Usability Evaluation Of an E-Learning Tutorial: Criteria, Questions and Case Study

RUTH DE VILLIERS
University of South Africa

Computing systems require rigorous evaluation both of functionality and usability. Evaluation of software within the growing e-learning sector is currently receiving attention. Relevant aspects are evaluation paradigms, techniques, and the issue of who does the evaluating. Squires and Preece developed a set of 'learning with software' heuristics to be used by experts/educators in predictive evaluation prior to adopting a system. These were adapted for post-production end-user evaluation of an operational e-learning tutorial lesson, *Relations*, used in Theoretical Computer Science. Findings are given from a questionnaire survey among learners. This process evaluated the artifact, and also reflectively confirmed the utility of the evaluation technique and criteria. Lessons have also been learned for the future development of educational software.

Categories & Subject Descriptors: *User Interfaces*; H5.2 [**Information Interfaces**]: User Interfaces - *Evaluation/methodology*; *Screen design*; *User-centered design* K.3.2 [**Computers and Education**]: Computer & Information Science education - *Computer Science education; Self-assessment.*
General Terms: Design, Human Factors, Measurement, Performance
Additional Key Words and Phrases: Cognition, e-learning, human-computer interaction, reflection, software evaluation.

1. INTRODUCTION

Software should undergo rigorous testing and evaluation to investigate its functionality, as well as usability analysis to examine the interaction design. Various evaluation paradigms exist, such as predictive evaluation, heuristic evaluation, field studies, formative and summative evaluations, and usability testing in the intended setting [Dix et al, 2004; Preece et al, 2002]. There are different evaluation techniques*:* empirical methods such as observation; experiments and quasi-experiments; surveys via questionnaires and interviews; formal testing in laboratories; usability inspection methods; and protocol analysis to record user actions. There are also formal methods – GOMs, keystroke modelling, etc.

Educational software has been primarily assessed by educators using checklists, but cognisance is now taken of the specialized requirements for evaluation of educational software and web-based learning [Mayes & Fowler, 1999; Preece & Squires, 1999; Reeves & Reeves, 1997; Reeves & Hedberg, 2003]. Instructional transactions involve interactive transfer of knowledge and skills, entailing information access, manipulation, and presentation. They are often conducted by novice learners. Unless there is supportive interaction, can learning occur at all? What techniques, criteria, and instruments should be used for usability evaluation of e-learning applications?

The Centre for Software Engineering (CENSE) in University of South Africa's (UNISA) School of Computing is currently redesigning the computer-aided instruction (CAI) programs it developed in the 1990s. This paper has a dual objective. First, to describe the evaluation of *Relations* – the first of these 2^{nd} generation conversions – so as to learn lessons relevant both to *Relations* and for redevelopment of others. Second, to identify suitable methods and criteria for such evaluations. In a user-centric climate, a learner-evaluation seemed appropriate and the next decision was the selection of criteria.

Squires and Preece [1999] build on Nielsen's [1994] usability heuristics and adapt them to generate a set of heuristics for 'learning with software' (see Section 2). Their approach *integrates usability and learning issues*, and is intended as a set of predictive evaluation principles for experts requiring an instrument for inspecting educational software ranging from generic software packages, WWW applications through to subject-specific multimedia artifacts, etc. This paper builds further on the Squires and Preece criteria, adapting and extending them – not for heuristic evaluation by experts – but as the basis of a questionnaire survey among learners who used *Relations*. *Relations* is introduced in Section 3, while Section 4 presents the evaluation findings.

2. SQUIRE'S AND PREECE'S USABILITY HEURISTICS FOR LEARNING WITH SOFTWARE

Heuristic evaluation is an informal evaluation method proposed by Nielsen [1994], whereby experts – guided by a set of heuristics or usability principles – evaluate designs, storyboards, or operational systems. Nielsen's classic set of ten usability heuristics comprises: (i) *Visibility of system status, (ii) match between system and real world, (iii) user control and freedom, (iv) consistency and standards; (v) error prevention; (vi) recognition rather than recall, (vii) flexibility and efficiency of use; (viii), aesthetic and minimalist design, (ix) help users recognize, diagnose, and recover from*

Author Addresses:
M.R. de Villiers, School of Computing, University of South Africa, P O Box 392, UNISA, 0003, South Africa

errors, (x) Help and documentation. By filtering these heuristics through the concepts of *cognitive authenticity*, *contextual authenticity* and *socio-constructivism*, Squires and Preece [1999] (S&P) customized them to systems for instruction and learning, to be used in predictive evaluation *prior* to selecting systems for use. In the present study they were converted to evaluation questions for a questionnaire survey, undertaken among learners *after* use:

1. *Match between designer & learner models*: This is determined by considering intrinsic feedback and whether it represents cognitive tasks in ways that foster formation of a learner model consistent with the designer model.
2. *Navigational fidelity*: Investigating navigational fidelity involves considering the structure of navigation, cosmetic authenticity and the effectiveness of limited representation of the world, as provided by a system.
3. *Appropriate levels of learner control*: This relates to the balance between learner control, self-direction, customization, consistent protocols, and system responsibility.
4. *Prevention of peripheral cognitive errors*: There is a relationship between domain complexity and error prevention. While usability-related errors should be avoided, cognitive errors are part of the learning process, substantiated by Mayes and Fowler [1999:485] who stress that in educational applications a 'seamless fluency of use is not necessarily conducive to deep learning … the software must make learners think'. Peripheral usability errors should be anticipated and avoided and where possible, novice versions could be provided.
5. *Understandable and meaningful symbolic representation*: Representational forms and symbols should be considered. Interfaces should present low cognitive demands and learners should not have to remember the forms of interaction. The symbols, icons and names used for learning objects should be those of the subject domain and should be used consistently.
6. *Support personally meaningful forms of learning*: There are multiple representations, moreover the software will be used in tandem with various learner support materials. Metacognition should be supported and the software should indicate clearly what learning styles are supported.
7. *Strategies for cognitive error recognition, diagnosis and recovery*: Established techniques, such as cognitive conflict, scaffolding, and bridging should be used to promote the recognition–diagnosis–recovery cycle.
8. *Match with the curriculum*: Software should correspond to curricula and lend itself to educator-customisation.
9. (This category was not used in the present study, since the software was custom-designed for the curriculum.).

3. THE CAI LESSON, 'RELATIONS'

3.1 What is e-Learning?

Some definitions equate e-learning solely with the use of the Internet, but others are broader. Cedefop (2002:5) views it as 'learning that is supported by information and communication technologies (ICT). e-Learning may encompass multiple formats and hybrid methodologies, in particular, the use of software, Internet, CD-ROM, online instruction, any other electronic or interactive media'. This paper views a broad range of learning technologies and computer-based approaches as fitting under the umbrella of e-learning, including the traditional CAI of the 1980s and 90s.

3.2 What is *Relations*?

Relations is a CAI lesson, developed as supplementary study material for *Theoretical Computer Science 1* (COS101), which addresses aspects of discrete mathematics. It conforms to Alessi and Trollip's (1991) definition of a CAI *tutorial lesson*, which 'presents information and guides the student' although it also offers opportunities for practice. It has been used since 1994, so in 2003 it was time for a revamp. Structure and approach were kept constant, but the content was extended. HCI aspects such as fonts, backgrounds, screen displays, colours, colour-coding, etc. were upgraded and general quality was enhanced, using Quest 7.0 as development environment. *Relations* comprises three sub-lessons:

1. Background information
2. Properties of relations
3. Special kinds of relations.

Learners may select which sections/subsections to do, the sequence, and how many times. Within each section definitions, theoretical concepts, and examples are presented, illustrated by diagrams and animations to show step-by-step unfolding of concepts. Teaching is interspersed with fill-in-the-blank questions, most of which entail a composite answer or series of mathematical characters. Some exercises entail synthesis of a relation to meet a particular condition, where more than one alternative would be correct. There are no multiple-choice questions. Detailed diagnostic feedback and explanations are provided, and second attempts must be made after incorrect answers. Each major section ends with a revision summary. *Relations* caters for different stages of learning, offering learners the option to omit theory and 'dive' straight into sets of exercises or revision screens. Answers are not graded, since the lesson is intended for non-threatening learning support. There is an optional test, but the score is not recorded. Figure 1 shows a question with an incorrect answer, followed by detailed system feedback. Figure 2 demonstrates a completed colour-coded animation, along with a question about equivalence relations and co-sets, where the concepts entail different styles of brackets: { }; (,); and [ ]. To answer the question, learners enter series of characters and notations, which are not marked until they have all been entered. The boxed '**Q**' and '**A**' are icons indicating question and answer respectively.

The rules of this 'well-structured domain' are tightly defined and each problem has a single solution or a finite set of answers, where [de Villiers, 2003; Jonassen, 1999; Landa, 1998]:

- Well-structured (closed) domains contain concepts that are procedural or algorithmic, e.g. syntactic, scientific and computational subject matter. Rules are prescribed and problems solved by objective principles.
- Ill-structured (open) domains contain problems with multiple solutions and alternative approaches, e.g. social sciences, business sciences, and the design disciplines, requiring require practice (Schön, 1987).

3.3 Androgogical Philosophy and Learning Theory Underlying *Relations*

The author-designer of *Relations* is the COS101 module leader, whose preferred stance is social constructivism. However, due to its rigid content, the lesson has fundamentally objectivist and instructivist tendencies and uses behaviourist stimulus-response-reinforcement sequences, alternating teaching and practice segments. Yet it also combines paradigms as it sets out to engage learners, support cognition, and anchor learning in authentic contexts. Multi-perspectives portray concepts by text, figures, graphics, evolving animations, and examples. The aim is to foster learning gain, providing meaningful feedback and an optional test as a capstone. There is no formal assessment or learning management. For innovation and engagement, a water 'recreational' theme is available after difficult sections.

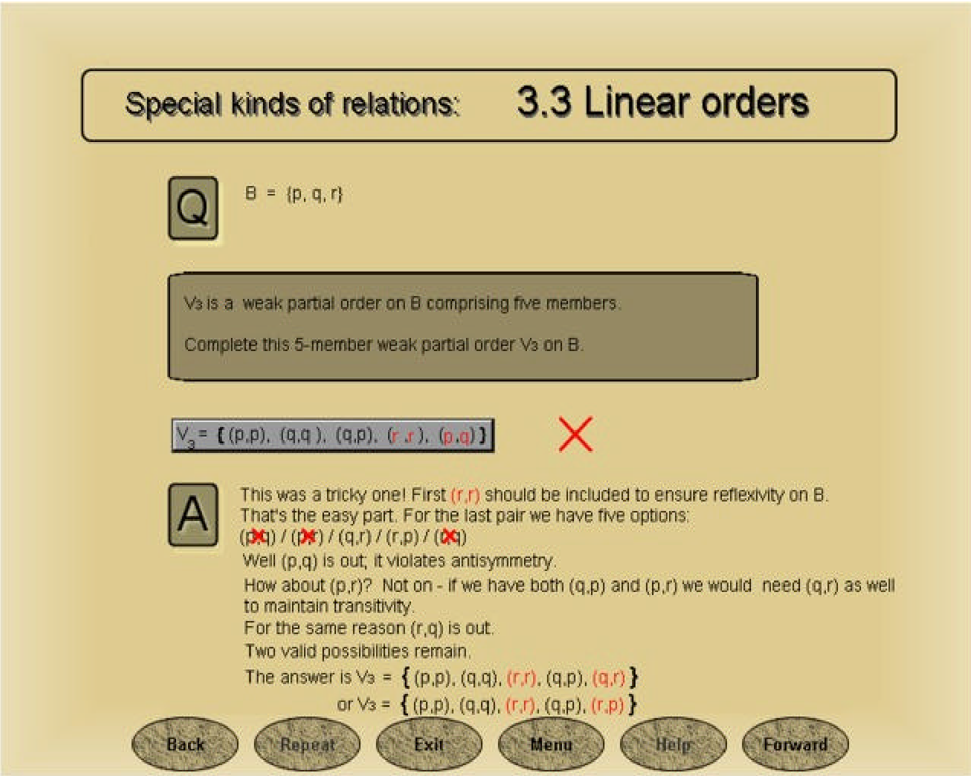

Figure 1: *Question, response, and feedback*

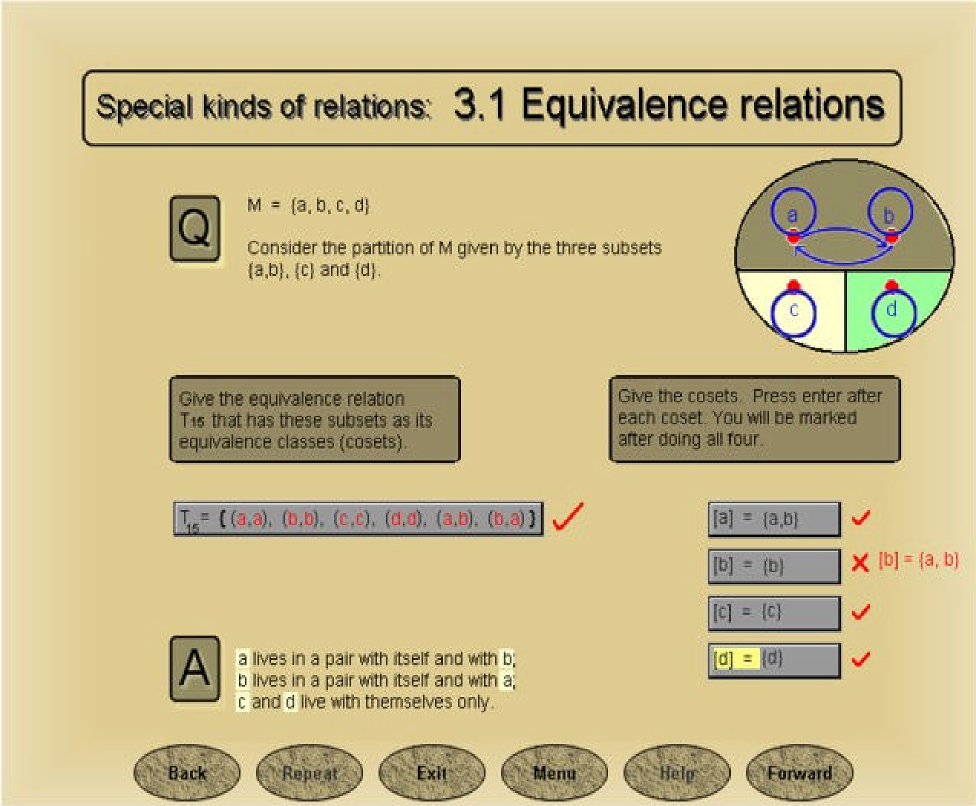

Figure 2: *Animation, question, response, and feedback*

4. THE EVALUATION AND ITS FINDINGS

4.1 From Squires's & Preece's Heuristics to the Evaluation Categories

S&P's *Learning with Software* heuristics were introduced in Section 2. Within these categories, questions were compiled to evaluate ease of use and value in supporting learning. Many students are not first-language English speakers, so questions were simply phrased and terminology was basic, avoiding terms such as 'models', 'cognition', 'navigation', etc. The final section, 'General' investigated attitudes of users to *Relations*' distinctive features, such as the recreational displays, the option to omit theory and the thematic logo. Approximately 60 students purchased the lesson and this study analyses responses to selected questions from the first 20 questionnaires returned. No objective measures, such as post-tests, were taken, but end-of-year exam performance will be noted.

4.2 The Participants and the Survey

The 20 participants were an excellent representation of COS101 students in terms of age, gender, cultural group and status (full-time or part-time student). Only one was encountering computers for the first time; 65% were simultaneously doing a programming module; and 45% used computers in the workplace. The next subsections each use as header an S&P category, followed by the terminology used in the questionnaire.

4.3 Match between Learner and Designer Models: '*The way I understand the theory of relations*'

There were highly positive ratings regarding the cognitive effectiveness of the lesson. Half (50%) felt that some exercises were 'really complicated', while the other 50% did not (third below), attesting to balance within the group.

Doing exercises in the *Relations* lesson helps me understand the theory of relations			
Strongly agree 5 (25%)	Agree 15 (75%)	Maybe —	Disagree —
The feedback (system responses) to my incorrect answers helped me to get them right.			
Strongly agree 9 (45%)	Agree 10 (50%)	Maybe —	Never got any wrong 1 (5%)
Some of the exercises were really complicated			
Strongly agree 2 (10%)	Agree 8 (40%)	Maybe 6 (30%)	Disagree 4 (20%)

4.4 Navigational Fidelity: '*Working my way through Relations'*

In general, responses described the tutorial easy to learn and to use, and the paths through the structure as easy to navigate. The five who chose the 'learned slowly' option were asked to explain and it appeared that four of them were referring to difficulties in the subject matter, not in the lesson.

I learned to use *Relations*:			
Very quickly 3 (15%)	Quickly 12 (60%)	Slowly 5 (25%)	Could not use it at all —
When I use Relations, I know exactly where I am, what parts I have done, and what I should still do.			
Strongly agree 11 (55%)	Agree 6 (30%)	Maybe 3 (15%)	Disagree —
The lesson is easy to operate			
Strongly agree 8 (40%)	Agree 10 (50%)	Maybe 2 (10%)	Disagree —
I got 'lost'			
Strongly agree/Agree —	Maybe 2 (10%)	Disagree 12 (60%)	Strongly disagree 6 (30%)

4.5 Appropriate Learner Control

Students appeared satisfied, but responses in the open-ended section (discussed in Conclusion) show that some want more control, also access at any time to any point in the lesson and the ability to leave a section without completing it. 80% agreed that they would like to choose the level of difficulty, a facility not provided. Within each set of exercises, the difficulty level increases from the start to the end and learners must do them all.

I enjoy being able to choose what to do next:			
Strongly agree4 (20%)	Agree 12 (60%)	Maybe 4 (20%)	Disagree —
I appreciate being able to choose whether to do the theory and the exercises or go straight to the exercises			
Strongly agree 6 (30%)	Agree 11 (55%)	Maybe 5 (15%)	Disagree —
I would like to choose the level of difficulty of the exercises			
Strongly agree 7 (35%)	Agree 9 (45%)	Maybe 3 (15%)	Disagree 1 (5%)

4.6 Prevention of Peripheral Cognitive Errors: '*Kinds of mistakes*'

This relates to usability errors. Domain complexity should not be simplified and cognitive errors relevant to the learning issues should be permitted, but situations that foster usability errors and errors of perception should be avoided. Referring to the red font used in feedback to pinpoint errors, a learner complained that it was hard to see on the blue background. It had already been changed to a tomato red, but perhaps it should be even brighter.

I made mistakes in the way I use Relations, i.e. mistakes in using the system.			
Strongly agree —	Agree 3 (15%)	Maybe 3 (15%)	Disagree/Strongly disagree 14 (70%)
I made mistakes in doing the exercises:			
Got most wrong —	Got some wrong 10 (50%)	Got a few wrong 8 (40%)	Hardly any mistakes / none (10%)
I made mistakes at this stage:			
Many, the first time I used Relations, but less the next time 13 (65%)	The same amount each time 3 (15%)	Hardly any most of the time 4 (20%)	
I made mistakes because I used operations and keystrokes I know from another system			
Strongly agree —	Agree 7 (35%)	Maybe 3 (15%)	Disagree 10 (50%)

4.7 Meaningful Symbolic Representation: '*Symbols, icons and names*'

Ratings were very positive, although one complained that inserting brackets and commas was tedious. Different styles of brackets are key notations within set theory and relations. In some questions these were pre-positioned and learners inserted content only, but in others they had to supply them to distinguish between different kinds. Though frustrating, notation is an essential facet of cognition and mathematical rigour.

The symbols and names that represent mathematical objects are used consistently in the lesson			
Strongly agree 4 (20%)	Agree 14 (70%)	Maybe 2 (10%)	Disagree —
It is easy to grasp the structure of the screen displays.			
Strongly agree 5 (25%)	Agree 12 (60%)	Maybe 2(10%)	Disagree 1 (5%)
Relations helps me understand a difficult section of COS101			
Strongly agree 4 (20%)	Agree 13(65%)	Maybe 3(15%)	Disagree —

I got 'stuck'							
Agree	1 (5%)	Maybe	3 (15%)	Disagree	1(50%)	Strongly disagree	6 (30%)

4.8 Personally Meaningful Ways of Learning

A tenet of socio-constructivist learning theory is that learners should experience personal satisfaction with the process and style of learning. The tutorial structure lends itself to behaviourist implementations, rather than to constructivist designs. But where possible, personal learning styles should be supported. Several attributes were investigated:

When a concept was taught or illustrated in more than one way, it helped me understand it.							
Strongly agree	6 (30%)	Agree	10 (50%)	Maybe	3 (15%)	Disagree	1 (5%)
The ways that *Relations* helped me understand are (participants could choose more than one option):							
Diagrams	13 (65%)	Real life pictures	4 (25%)	Animations	8 (40%)	Practicing	13 (65%)
I enjoy approaching studies collaboratively, i.e. working with a fellow-student.							
Strongly agree	7 (35%)	Agree	4 (25%)	Maybe	5 (25%)	Disagree	3 (25%)
I like doing examples on the computer.							
Strongly agree	10(50%)	Agree	8 (40%)	Maybe	2 (10%)	Disagree	—

The multiple representation approach was used frequently, defining concepts, then illustrating them by diagrams or animations. Only 65% and 40% respectively acknowledged the value of these – less than we would have liked. Collaborative work is an intrinsic part of the social constructivism trend, and 60% of respondents enjoy this style. Although it is complex to implement officially in distance learning, it happens naturally. Two students said they had used *Relations* with fellow-students: 'We can discuss, argue …'. Three used the CAI lesson along with textbook and summaries of rules/definitions. There were spontaneous comments about how examples helped them understand concepts and how they concretized the theory, i.e. inductive learning.

4.9 Cognitive Error Recognition, Diagnosis and Recovery: '*Recognising mistakes and recovering*'

The errors investigated here differ from those of Section 4.6. They address complexities of the domain, common misconceptions and pitfalls, and the adequacy of the feedback in rectifying them. The redesign of the lesson concentrated on HCI features, aesthetics, and enhanced control. An aspect not changed was the original feedback, which had paid careful attention to potential errors. Apart from two requests for 'harder exercises', the feedback obtained high praise, for example: 'It accurately guessed what I did wrong and pointed me in right direction.' / 'I could easily pinpoint my mistake the next time.' / 'The incorrect part was always pointed out so I knew exactly where I went wrong.' / 'Explanations helped.' / 'The feedback uses definitions and helps me to remember definitions and theory. And finally, 'Keep it up for giving us such a chapter on the computer. I think you can do it on other chapters too.'

The feedback to my incorrect answers was useful.							
Strongly agree	6 (30%)	Agree	13 (65%)	Maybe	1 (5%)	Never got any wrong	—
When I got an answer wrong, I was able to get it right on the next try.							
Always	4 (20%)	Nearly always	13 (65%)	Sometimes	2 (10%)	Not often	1 (5%)
There is too much information on the screens; it confuses me.							
Agree	1 (5%)	Maybe	4 (20%)	Disagree	11(55%)	Strongly disagree	4 (20%)

4.10 General: distinctive features, and engagement

Here the distinctive attributes of the lesson were addressed, such as its water sport recreational theme. Following complex sections, screen displays present water-based activities – e.g. a pool or boardsailing – to help users take a break. Figure 3 shows a clip from one such screen, as well as the logo theme character.

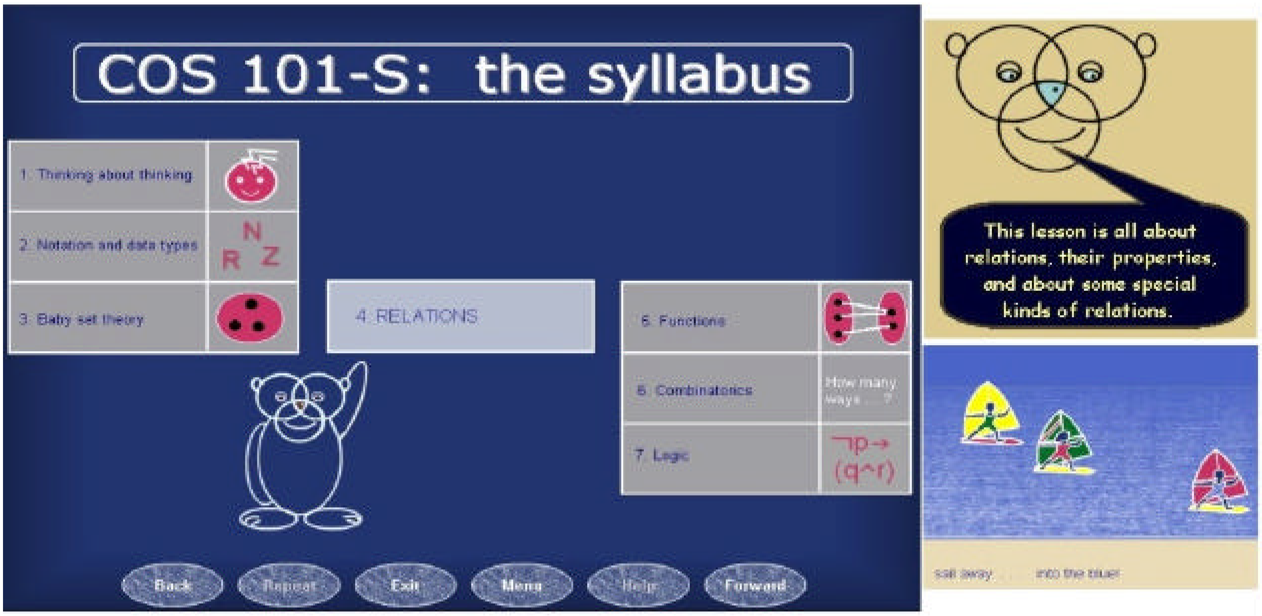

Figure 3: *Clockwise from left: the logo theme character*; *the logo's face*; *a water-sport recreational screen*

The recreational screens with the water theme (participants could choose more than one option):			
Are fun 7 (35%)	Give me a break 1 (55%)	Make no difference/irrelevant 9 (65%)	Are irritating 1(5%)
The shortcut 'Dive' screens to avoid redoing theory are useful.			
Strongly agree 4 (20%)	Agree 10 (50%)	Maybe 5 (25%)	Disagree 1 (5%)
Think about the lesson's logo character. What do you think it represents?			
Venn Diagram 7 (35%) (correct)	Relations / COS101 4 (20%) (acceptable)	Other (e.g. teddy) 4 (20%) (incorrect)	No answer 4 (20%)

In open responses, students commented on the water theme: 'It made me laugh.' / 'It's too stressful to do the whole CD in one go – they help me take a break.', but also: 'They wasted time'. A 19-year old criticised the animations: 'This CD helped me grasp concepts much easier. Thanks for a terrific job. But the animation is poor. You guys aren't Hollywood studios, but you are capable of something more exciting. Take it as a challenge.'. Another 19-year old said: 'Well done, you guys, it's great! It beats reading the books as it goes straight to the point'. So '*us guys*' get both bouquets and brickbats ... Students appreciated the 'dive' option: 'If you know the theory, you just go for the exercises.' / 'It saves time.' / 'Useful when revising for exams.' / ' I did not use the Dive, but went through the whole lesson every time.'. Referring to the neutral character who dives into a pool on command: 'Pictures of real babes diving would be nice!'. Regarding the theme logo, based on a Venn Diagram, comments varied: 'It's kinda Cool.' and 'I'm one of those people who don't appreciate cuteness.', but only 35% realized that it represented a Venn Diagram.

Some requested more customisation, 'Allow us to select at the beginning what screens we want.' / 'I would rather use the menu to be where exactly I want to be.' / 'It would be useful to navigate to *any* parts of the lesson at any time.'

5. CONCLUSION

First, a reflective review of the Squires and Preece heuristics. Although intended for predictive heuristic evaluation, they handled the shift to post-production admirably. The eight categories of questions served well in an end-user questionnaire survey to investigate the educational worth and usability of *Relations*. Useful findings ensued, relating mainly to strengths and utility of the tutorial, but also identifying matters requiring improvement and refinement. The type of information obtained under our eighth category, 'General', prompts the suggestion of two further categories:

9. Distinctive features, unique to the environment being evaluated.
10. Capacity of the system to engage learners and hold their attention.

In further research, comparison is planned between these findings (using 50 students) and expert heuristic evaluation.

Second, what do the findings contribute to future development and upgrades of e-learning lessons? Complex and creative programming techniques were ably used in constructing *Relations Version 2* and a professional product was produced. Learners praised it and many requested CAI lessons for other parts of COS101 or for further modules. Some, however, required alternative navigation strategies and advanced customisation. This request came from only three students, but the issue should be considered, and hyperlinked control should be implemented, where appropriate. For foundation teaching in a structured domain, linear progress through material is essential, but where environments are supplementary material, flexibility can be enabled. Regarding the request for further topics, a prototype on more complex relations is being upgraded to a full environment. This artifact satisfies the requirement for access to any point at any time, since it is not a sequenced tutorial but an interactive practice environment with definitions, etc. available on demand. Further valid navigational requests were:

- exit and re-entry facilities.
- direct return to an exercise in question after using the <Back> option to access definitions/theoretical screens, i.e. immediate return, status-maintained, without re-navigating through interim screens.

Some students, particularly computing professionals, expect e-learning applications to maintain corporate standards and conform to global conventions. For example, in responses under 4.6, 35% made mistakes due to using functions they knew from other systems. *Relations* does not conform to Dix et al's [2004] usability principles of *consistency*, *predictability* and *familiarity* in all respects, although it is internally consistent with respect to its own patterns. Courseware authoring tools do not naturally support non-linearity and certain other contemporary paradigms, but developers of educational software can creatively overcome obstacles. However, criticism was accompanied by praise, e.g. 'I found it very useful. From a learning perspective, nothing wrong. Great stuff. Wish there was more!!' Spontaneous comments indicated that participants had experienced 'flow' [Csikszentmihalyi, 1990]: 'I never got frustrated.' / 'Once you start, it's not easy to quit.' / 'I must say, it was FUN working with it.'.

ACKNOWLEDGMENTS

Malene le Roux: for meticulous, innovative programming; Tersia Parsons: for the graphics and animation; Estelle de Kock & Daphne Becker: evaluation and valuable contributions. And gratitude to the founders of the CAI branch of CENSE.

REFERENCES

ALESSI, S.M. and TROLLIP, S.R. 1994. *Computer-based Instruction: methods and development.* Prentice Hall., Englewood Cliffs, N.J.

CEDEFOP (EUROPEAN CENTRE FOR THE DEVELOPMENT OF VOCATIONAL TRAINING) 2002. *E-learning and training in Europe*. Luxembourg: Office for Official Publications of the European Communities.

CSIKSZENTMIHALYI, M. 1990. *Flow: The psychology of optimal experience*. Harper & Row, New York.

DE VILLIERS, M.R. 2003. Foundation for structure: a learning theory approach to instructional design for e-learning. *International Journal of Learning Vol 10. Proceedings of the Learning Conference 2003*. Common Ground Publishers. In press.

DIX, A., FINLAY, J., ABOWD, G.D. and BEALE, R. 2004. *Human-Computer Interaction*. Pearson Education, Ltd, Harlow, 318–364.

JONASSEN, D. 1999. Designing Constructivist Learning Environments. In *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory Volume II*. C.M. REIGELUTH. Ed. Lawrence Erlbaum Associates, Mahwah, N.J.

LANDA, L.N. 1998. Methodological Drawbacks of Current Educational Software and Suggestions for Radical Improvement Based on Algo-Heuristic Theory. *Educational Technology*, *38*(5):54-56.

MAYES, J.T. and FOWLER, C.J. 1999. Learning technology and usability: a framework for understanding courseware. *Interacting with Computers 11(5)*: 485–497.

NIELSEN, J. 1994. Heuristic evaluation. In *Usability Inspection Methods.* J. NIELSEN & R.L. MACK, Eds. John Wiley & Sons, New York, 25–62.

PREECE, J., ROGERS, Y., and SHARP, H. 2002. *Interaction Design*. John Wiley & Sons, New York, 317–358.

REEVES, T.C. and HEDBERG, J.G. 2003. *Interactive Learning, Systems Evaluation*. Educational Technology Publications, Englewood Cliffs, NJ.

REEVES, T.C. and REEVES, P.M. 1997. Effective dimensions of interactive learning on the World Wide Web. In: *Web-based Instruction*. B.H. KHAN, Ed. Educational Technology Publications, Englewood Cliffs, N.J.

SCHoN, D.A. 1987. *Educating the Reflective Practitioner.* Jossey-Bass Publishers, San Francisco…

SQUIRES, D. and PREECE, J. 1999. Predicting quality in educational software: Evaluating for learning, usability and the synergy between them. *Interacting with Computers 11(5)*: 467–483.