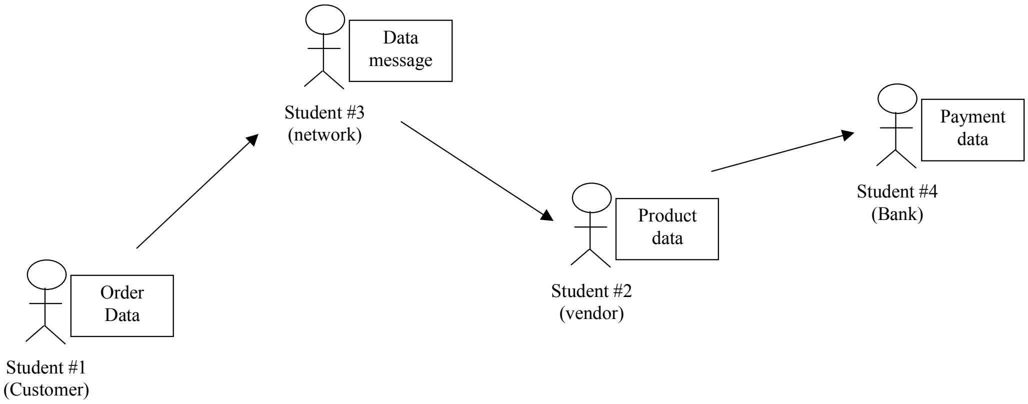


Figure 3. E-commerce system simulation example.

In this tutorial simulation students act out information exchanges between people and the systems provided by vendor, supplier and bank. The behaviour of a "traditional" commerce model where people interact and computers are more-or-less isolated and used by staff only is often run first, to highlight both similarities and differences between traditional and E-commerce models (Bambury, 1998; Bolin, 1998). A variety of business services can be simulated in this way, including customer purchase, supplier order placement, and inventory management. Different E-commerce models, such as traditional or on-line retailer, virtual organisation or information brokering organisation can be investigated, as can different payment methods, such as traditional cheque banking, electronic credit card authorisation and digital money.

As an example, a student (customer) searches for products on-line and pays by credit card. Other students simulate the various interactions between customer and vendor to do searching, vendor and bank to do payment; and vendor and supplier for just-in-time manufacturing business models. Props that can be used to illuminate the simulation include messages in envelopes for business data exchange, balloons for products or messaging, or even candy for a more rewarding treat for a successful on-line purchase.

SIMULATION IN LECTURES

While the (typically) small class sizes, often non-fixed, flat seating, and intimate nature of tutorials is ideal for many simulation-based learning exercises, traditional lecture settings can also be used effectively. Generally greater planning and care running simulations is needed, in the author's experience, as there is usually less room for error than in a tutorial. When running a simulation exercise, typical things that can go wrong include students missing out crucial steps, losing or mis-calculating important data or messages, or students making up their own additional steps, often confusing them in the process. In a tutorial such mistakes can generally be observed by the instructor and corrected, or the instructor at least be able to help the group recover to some degree from them. In a lecture setting, particularly if small grup exercises are used, this is virtually impossible to do.

In this section the term "lecture" is used to refer to a situation where almost all speaking is done by the instructor, a large (usually 50 or more) group of students are present, and seating is usually in a fixed, multi-tier arrangement.

Some key issues the author has encountered when using simulation in medium-to-large lecture situations are summarised below:

- *Large class size*. The author has run simulation-based activities in lectures with anything between 50 to over 300 students present. This is a huge difference to the 5 to 30 in typical tutorial classes and greatly affects what can and can't be done.
- *Seatting*. Most lecture rooms come with fixed desks and seating, multiple tiers, and safety requirements dictate that students can not move around much (if at all). Usually a simulation must be done with a small selection of the class at the front of the lecture room, or require brief pen-and-paper work by 1 to 3 students.
- *Student Participation.* Involving a whole class requires a simulation be done by individuals or at most groups of 3 students all sitting next to each other. Its obviously extremely difficult for the instructor to see what even a small number of groups are doing, and if they go wrong impossible to correct. An alternative which sacrifices each student's direct involvement in the simulation but enables the instructor to more easily control and explain it is to have a small number of students come the front of the room and act out the simulation while the rest of the class observes and give directions or suggestions.
- *Simulation review*. In the lecture setting, the instructor generally must review the simulation afterwards to the whole class, compared to a more usual group discussion of a simulation in a tutorial. Homework exercises can be set for students in a tutorial setting that build on a simulation and later reviewed/discussed, but this has been found this much less feasible and effective in the lecture environment

Two case studies of the author's experiences using two very different kinds of simulation in lectures are presented below. It is certainly possible to run both of these in tutorials (and this has been done by the author), but the nature of the simulation exercise done is quite different.

Software Architecture

Software architectures describe the ways developers organise the software components of their systems (Bass et al, 1998). The author has used a software architecture simulation successfully in a final year undergraduate Distributed Object and Algorithms course with over 200 students in the lecture theatre, as well as a much smaller graduate Information Systems engineering course. The lecturer brings along some props that are used to assist in running the simulation. These have included boxes (to hold "data"), balloons (to act as inter-process "messages"), large cards (to identify students acting as "system components"), and string (used to mimic communication network links. Students act as various architectural abstractions that must exchange messages and data in order to "run" a simple system. Usually an e-commerce system example scenario is used, as many of the students have used such a system themselves. A small group of students are chosen or asked to volunteer by the instructor and are organised at the front of the class as a client, server and database and sometimes as a database table, a remote object or an end user. Balloons are used to represent messages or data items flowing between architecture components. Boxes represent database tables or message queues holding data values. Multiple people can be used as clients to simulate concurrent server or database access. Figure 4 illustrates this simulation scenario.

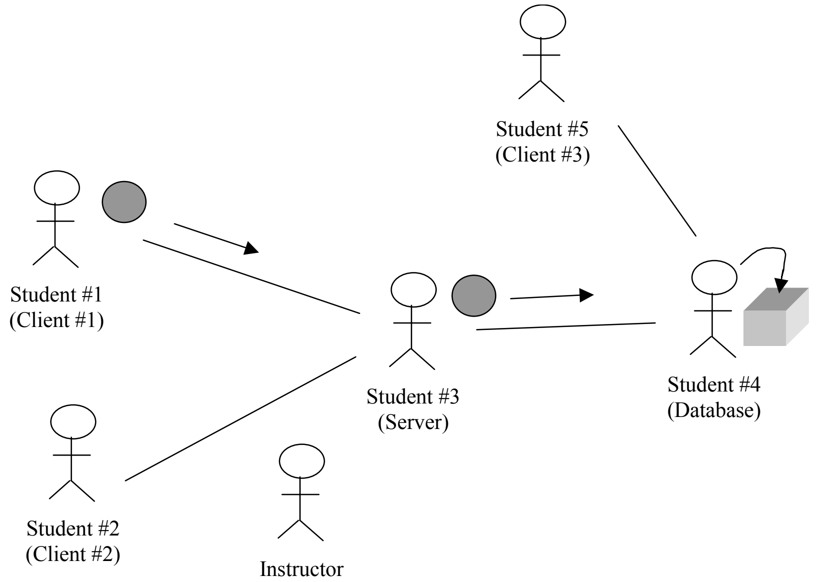


Figure 4. Software architecture simulation example.

The "clients" send requests (represented by balloons) to the "server". The server processes these and sends requests to the "database". Typically the instructor talks through the simulation, sometimes with the input of the rest of the observing class. Network connections can be "broken" or servers/database "over-loaded" to illustrate problems that can occur and what happens. Similar simulations have been done by the author for object-oriented implementations (using socket and database communications) in a second year undergraduate Object-oriented Systems course; in a final year undergraduate Information Systems Project course; and in a final year undergraduate Software Engineering project course. Such simulations work well (in general) and the class gets a good "feel" for key architecture and implementation issues, even with complex systems. Alternative architectures can be simulated and results compared and contrasted.

Round-trip Engineering with CASE Tools

Some Computer-Aided Software Engineering (CASE) tools provide "round-trip engineering" facilities. This is where the tool generates code and can reverse engineer design changes from code changes (Meyer, 1998; Smith, 2001), keeping changes to the design and changes to the code synchronised. The author has used a simple round-trip engineering simulation scenario in a graduate Information Systems Engineering course and in a final year undergraduate Information Systems Project course. This simulation can be done by pairs of students in a lecture at their desks or with several students in front of the class. With the pair model, one person is the CASE tool, one a programming environment. A rapid applications development example is simulated by having the "CASE tool" used to build (very) simple design, and generate (very) simple code to give to the "programming tool". The programming tool then adds detailed code to this generated code skeleton, including statements, new attributes, methods and classes. The CASE tool is then given the changed code to "parse" and extract design changes. A key element of this simulation is the recognition that the CASE tool and programming tool each hold information "lost" between the tools (such as design rationale and detailed method code).

Simulations involving a whole class doing the exercise differ from the previous example in that individual or pair results may wildly differ, and its impossible to directly review every ones progress. Our experience has shown that very simple whole-class simulations are feasible in a lecture (large or small) environment, but anything involving moderately complex concepts of processes needs to be done with the instructor controlling the simulation with a small group in front of the lecture.

SIMULATION IN DISTANCE LEARNING

This section briefly reports on the author's experiences using a distance education model with which to run simulations. Some key issues when using simulation in distance learning situations include:

- *On-line materials*. Lecture notes, tutorial notes and readings. may be used to complement lectures, tutorials or be the entire basis of course work. Students are used to making use of these materials in their own time and directing their own learning.
- *Off-line Simulations*. Unless very sophisticated multi-media technology is available to all participants of a class, simulations generally need to be designed and packaged "off-line". The author has found a good way to do this is to package them as part of tutorials or hyper-media animations, for example Powerpoint™ slides or Macromedia™ presentations. Simulations can also be implemented using web-based facilities (web pages or applets). Another form of packaging is to video a simulation done in a tutorial or lecture and make this available on-line for students to review.
- *Synchronous Simulations*. It is very difficult to run such simulations with current readily-available technologies such as email and chat. Howsver, the use of near-

real time internet audio and video phone facilities may provide useful technology for on-line group simulations.

- *Student participation*. It can be a challenge to monitor the level of student participation in distance learning, and this includes the off-line use of simulations. Download hits can be counted, but this is a very crude measure, telling the instructor nothing about what steps in the simulations students went through, whether they finished the simulation exercise and what their results were.

In the future it is likely instructors can make more use of Computer-Supported Co-operative Learning systems to facilitate synchronous on-line simulations by distributed students (Newman et al 1997; Despres and George, 2001). Packaged simulations with limited steps could also be done by individual students or small student groups alone. As with lecture and tutorial simulation exercises, the instructor is still required for simulations with more flexibility to co-ordinate students and, importantly, to co-ordinate simulation review and discussion. Two examples of "distance learning" packaged simulations from our experiences are given below. Both worked reasonably well.

On-line Simulation Exercise

The author has used static, simulation exercise descriptions for students to make use of in their own time in several courses. This includes final year undergraduate Software Engineering courses and Graduate E-commerce Engineering courses. An exercise like the E-commerce system simulation, software architecture or groupware simulation, or CASE tool round-trip engineering simulation, can be written up as a set of tasks a student (or small group of students) are to perform by themselves. Typically Students are required to write a summary of the exercise, possibly for hand-in and assessment. Usually PDF-format static descriptions of the simulation purpose, tasks to perform, and post-simulation review notes are provided.

This approach is easy to incorporate into on-line course materials, takes little time for the instructor, and is done by a surprisingly high number of students, particularly when a review is part of course assessment deliverables. However, it is easy for students to not complete the exercise or to make mistakes which confuse them, unless the simulation is kept very simple. The author has found such exercises seem to work better for graduate students than undergraduates though it is unclear just why this should be.

On-line Animated Simulation

For an industry short course on object-oriented analysis and design co-taught by the author, a rather more crafted simulation exercise was packaged. The course included a description on object-oriented analysis model to object-oriented design model refinement, along with exercises on object-oriented analysis and design modelling. Many participants were able to do the fairly standard modelling exercises, but found the concept of refinement and the refinement process challenging. A post-course, on-line simulation using PowerPoint™ slides was constructed to lead course participants through a simulation of refining a very small object-oriented analysis model to a more detailed design model. This included showing through animation the steps taken by designers to do object-oriented analysis refinement, illustrating the way design-level concepts are derived from related object-oriented analysis concepts. This simulation exercise both complemented the during-course exercises and discussions and complemented post-course on-line exercises.

DISCUSSION

Simulations differ from typical “exercises” in the traditional sense in that rather than tackling a concrete problem, the simulation is like staging a drama. Students “act” as E-commerce system components or participants, CASE tool and programming environment components, networks and distributed system processes, co-operating workers, and analysis and design modellers. This playing the part of anything ranging from an organisational worker interacting with a complex computer system to a discrete component of such a system allows students to get a very concrete feel for the issues relating to such roles in complex Information Systems. Course evaluation feedback from students consistently demonstrates that students find these simulation exercises very illuminating, compared to traditional problem-based learning projects and assignments.

IS educators may not directly need to use any computer-based resources when running their simulations. In fact, in the author’s experience the most successful simulations use simple, readily-available props that are very abstract and simple, with the simulation focusing on the key conceptual or technical issues without recourse to information technology. A key benefit observed from the kinds of simulation exercises described in this paper is the hands-on nature of the tasks, unencumbered by students having to sit at computers or move around complex computing hardware. This has great advantages: a tutorial or lecture is not technology-dependent; students experience first-hand issues to do with conducting tasks rather than second-hand via information technology, and these simulations are invariably fun – often (and sometimes especially) when things go wrong. Students appear to learn more and to retain more when they have enjoyed their classes, and simulation exercises break the (potential) monotony of traditional lecture and tutorial settings. Very importantly, they also encourage students to co-operate and share ideas and problem-solving techniques.

From the author’s experiences when planning on incorporating simulation exercises into tutorials, lectures and distance education materials several things need to be carefully considered:

- Simulations are good for illustrating things like business model functioning; basic system architecture and application operations; the basic nuts-and-bolts of development processes, and how software tools function and can be used.
- Simulations appear not so useful for explaining the detailed techniques of specification and design (the author has found worked examples and group project work is better-suited to these tasks); project management (group projects and selected case studies are better); and IS application testing (case studies and project work are generally better).
- It is very important to choose simple examples to simulate – complex, many-step simulations have too much chance of going wrong and becoming confusing. Instructor participation in simulation is usually good – the instructor can direct things, can move from group to group, or can be an element in the simulation props themselves.

- Students seem to most appreciate and have most success with simulation exercises when the author has used “fun” yet plausible example data.
- Where possible, have students write-up results of simulations for instructor review. This is important, as it ensures students reflect on the concrete steps done in a simulation (whether participated in themselves or observed).
- Be prepared for things to go wrong (and try to recover from it).
- Make it fun. Students invariably give very positive feedback on simulation exercises because they see them as valuable in aiding their understanding but also because they enjoy them.

CONCLUSIONS

Simulations are a valuable tool in helping students understand challenging IS processes, business models, technologies and application behaviour. Simulations naturally suit small tutorial-style settings, where all students can participate in large or small groups, a variety of abstract props can be used to aid running the simulation exercise, and the instructor can spend time discussing simulation results. However, the author has found lecture-based simulations can also work very well when carefully chosen and designed. Typically a small number of students will act out a simulation while watched by the rest of the class, but it is also possible to have small groups of two or three, or even each individual, acting out simulating at their lecture desks, even in large classes of 200 or more students. Distance learning brings new challenges with simulations being pre-packaged and “static”, run and reviewed by students in their own time, or synchronous and “dynamic”. The later requires considerable effort to co-ordinate and reasonably high-end Computer-Supported Co-operative Learning software to support.

ACKNOWLEDGEMENTS

The enthusiastic participation in simulation exercises and the much helpful feedback by many students of the author over many years is gratefully acknowledged. The many helpful comments of the anonymous referees are gratefully acknowledged.

REFERENCES

Bambury, P. A. (1998). Taxonomy of Internet Commerce, *First Monday 3* (10), www.firstmonday.dk.

Bass, L., Clements, P. and Kazman, R. (1998). *Software Architecture in Practice*, Addison-Wesley.

Bolin, S. (1998). E-commerce: a market analysis and prognostication, *StandardView 6*(3): 97-108.

Calazans, N.L.V., Moraes, F.G. (2001). Integrating the teaching of computer organization and architecture with digital hardware design early in undergraduate courses. *IEEE Transactions on Education* 44(2), 109-119.

Coker, D., Marsh, J., Pick, P., Rusjan. E., Thistleton. W. (1999). A Web centered project based learning environment. *Journal of Educational Technology Systems* 27(2), Baywood Publishing, 105-109.

del Rio, A., Rodriguez. J.J., Nogueiras. A.A. (2001). Learning micro-controllers with a CAI oriented multi-micro simulation environment. *IEEE Transactions on Education* 44(2).

Despres, C., George, S. (2001). Supporting learners' activities in a distance learning environment. *International Journal of Continuing Engineering Education* 11(3), 261-272.

Drummond, S., Boldyreff, C., Ramage, M. (2001). Evaluating groupware support for software engineering students. *Computer Science Education* 11(1), Swets & Zeitlinger, Netherlands, 33-54

Ellis, C.A., Gibbs, S.J. and Rein, G. (1991). Groupware: some issues and experiences, *Communications of the ACM* 34(1), 39-58.

Hasman, A., Boshuizen, H.P.A. (2001). Medical informatics and problem-based learning. *Methods of Information in Medicine* 40(2), 78-82.

Land, S.M., Greene, B.A. (2000). Project-based learning with the World Wide Web: a qualitative study of resource integration. *Educational Technology Research & Development* 48(1), 45-67.

Meyer, B. (1998). The power of round-trip engineering. *Journal of Object-oriented Programming* 11(6), SIGS Publications, 93-95.

Newman, R., Johnson, C., Webb, B., Cochrane. C. (1997). Evaluating the quality of learning in computer supported co-operative learning. *Journal of the American Society for Information Science* 48(6), Wiley, 484-495.

Shah, J., Darzi, A. (2001). Simulation and skills assessment. *Proceedings of the 2001 International Workshop on Medical Imaging and Augmented Reality*. IEEE CS Press, pp.5-9.

Smith, R. (2001). Snapshot of a UML tool. *Software Development* 9(4), Miller Freeman, USA, 37-40.

Santoro, F.M., Borges, M.R.S., dos Santos, N. (2000). An infrastructure to support the development of collaborative project-based learning environments. *Proceedings of the Sixth International Workshop on Groupware*. IEEE CS Press, 2000, pp.78-85.