

SHAARON AINSWORTH & PIERS FLEMING

TEACHERS AS INSTRUCTIONAL DESIGNERS:
DOES INVOLVING A CLASSROOM TEACHER IN THE
DESIGN OF COMPUTER-BASED LEARNING
ENVIRONMENTS IMPROVE THEIR EFFECTIVENESS?

Abstract. The REDEEM authoring tool was developed to allow teachers without programming knowledge to design computer-based learning environments (simple Intelligent Tutoring Systems) for their students in a time-effective manner. It was hoped that the systems that resulted from teachers authoring with REDEEM would be both effective for learners and easy for teachers to support and to integrate into their classroom practice. In this paper, we review a five year programme that was aimed at assessing the strengths and, just as importantly, the weaknesses of this approach to developing computer-based learning.

1. INTRODUCTION

The REDEEM (Reusable Educational Design Environment and Engineering Methodology) authoring environment is designed to allow classroom teachers to create simple Intelligent Tutoring Systems (ITSs). It works by allowing teachers to import computer-based training (CBT) as the domain content and then overlay their views about how such material should be taught to their students. Thus, teachers' roles are to select relevant material for their students and then make pedagogical decisions about such factors as the complexity of the topics, appropriate sequences through material, how to enhance interactivity and what teaching strategies would be most appropriate for particular types of learners.

Researchers coming from a strong Artificial Intelligence tradition might argue that the REDEEM approach is likely to prove ineffective. REDEEM ITSs are very simple systems and as a number of studies have shown that increasing the "intelligence" of systems improves learning outcomes. For example, Mark and Greer (1995) compared versions of a video recorder tutor. The "smartest" one used sophisticated student modeling techniques to monitor learners' performance and could give detailed feedback on misconceptions, whereas a "dumb" version allowed learners only one way to perform a task and provided only simple prompting. The smart system decreased the number of steps, errors and time required for students to complete the post-test. Designers of ITSs hope that one day their systems will perform as well as expert human tutors. Bloom (1984) found that one-to-one tutoring by expert tutors compared to traditional whole class teaching improved students learning by a 2 sigma effect size. Currently state of the art in ITSs is around a 1 sigma effect with evaluations of ITSs revealing effect sizes of between 0.4 and 1.2 compared to classroom teaching (*e.g.* Graesser, Person, Harter, et al, 2001; Koedinger, Anderson, Hadley, & Mark 1997).

However, the time and expertise needed to produce such “clever” systems has meant that such ITSs have not yet achieved widespread application in schools, colleges or workplaces - creating an ITS is estimated to take more than 300-1000 hours to produce an hour of instructional material (*e.g.* Murray, 1999; 2003). Furthermore, teams needed to create ITSs typically include knowledge engineers, software developers, domain experts, teaching experts, usability engineers, *etc.* Most authoring tools substantially reduce the time needed by an order of magnitude and many supplement the need for some members of the development team. However, research suggests that the difficulty of creating ITSs may not be the only reason that such systems are not found in every classroom - teachers may resist the introduction of such systems. For example, Major (1995) found that teachers were worried that ITSs would not reflect their own pedagogic concerns and approaches but would instead embody the beliefs of the system designers. One obvious solution is therefore to empower the teachers to be the designers of the environments that their students learn with in the classroom. But, most teachers are not trained as computer programmers and have only a very limited amount of time available for lesson preparation. Furthermore, it remains an empirical question whether teachers have appropriate expertise to create such systems even if tools are provided that are usable by non-programmers in a time-effective way.

Consequently, the value of an approach such as REDEEM's is not easy to predict. On the positive side, non-programmers can use REDEEM to create courses in a very broad range of areas, and will be able to do so more quickly with REDEEM than with almost any other authoring environment. The people making the decisions about what and how students should learn will be professionals who make these decisions (in non-computational situations) everyday of their working lives. On the negative side, the systems created will have very shallow domain models and will be severely limited by the content of the imported material. To evaluate the value of REDEEM's position on the tool usability versus ITS flexibility trade-off, it is necessary to explore both the experiences of the authors and the learners. In this paper, we will briefly describe how REDEEM works and then describe the results of a five year evaluation programme.

2. SYSTEM DESCRIPTION

The REDEEM suite was developed in Click2Learn ToolBook Instructor and runs on Windows 95+. It consists of two main pieces of software (authoring tools and ITS shell) through which users interact with external courseware catalogs (see Figure 1).

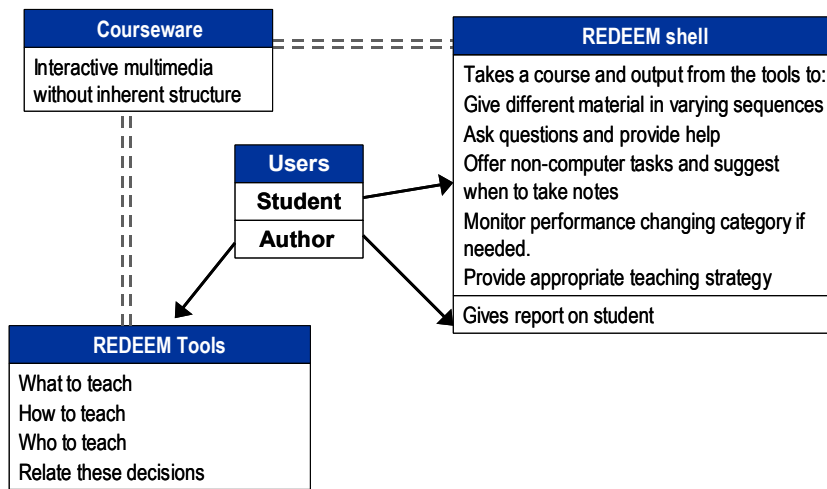


Figure 1. REDEEM schematic

2.1. Courseware Catalogues

Domain material in REDEEM is based on the idea of a courseware catalogue. It consists of pages of either Click2Learn ToolBook or HTML/Gifs/JPEGS. The ideal courseware for REDEEM presents discrete pages of material showing different aspects of the domain at varying levels of difficulty. Pages can contain multi-media, simulations, animations, questions and exercises. REDEEM does not model learners' actions on these objects.

2.2. Authoring Tools

REDEEM's authoring tools decompose the teaching process into a number of separate components. Essentially authors are asked to describe what they are teaching, whom they are teaching and how they would like to teach these students. This information is then combined by assigning particular teaching strategies and types of material to different learner groups.

2.2.1. *What to Teach*

The first task that the author must perform is to give each page a name; other tasks can then be performed in any order. Sections are created by combining pages, which need not consist of contiguous pages in the underlying CBT. Pages can be in multiple sections. Sections are then described upon a number of dimensional ratings, *i.e.* familiar, easy, general or introductory. Authors can describe relations between sections – for example, the prerequisite relation, which ensures that a section is not taught until prerequisite sections have been completed. Pages themselves are described in terms of these dimensional ratings and relations. These tools provide information that the system uses as a semantic network to describe the structure of the teaching material. This network enables the shell to make default decisions about adapting content and to implement teachers' preferred routes through material.

The next stage is to add interactivity. Authors create questions (multiple choice, fill in the blank, multiple true, true-false or matching questions) and provide feedback that will explain to the student why an answer is correct. The author can create up to five different hints for each question, which ideally increase in specificity. Authors describe a number of characteristics of the question that the ITS shell uses to implement a specific teaching strategy (*e.g.* difficulty, pre or post-test). Authors can also associate a reflection point (which means that students are prompted to take notes) or non-computer task (which directs student attention to an another activity) with a page.

2.2.2. *Who to Teach*

Authors define a set of student categories at any degree of granularity, ranging from a whole class to an individual child. Teachers in our studies have tended to use performance-based measures (*e.g.* high flyer, struggler) or task-based measures (*e.g.* revising) or have combined these (*e.g.* high reviser). But, it is possible to use any dimension that authors find appropriate. The validity of performance-based categories can be evaluated against students' question performance. The shell can automatically change the category as the overall standard of the student (as defined in the shell's student model) changes. If this occurs then both content and teaching strategy may change as the system macro-adapts to the new category.

2.2.3. *How to teach*

REDEEM supports multiple teaching strategies. Different instructional principles can be embodied in various strategies by manipulating sliders. Each slider in Figure 2 has three discrete positions that result in different instruction. For example, teachers could create a "Free Discovery" strategy where students choose the material they see, the questions they answer and the number of attempts allowed for each question, when to receive help and whether to perform non-computer-based tasks. In contrast, a teacher might create a "Guided" strategy where students have no choice over material, when questions (of certain types and difficulties only) are included and asked immediately after the relevant material has been presented and help is

TEACHERS AS INSTRUCTIONAL DESIGNERS

given on error with a limited number of attempts for each question. REDEEM can offer nearly 10000 different teaching strategies, each (very) subtly different to each other. Although to date no author has created more than seven for a particular class of students.

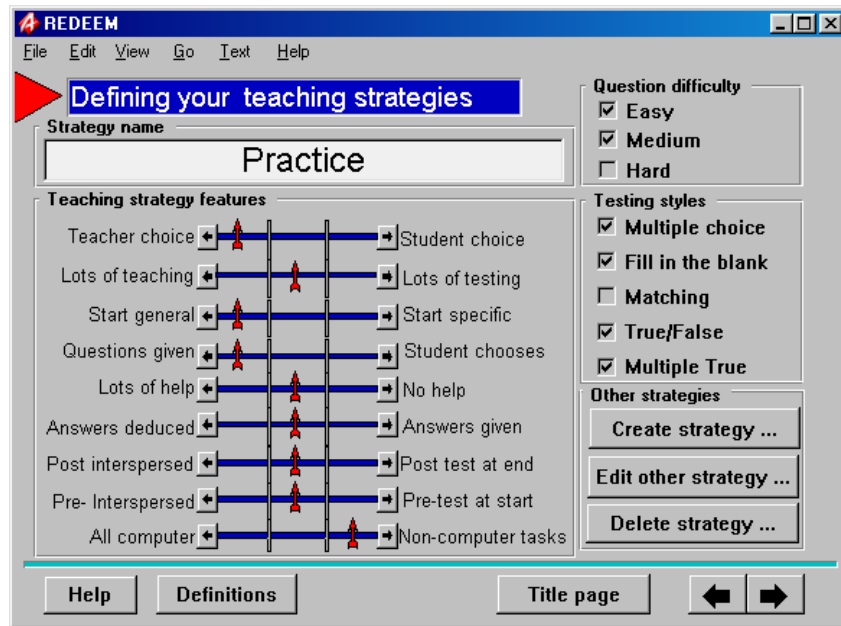


Figure 2. Creating a teaching strategy

2.2.4. Adapting Material and Strategies to Student Categories

By default, learners see all the material, but the author can choose to remove sections for a particular category (*e.g.* to focus on introductory material for learners who need more help or include extension activities for students who could benefit from more challenging tasks). Each student category is also given a teaching strategy. To date, authors have varied from creating a single preferred strategy to creating a unique strategy for each group or even for an individual student.

2.3. ITS Shell

The REDEEM ITS shell uses the output of the authoring tools, together with its own default teaching knowledge, to interpret the courseware in such a way as to deliver adaptive, interactive instruction. The main role for the ITS shell is to deliver the course material to each student in the way that the teacher specified with the authoring tools. Tutorial actions available to the shell (depending upon the teaching strategy) are: to teach new material; offer a question (and help if appropriate);

suggest that students make notes on the on-line tool; offer a non-computer based task and by means of password protection check that it has been completed; or summarize students' progress. To achieve these functions, REDEEM employs a basic overlay model that records the system's understanding of the students' knowledge of an area. The values of the model change over the course of a session as the student sees new material and answers questions. This model is primarily used when student categories are performance based and determines if learners should change student category. The shell also maintains a student history. This is used to offer reports to the author either on an individual student's progress, a student category's progress or to give a report on the course.

3. EVALUATING REDEEM

The success of REDEEM depends on two main features – the usability of the authoring tools and the effectiveness of the learning environments.

Firstly, teachers must be able to use REDEEM to create learning environments that match their requirements in a time-effective manner. REDEEM must be easy to learn and require only a short training period. Interfaces should be matched to authors' needs, providing simple to use tools and providing appropriate feedback on the consequences of their authoring decisions. REDEEM, because its intended users are classroom teachers, requires higher usability standards than is necessary in ITS authoring tools aimed at more specialized authors. Authors should be able to create ITSs that reflect their own pedagogic principles and that meet the learning needs of their students so REDEEM must offer appropriate teaching decisions in ways that match teachers' ontologisation of the teaching-learning process. Given the time constrained nature of the process, it is equally important to limit the functionality that authors do not require. For example, if all authors make very similar decisions about a teaching strategy dimension, then choice in this dimension is redundant and should become hard-coded into future systems.

Secondly, the environments that teachers create should be effective at supporting their students' learning. Ideally, learners will come to understand the subject matter more completely, will have found the experience of learning motivating and should reach the desired outcome in a more time efficient way. However, to prove that a system led to successful outcomes is not an easy task. To be done successfully, large-scale experiments are often needed. Furthermore, the effectiveness of any learning environment is influenced by its context of use. For REDEEM the evaluation problem is compounded by the fact that system that results from authoring depends on the author decisions, the choices the authoring tools offered, the shell's interpretation and delivery of these decisions and the externally created domain material. The nature of the experimental control in such evaluations is very important. Often, a system may be compared against a human teacher or learning through textbooks. We believe that such a comparison is not useful for our purposes because the differences between the two are so large that little sensible can be said about **why** any difference in learning processes or outcomes occurred. One alternative often used in Artificial Intelligence are ablation experiments where

TEACHERS AS INSTRUCTIONAL DESIGNERS

particular design features are removed and performance of the systems compared (*e.g.* Cohen, & Howe, 1988). They also allow analysis of the contribution that specific system features bring to the learning experience. In REDEEM's case this seems a very appropriate methodology as the question we want to answer is whether investing teachers' time in REDEEMing existing CBT is worthwhile. Hence, we have conducted a number of experiments that have compared learning with the original CBT to learning with the REDEEMed CBT.

3.1. Results of Evaluations

3.1.1. Usability of the Authoring Tools

The REDEEM authoring tools have now been used by many different types of author: including primary (K-12) and secondary (high) school teachers, lecturers, researchers, military instructors, and student teachers. This has provided an unprecedented opportunity to explore their usability and functionality over many user groups. Overall, we have found that it is possible for all these classes of user to use the tools to express, represent and assess their teaching knowledge to create an ITS within a feasible time scale. Initial training in the use of the REDEEM tools requires between one and two hours. No author has found the overall decomposition of teaching process incompatible with his or her approach. REDEEM's reliance on graphical manipulation of sliders and form-fill style interview tools has proved simple and easy for authors to use (see below for the single exception).

To stand a realistic chance of use in the classroom, authoring tools must also be efficient of teachers' time. In one study (Ainsworth, Grimshaw & Underwood, 1999) authors took between six and eleven hours to author a four hour course on "Understanding Shapes) - a ratio of around three hours per hour of instruction. Ainsworth & Grimshaw (2002) found that a teacher took less than 25 hours to create two ITSs (around eight hours of instruction). Navy authors began by requiring 10 hours per chapter (around 6:1), which dropped to six hours by the end of authoring (around 3:1) (Ainsworth, Williams & Wood, 2001). Furthermore, the time consuming aspects of the ITS development lies in the domain authoring. When trainee teachers were presented with a previously authored course that they simply had to individualise to their students, they only required 90 minutes to customize the four-hour course (Hayes et al, 2001).

We found that for the most part, authors wanted the functionality that REDEEM provides and they could use it to create ITSs that reflect their own pedagogic preferences. However, the focus of authors' attention was radically different depending upon whether they were classroom teachers or military instructors. The teachers wanted to structure the domain material in ways that reflected their own beliefs (*e.g.* by creating sections of differing complexity they could differentiate course material for different groups of learners). Classroom teachers also used REDEEM to differentiate their teaching strategies. Five out of six teachers created at least five student categories and associated a unique teaching strategy with each one. In contrast, the military authors were much less interested in creating individualised

experiences. They used REDEEM to create a single teaching strategy that reflected their own preferred strategy rather than the courseware designers' strategy. What both groups had in common was the way they increased interactivity by adding questions and reflection points. Adding this interactivity was one of the time-consuming tasks, not because of its complexity but for the most part because the authors valued it so highly.

After authoring, the designs of the ITSs were recognized by all the authors as closely related to their instructional approach. They expressed more satisfaction with their ITS than with the original CBT and felt they would be more likely to accept their REDEEM ITS in their classroom than the CBT alone or another author's ITS. However, these studies have revealed some interesting dilemmas about REDEEM's design. Unsurprisingly, a number of small changes to the REDEEM interface have been made over the years based on author feedback (*e.g.* more undo functions), but a number of more interesting design issues have been revealed.

Firstly, in common with most ITS authoring tools, REDEEM requires users to shift from story boarding to knowledge-based authoring (*e.g.* Major, 1995). For example, authors describe characteristics of the page (*e.g.* familiarity and complexity) and then the shell computes a route. However, this proved too time-consuming, unpopular and was resisted. This knowledge-based authoring ignores the important role that narrative plays for authors and for learners when they are interacting with new material. Our experience suggests that this call for knowledge-based authoring may have been overstated, at least for designing declarative instruction.

A second related dilemma is whether we should encourage teachers to envisage a future where they work in collaboration with more intelligent software or instead provide them with tools to create software to fulfill their needs today. Compared to the functionality that REDEEM offers, it is apparent that teachers mostly used REDEEM to create customized CBT rather than adaptive ITSs. For example, authors often wanted control over the order of presentation of material, like CBT, and felt uncomfortable at releasing this role to REDEEM. No teacher has chosen to let the system macro-adapt based on student performance.

Thirdly, it is apparent that more attention is needed on how the REDEEM tools support authors in understanding the consequences of their decisions. In the first version of REDEEM, the only way to check the consequences of an authoring decision was to swap to the ITS shell and behave as a student. This is easily achieved and is an excellent way to check the consequences of adjusting features such as teaching strategies, but it is a poor way of checking micro-structural issues such as the position of questions. Consequently, visualization tools have been developed so that authors can more easily see the consequences of their actions. However, as any visualization will remain an abstraction from what might actually occur in the ITS shell (given that an ITS responds to a student's performance) this raises a host of other design problems. For example, graphical representations are not an ideal way to represent information with a lot of uncertainty (*e.g.* Stenning & Oberlander, 1995).

Finally, time is crucial. Authors need to create learning environments in a time-efficient way. REDEEM achieves a ratio of around three hours of authoring to one

hour of instruction when a trained author, familiar with the domain and with teaching, creates an ITS from imported domain material. This ratio substantially undercuts the majority of ITS authoring tools. A lesson from our authors was “this is still not fast enough!”. One tension here was the need for control and ownership of the software versus the time taken to customize it. As discussed above authors were uncomfortable with leaving the computer to determine sequence and so spent lots of time ensuring the system was eventually hard coded to their preferred sequence. They often spent a considerable amount of time on details that may not be too important (*e.g.* one teacher edited another ones help messages to include a full stop after each message). It would be interesting to see how authors would behave if they were routinely authoring many courses. However, on the positive side one of the features that teachers appreciated must was the way that REDEEM allows quick assignment of different strategies and content to different student categories. Analysis of authoring times suggests that this is normally achieved in around 50 minutes, irrespective of the size of the course.

3.1.2. *Effectiveness of the REDEEM courses*

Analysis of the usability and functionality of REDEEM has mostly revealed positive results. However, for REDEEM to be truly useful, it should also make learning more effective. We have conducted five studies with REDEEM with a variety of authors (classroom teachers, military trainers and researchers) and learners (schoolchildren, university students, military personnel) (Table 1). We also used three different types of content material: Genetics, which was developed for REDEEM; Communications and Information Systems Protocols (CISP), which was developed for use in Naval classrooms as standalone CBT; and PC and Networking, which was adapted from legacy CBT in use in the Navy.

In order to reduce the effects of participant variance, a crossover design was employed in all studies such that all participants received half the material under REDEEM and half as CBT only. For example, in Study 5 half the learners received REDEEM PC and CBT Networking and half CBT PC and REDEEM Networking. Multiple Choice pre and post-tests were developed that contained some questions that identified knowledge that was directly questioned during the REDEEM intervention versus those that that addressed material covered by the course but were not directly questioned by REDEEM. In Studies 1 and 2, the conceptual richness of the domain also allowed us to create near transformation questions which addressed the same issue as a REDEEM question but were manipulated so that memorization was not sufficient. Subjects’ scores at pre-test were used to ensure that there was an equal distribution of ability across the two conditions. Hence, the design was identical for all the studies with the exception that Studies 1 and 2 used versions of multiple REDEEM ITSs macro-adapted to different learner categories and Studies 3 to 5 used only a single REDEEM ITS.

Table 1. Summary of learning outcomes studies conducted with REDEEM

Study	Subjects	Course	Time	Author	Versions	Gain	Effect size
1. Genetics at Uni.	86 14-16 yr	Genetics	8 hrs	Teacher	5 varying content, strategies	RED = 10% CBT = 8%	0.21
2. Genetics in School	15 14-16 yr	Genetics	8 hrs	Teacher	3 varying content	RED = 16% CBT = 8%	0.82 *
3. Navy	19 17-22 yr	CISP	15 hrs	Trainer	1	RED = 21% CBT = 22%	-0.04
4. Student	25 20-28 yr	PC & Networks	3 hrs	Researcher	1	RED = 53% CBT = 44%	0.82*
5. RAF	16 20-45 yr	PC & Networks	3 hrs	Researcher	1	RED = 47% CBT = 32%	0.76 *

Note * = statistically significant difference. Effect size = (gain in experimental condition – gain in control)/standard deviation in the gain of the control group.

Overall, REDEEM led to 30% improvement from pre-test to post-test, whereas CBT increased scores by 23 %. This advantage for REDEEM ITSs translates into an average effect size of 0.51. This compares well to non-expert human individual tutors (an average of 0.4 sigmas; Cohen, Kulik, & Kulik, 1982) and is around 0.5 below the typical full blown ITS.

There are a number of things to observe about the results. Firstly, the variability in the outcomes is incredible and is in fact even greater for individual courses (the highest effect size is for REDEEM Genetics 1 versus CBT Genetics 1 in study 2 where the effect size was 1.33 sigmas). Interpretation of these results requires overcoming a large credit assignment problem. There is no consistent relationship between whether the course was authored by a researcher or a practitioner, the topic taught, whether the study was conducted in an artificial situation or in a realistic context, and whether REDEEM's macro-adaptive features were used to create learning environments for specific learner categories.

However, analysis of process data does go some way to explaining these results. Many studies showed significant correlations between amount of notes written, and time spent learning with REDEEM and learning outcomes. All studies found a significant relationship between percentage of questions answered correctly first time during the learning session and outcome, even when prior knowledge is partialled out. Unsurprisingly learners who took advantages of REDEEM's features learnt more than those who did not. The two studies without a REDEEM benefit were also the ones with the highest degree of students who were least interested in learning this material. Analysis of the process logs suggests a wide variation in how much learners interacted with the systems. In some studies, we have also presented learners with questionnaires asking them to identify what they felt most benefited their understanding. The most consistent response has been that questions with appropriate feedback and hints are viewed as most beneficial. It remains a valuable

TEACHERS AS INSTRUCTIONAL DESIGNERS

lesson to system designers and researchers – adding features to enhance learning can only impact when learners chose to engage with them.

REDEEM differs from the underlying CBT in a number of ways (depending upon authors' decisions) but the two most crucial are the interactive and the macro-adaptation features. The results of the studies suggest that any advantage of REDEEM was due more to increasing interactivity than to macro-adaptation. Studies 1 and 2 included macro-adaptation to different learner categories (either by content or by strategy and content) but were no more successful than studies with only a single student category. That increasing learner interaction increased learning outcomes does not seem contentious but the question that remains is why we did not observe benefits from macro-adaptation. The teacher's use of REDEEM's content and teaching strategy adaptation features are primarily in line with that of the research literature (see Ainsworth & Grimshaw, 2002). However, the two teachers who used macro-adaptation categorized learners solely in terms of prior performance, but many other variables can be explored (*e.g.* learning style, working memory capacity, self-regulatory skills, visualiser/ verbaliser, gender, high anxiety/low anxiety, level of cognitive development) and potentially these should have been included. Furthermore, many of these factors may show up in laboratory studies but their effect size may be somewhat weak and they may have little impact in the classroom. The number of students in each category would often not have been sufficient to allow identification of benefits unless they were very substantial. Moreover, from this design they might be difficult to identify. For example, if an author assigned a unique teaching strategy to every category of learner and they all made equal gains, does this mean that the strategies were ideally targeted or that they had no effect? Consequently, we need further research to examine the educational significance of macro-adaptation and to consider which are the most important learner characteristics and strategy dimensions. It remains an open question whether it is educationally effective to perform pre-test or psychometric tests to assign students to different learning environments.

Finally, all studies found a greater advantage for learners who received REDEEM first rather than CBT first. The direction of this benefit is positive, *i.e.* students transferring from REDEEM to CBT do better than predicted. This may suggest that learning with an environment that encourages greater interactivity may help when learning with an environment without these beneficial features. It is also worth noting that this will have served to reduce any differences between learning outcomes in the experiments.

4. CONCLUSION

The REDEEM authoring environment was developed as a response to a desire for teachers and trainers to be involved in the development of adaptive learning environments for their classroom. The evaluation studies we have conducted over the last five years suggest teachers can be successful instructional designers if given usable tools that offer appropriate functionality in a time-efficient way. The learning experiences of their students has generally been enhanced, especially when learners

take advantage of the features authored by the teachers. The studies have revealed that teachers have complex models of teaching and learning which underpin their work. They do not adhere to simple one-dimensional accounts of learning. The flexibility that REDEEM provides for adapting CBT should allow such CBT to be used in ways that are appropriate to different teacher's methods. It is often said that we don't expect every teacher to write their own textbook, why should we expect them to design their own ITS? However, all teachers do customize their textbooks to use in their classroom by suggesting an order to read chapters, explaining difficult terms, providing exercises and worksheets, *etc.* REDEEM supports the teaching and learning process in the same way - the domain material is not created by teachers but its pedagogical functionality is. Hence, although we do not consider that the REDEEM approach is suitable in all circumstances and the results of the empirical studies suggest that REDEEM ITSs fall short of the 2-sigma effect, we propose that much can be gained by providing teachers with simple ITS authoring tools.

5. ACKNOWLEDGEMENTS

This work was supported by the ESRC at the ESRC Centre for Research in Development, Instruction and Training and by the Office of Naval Research under grant no N000 14-99-1-0777. Many researchers participated in this project including Nigel Major, Shirley Grimshaw, Alan Richardson, Mary Hayes, Jean Underwood, Ben Williams and David Wood. Iona Bradley, Sue Cavendish, Wendy Crossland, Ruth Guy-Clarke, Sue Hewes, CPO Pete Skyrme and CRS JJ O'Neil were invaluable as the authors.

AFFILIATIONS

Shaaron Ainsworth
School of Psychology and Learning Sciences Research Institute
University of Nottingham, Nottingham, NG7 2RD, UK
Email sea@psychology.nottingham.ac.uk

Piers Fleming
School of Psychology, University of Nottingham, Nottingham, NG7 2RD, UK
Email pff@psychology.nottingham.ac.uk

REFERENCES

- Ainsworth, S. E., Grimshaw, S., & Underwood, J. (1999). Teachers implementing pedagogy through REDEEM. *Computers & Education*, 33(2-3), 171-187.
- Ainsworth, S. E., & Grimshaw, S. K. (2002). Are ITSs created with the REDEEM authoring tool more effective than "dumb" courseware? In S. A. Cerri & G. Gouardères & F. Paraguaçu (Eds.), *Proceedings of the 6th International Conference on Intelligent Tutoring Systems* (pp. 883-892). Berlin: Springer-Verlag.

TEACHERS AS INSTRUCTIONAL DESIGNERS

- Ainsworth, S. E., Williams, B., & Wood, D. (2001). Using the REDEEM ITS authoring environment in naval training, *Proceedings of the IEEE International Conference on Advanced Learning Technologies* (pp. 189-192).
- Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4-16.
- Cohen, P. A., Kulik, J. A., & Kulik, C. C. (1982). Educational outcomes of tutoring: A metaanalysis of findings. *American Educational Research Journal*, 19, 237-248.
- Cohen, P. R., & Howe, A. (1988). How evaluation guides AI research. *AI Magazine*, 9, 35-43.
- Graesser, A. C., Person, N. K., Harter, D., et al. (2001). Teaching Tactics and Dialog in AutoTutor. *International Journal of Artificial Intelligence in Education*, 12, 257-279.
- Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of AI in Education*, 8, 30-43.
- Hayes, M., Underwood, J.D., Ainsworth, S.E., & Grimshaw, S.K. (2001). Exploring pedagogy through a virtual class. Presented at *Computers and Learning 2001*, Warwick, April 2001.
- Major, N. (1995). Modelling Teaching Strategies. *Journal of AI in Education*, 6(2), 117-152.
- Mark, M., & Greer, J. E. (1995). The VCR tutor: Effective instruction for device operation. *The Journal of the Learning Sciences*, 4, 209-246.
- Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of AI in Education*, 10, 98-129.
- Murray, T. (2003). An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. In T. Murray & S. Blessing & S. E. Ainsworth (Eds.), *Tools for Advanced Technology Learning Environments*. (pp. 491-544). Amsterdam: Kluwer.
- Stenning, K., & Oberlander, J. (1995). A cognitive theory of graphical and linguistic reasoning: logic and implementation. *Cognitive Science*, 97-140.