ED 477 102 IR 021 830

AUTHOR	Thomas, Pete
TITLE	On the Effectiveness of a Programming Coach in a Distance Learning Environment.
PUB DATE	2002-06-00
NOTE	7p.; In: ED-MEDIA 2002 World Conference on Educational Multimedia, Hypermedia & Telecommunications. Proceedings (14th, Denver, Colorado, June 24-29, 2002); see IR 021 687.
AVAILABLE FROM	Association for the Advancement of Computing in Education (AACE), P.O. Box 3728, Norfolk, VA 23514. Tel: 757-623-7588; e-mail: info@aace.org; Web site: http://www.aace.org/DL/.
PUB TYPE	Reports - Evaluative (142) -- Speeches/Meeting Papers (150)
EDRS PRICE	EDRS Price MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Software Development; *Distance Education; Educational Technology; *Instructional Effectiveness; *Instructional Materials; Intelligent Tutoring Systems; Learning Strategies; Material Development; Student Behavior; Student Reaction

ABSTRACT

Previous research into the behavior of students while learning to program by automatically recording their actions has revealed that such recordings contain a wealth of information that can be collected together into a diagnostic tool that can support students' learning. The first step has been to construct the Coach, a software component that can be invoked on demand to provide a variety of support based on students' previous experiences. The Coach has undergone an initial stage of usability and usefulness testing to determine its effectiveness in practice. This paper describes the design of the Coach and reports on a small-scale experiment to assess its effectiveness with two groups of students. It was found that students did indeed turn to the Coach for help and that the control group also searched for help, but had to get it elsewhere. The paper also reports on other differences in behavior between the two groups. Includes seven figures and two tables. (Contains 17 references.) (Author)

On the Effectiveness of a Programming Coach in a Distance Learning Environment

Pete Thomas
Department of Computing
The Open University
Milton Keynes
UK
MK7 6AA
p.g.thomas@open.ac.uk

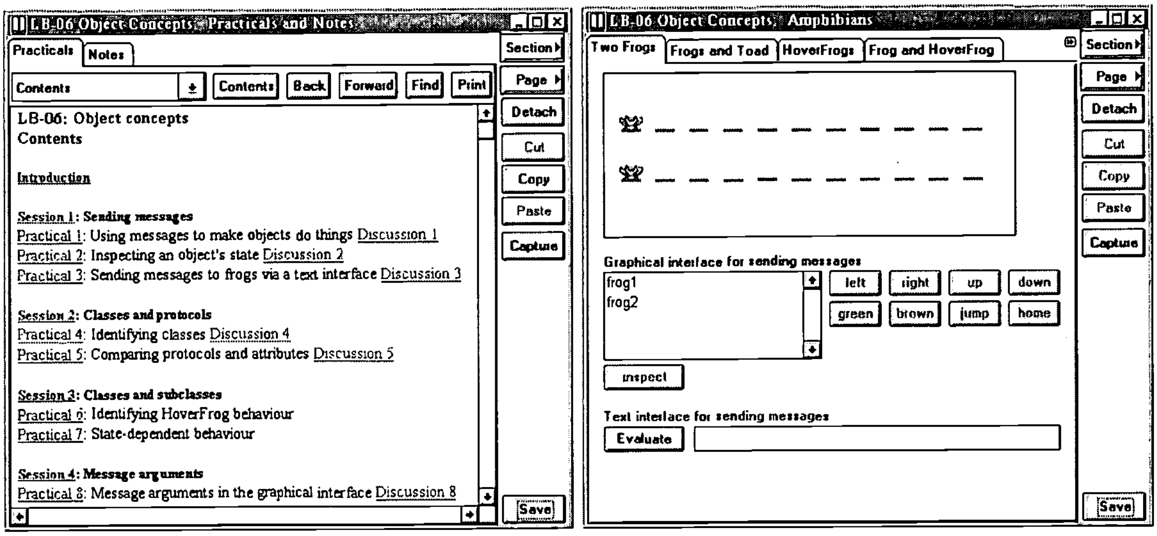
Abstract

Previous research into the behaviour of students while learning to program by automatically recording their actions has revealed that such recordings contains a wealth of information that can be collected together into a diagnostic tool that can support students' learning. The first step has been to construct the Coach, a software component that can be invoked on demand to provide a variety of support based on students' previous experiences. The Coach has undergone an initial stage of usability and usefulness testing to determine its effectiveness in practice. This paper describes the design of the Coach and reports on a small-scale experiment to assess its effectiveness with two groups of students. It was found that students did indeed turn to the Coach for help and that the control group also searched for help, but had to get it elsewhere. The paper also reports on other differences in behaviour between the two groups.

Introduction

Students on the Open University introductory course in computing M206, *Computing: An Object Oriented Approach* (M206 2000), are taught the OO paradigm using Smalltalk. The students, of which there are approximately 5000 per presentation, are taught at a distance. The teaching is supported by a series of practical activities performed with the LearningWorks system (Goldberg, Abell et al. 1997). The LearningWorks environment consists of over 30 LearningBooks, each one of which contains a set of practical activities. The structure of a LearningBook is based on the paradigm of a book in that the student can read through some pages of material that describe practical activities which have to be carried out in other pages of the book (Woodman, Griffiths et al. 1999). The early programming exercises ask students to interact with a series of micro-worlds in which they observe the effects of small portions of Smalltalk code; later exercises ask the students to construct their own code.

The left-hand side of Figure 1 shows the contents list of LearningBook 06, LB-06. All LearningBooks are divided into a sequence of sessions. A session is designed to be studied in one continuous interaction with the computer. Each session comprises a sequence of practicals, and each practical has an associated discussion page. Thus, students are encouraged to attempt a practical and then examine the discussion where the outcomes of the practical are examined.



LB-06: Object concepts
Contents

Introduction

Session 1: Sending messages
Practical 1: Using messages to make objects do things Discussion 1
Practical 2: Inspecting an object's state Discussion 2
Practical 3: Sending messages to frogs via a text interface Discussion 3

Session 2: Classes and protocols
Practical 4: Identifying classes Discussion 4
Practical 5: Comparing protocols and attributes Discussion 5

Session 3: Classes and subclasses
Practical 6: Identifying HoverFrog behaviour
Practical 7: State-dependent behaviour

Session 4: Message arguments
Practical 8: Message arguments in the graphical interface Discussion 8

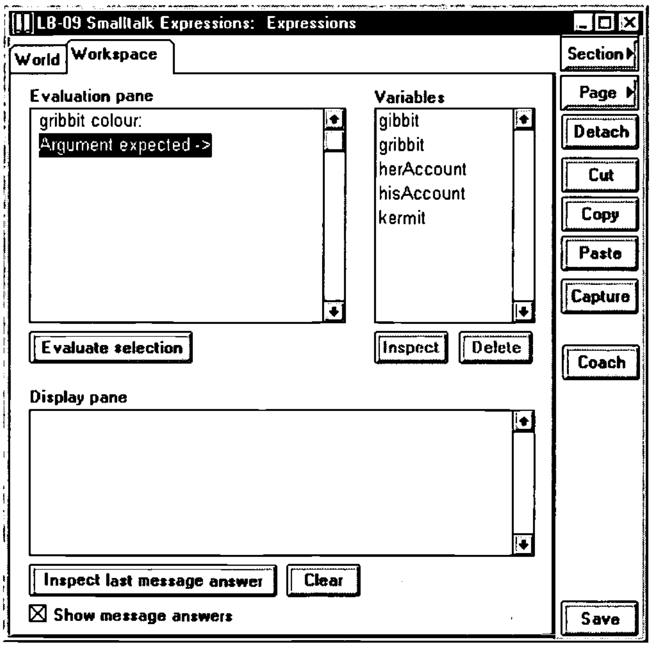
Figure 1 The interface of a typical LearningBook

On the right-hand side of Figure 1 is a page from LB-06 shown detached from the LearningBook containing the Amphibians micro-world. This micro-world is intended to support the learning of message sending. The Amphibians page also contains an evaluation text area in which students can enter small sections of Smalltalk code for the system to evaluate. In later LearningBooks, the idea of a workspace is introduced where students can enter significantly larger pieces of code and investigate their execution (see Figure 2).

Given this novel approach to the teaching of programming, we were interested to know how effective it would be. Therefore, we set up a significant research project (Thomas, Macgregor et al. 1998) to investigate how students learn to program in this environment. Our first step was to develop a student Observatory – an electronic system for recording the actions that students take when interacting with LearningBooks (AESOP 2001). An analysis of the recordings (Thomas and Paine 2000) showed a variety of student behaviours, particularly when dealing with errors (Logan and Thomas 2001). Therefore, we set about adapting the observatory software to provide additional support for error message comprehension and error correction – the Coach (Paine 2001). This approach opens up the question of how effective the Coach would be. There are two aspects to this question. First, how easy would students find the system to use and second, how useful would students find the system? The effectiveness of the Coach would be ascertained by examining the extent to which students used it to solve problems. In this paper we shall discuss the effectiveness of the Coach; the usability issues are discussed in (Thomas, Paine et al. 2000).

The Coach

Whenever the LearningWorks system detects an error in an item of Smalltalk code, it issues an error report. Figures 2 and 3 show two kinds of report. In Figure 2, the error report is shown highlighted. To obtain help with this error the student can invoke the Coach by clicking on the appropriate button. In Figure 3, the report appears in a modal dialog box that would normally be cleared by clicking on the OK button. However, in the modified system, the student has the option of invoking the Coach. The result of clicking on the Coach button is a new window similar to the one shown in Figure 4.



```
gribbit colour:
Argument expected ->
```

```
gibbit
gribbit
herAccount
hisAccount
kermit
```

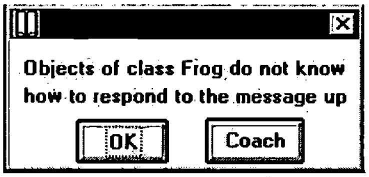
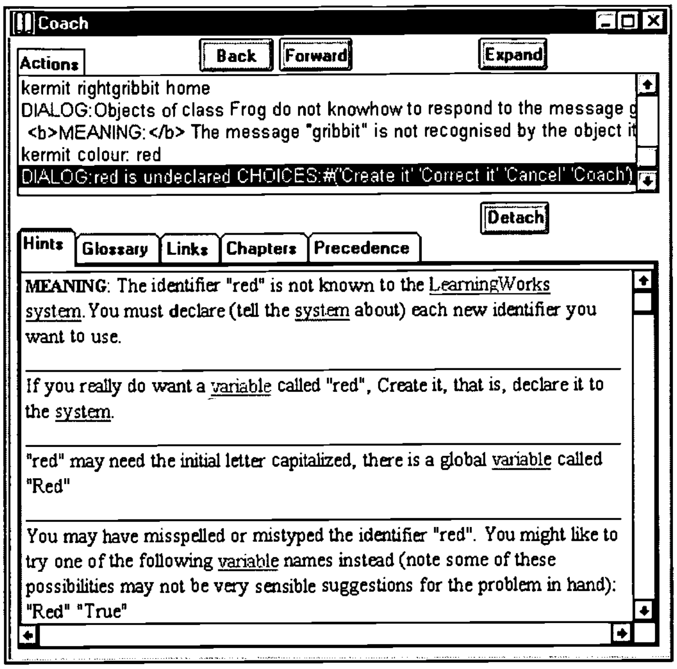

Objects of class Frog do not know how to respond to the message up

Figure 2 The LearningBook interface

Figure 3 A typical error report

The prototype Coach window has two main areas. At the top of the window is a tabbed area labelled 'Actions' which enables the student to scroll through a history of their actions. This uses the Observatory's recording of the student's actions throughout the LearningBook. The error report is shown highlighted and, in Figure 4, is a textual representation of the contents of a dialog box similar to the one shown in Figure 3 (the DIALOG is the error report and the CHOICES are the buttons that appear in the box). The second tabbed area that occupies most of the window shows a series of hints, the first of which is headed MEANING and contains an expanded explanation of the error that has been detected. The remaining items on the Hints page are possible reasons, given contextually, for the occurrence of the error. The hints contain hyperlinks to the Glossary containing definitions of the terms used in the descriptions. The remaining tabs give access to possibly useful materials such as the main teaching texts (Chapters), links to related web sites (Links) and a graphical Smalltalk syntax analyser (Precedence).



```
kermit rightgribbit home
DIALOG:Objects of class Frog do not knowhow to respond to the message g
 <b>MEANING:</b> The message "gribbit" is not recognised by the object it
kermit colour: red
DIALOG:red is undeclared CHOICES:#('Create it' 'Correct it' 'Cancel' 'Coach')
```

MEANING: The identifier "red" is not known to the LearningWorks system. You must declare (tell the system about) each new identifier you want to use.

If you really do want a variable called "red", Create it, that is, declare it to the system.

"red" may need the initial letter capitalized, there is a global variable called "Red"

You may have misspelled or mistyped the identifier "red". You might like to try one of the following variable names instead (note some of these possibilities may not be very sensible suggestions for the problem in hand): "Red" "True"

Figure 4 The Coach interface

Effectiveness experiment

In a small experiment designed primarily to investigate usability issues, we took the opportunity to study the effectiveness of the Coach. 14 student volunteers were provided with two additional LearningBooks, each containing a small number of practical exercises related to the work of LearningBooks 09 and 10. The practicals asked the student to evaluate a number of simple expressions, each of which resulted in an error, and to attempt to rectify the errors. Those with the Coach were told of its purpose, but were not required to use it to solve the problems.

The experiment used an independent samples design to compare the actions of students who used the Coach with those who did not (Siemer and Angelides 1998; Budgen and Pohthong 1999). Students were divided into two groups, providing us with 2 conditions. In condition 1, students were provided with the additional LearningBooks, the AESOP Recorder and Coach software. In condition 2, students were given the additional LearningBooks and the AESOP Recorder software (i.e., they did not receive the Coach software). Students were assigned to a condition on the basis of a pre-experiment questionnaire aimed at controlling variables related to gender and age. This resulted in 8 students being assigned to condition 1 group and 6 to condition 2 group. Having completed the test exercises, the students e-mailed their recording to us for analysis.

Post-Questionnaire

Once they had finished the tests, all students were asked to complete a questionnaire designed to assess usability and usefulness issues. When asked to rate the ease of completing the tests on a scale of 1 (difficult) to 5 (easy), on average students without the Coach rated the practicals as easier (4.4) than students with the Coach (3.75). Generally, students who used the Coach found the amount of information on its interface slightly distracting. When asked about the likelihood of using the Coach in other LearningBooks on a scale of 1 (unlikely) to 5 (likely), on average they said 3.25.

The following comment from one student summarises experiences with the Coach: "The information in the Coach was useful, although it was a little difficult to home in on the appropriate comments for the problems I had. When I went back to using LearningWorks I found myself looking for it in my next LB and on a couple of occasions wishing it was there. Sometimes the Smalltalk error messages are difficult to interpret and I think the Coach would help."

Analyses of LB Test-09 Coach Recordings

Of the 8 students with the Coach, 7 used it 'for real'. One student did not open the Coach at any point during either test LearningBook. All 8 students attempted all of the practicals. Figure 5 shows the percentage of students who used the Coach on each practical activity: it only includes those students who used the Coach to help them attempt to solve the practical i.e. not those who simply opened the Coach to look at it. It shows that, for each practical, some students felt the need to use the Coach.

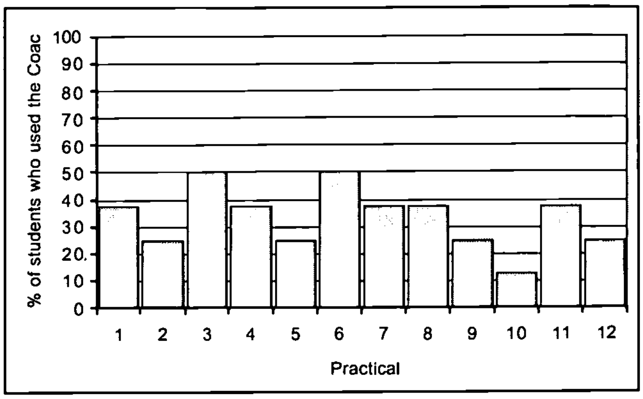


Figure 5 The percentage of students who used the Coach in each practical

With the exception of 1 student in one practical, *all* students with the Coach who solved the problem in a practical did so *without* looking at the discussion pages. However those students without the Coach who solved a practical sometimes looked at the discussion page before the solving the problem, as shown in Table 1. A few students without the Coach software also looked at the discussion page but did not solve the practical. We concluded that some students in each group needed additional support to solve the problems.

	No. of students who:	
Practical	**Solved practical**	**Looked at Discussion page**
1	6	1
2	6	0
3	5	1
5	5	2
8	6	2
10	6	1
11	6	5
12	5	0

Table 1: Students without the Coach who looked at the discussion before solving each practical

Figure 6 shows the percentage of students who correctly solved the problems that had a single specific solution. There were students in each group who failed to solve some problems. In three out of these four cases, more students without the Coach succeeded in solving the problems. Nevertheless, in practicals 1, 8 and 11, some students without the Coach tried the practicals only after having read the discussion, so one might conjecture that not all students in this group would have successfully completed these exercises. However, the results of the post-questionnaire indicates that the group with the Coach were weaker. This is confirmed by the number of student errors (other than those mandated by the practical activities) in which those without the Coach made on average 18.83 errors each, whereas those with the Coach made 23 errors each on average.

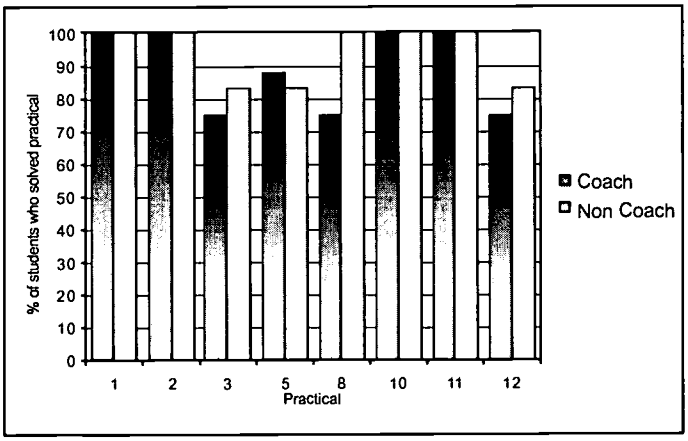


Figure 6 The percentage of students with and without the Coach who solved each problem

A summary of the data extracted from the recordings for LB Test-09 from students with and without the Coach software is shown in Table 2.

		Average number of:				
	Time spent	Open	Close	Hyperlinks	Evaluations	Dialogs / Notifiers
With	48 mins	2.63	9.75	39.25	35.75	35.00
Without	20 mins	1.33	4.50	22.50	32.83	30.83

Table 2 A summary of LB Test-09 recordings.

From Table 2 it can be seen that students with the Coach spent, on average, over twice as long in LearningBook Test-09 than students without the Coach.

Students with the Coach appear to do LB Test-09 in more sittings that student without the Coach (shown by the higher number of 'Open' events). Students with the Coach also on average accessed more hyperlinks than students without the Coach. This is to be expected as students with the Coach have access to the Coach Links, Chapters and Glossary pages which all contain a number of hyperlinks.

We analysed the results of the second test (LB Test-10) in a similar way. The difference between the two groups of students was less marked particularly in the amount of time spent solving the problems. This is to be expected as students get used to using the Coach software. This gives us confidence that using the Coach need not be a significant overhead, especially when it is clear that some students look for additional support to solve some problems.

Future work

We have implemented a revised Coach interface and slightly amended the two test LearningBooks so that we can repeat this experiment during the 2002 presentation of the course but with a much larger sample of students.

The next major step in the development process is to utilize the data contained in the student recordings to improve the feedback given by the Coach based on actual student experiences. Figure 7 shows the architecture of the system as envisaged. The basic idea is that the Coach obtains its data from a database on a student's machine (downloaded with the Coach). The local database is augmented with additional data based on the student's interactions with the LearningBooks (using the Events Analszer component). The recordings are sent to a central recordings database as with the present system. The recordings will contain all the events that occurred in each LearningBook, including those related to the use of the Coach since it is a LearningBook, too. The complete set of all students' recordings are analysed to update a central database of Coach data. Students will be able to update their local database with revised Coach data as and when they wish. In this way we hope to be able to adapt the Coach in the light of all students' experiences.

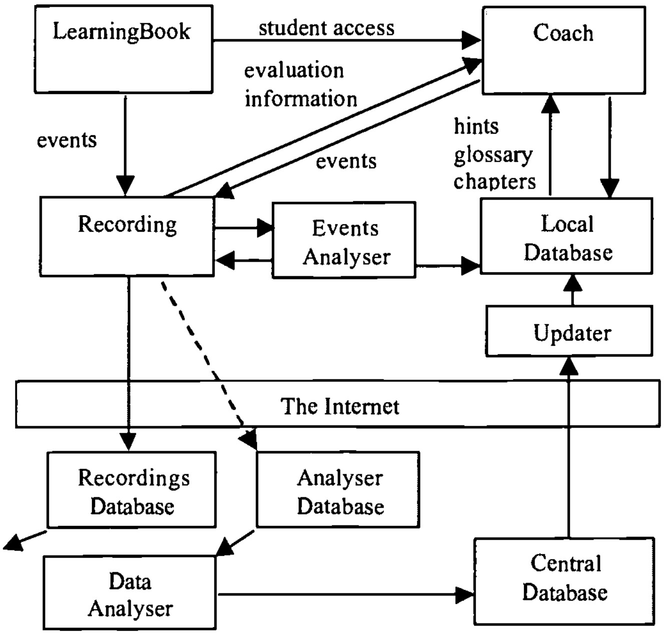


Figure 7 The Coach System

A third avenue for exploration is to follow up on a discussion of models of intelligent tutoring in(Gertner, A. & VanLehn, K (2000)) by using the Coach to provide a model of how a given problem should be solved. This will be based on work in another area [Thomas, 2001] for which we are developing mechanisms for specifying tasks and identifying attempts at solving them. We want to see how these mechanisms might be used to create a model of how a problem is to be solved and to detect student attempts at the solution. As the student tries to solve the problem, his/her actions are compared to those that the model would make. If the student's actions diverge sufficiently from the model, the Coach would offer the student some advice or feedback.

Conclusions

Overall, the majority of students who used the Coach found it useful. However, the prototype interface was found to be distracting and students found it difficult to home in on the appropriate hint. We obtained useful feedback on the interface and have simplified it. On balance, the Coach seems to offer a beneficial tool that some students found attractive.

We believe that the Coach is effective and that it is worth investing further effort to improve it. In particular, we can see ways of adapting the Coach to individual student needs. We also believe that it will be possble to provide further help through the idea of a model solution and comparing it with student attempts at solving programming problems.

References

AESOP (2001). http://mcs.open.ac.uk/aesop

Budgen, D. and A. Pohthong (1999). Component Reuse in Software Design: An Observational Study. Software Technology and Engineering Practice (STEP), Pittsburgh, PA.

Biggs, J. (1999). Teaching for quality learning at university: what the student does. Buckingham: Open University Press.

Gertner, A. & VanLehn, K (2000). Andes: A Coached Problem Solving Environment for Physics. (pp. 131-142) In Gauthier, G., Frasson, C. & VanLehn, K. (Eds) *Intelligent Tutoring Systems: 5th International Conference, ITS 2000*, Berlin: Springer.

Goldberg, A., S. Abell, et al. (1997). The LearningWorks Delivery and Development Framework. CACM **40**(10): 78-81.

Goodyear, P (1999). Environments for lifelong learning: ergonomics, architecture and the practice of educational technology. CSALT Document 99111. http://csalt.lancs.ac.uk/jisc

Kivi, M.R., Gronfors, T. and Koponen, A. (1998). MOTHER: System for continuous capturing of display stream. Behaviour and Information technology, 17 (3) pp152-154.

Logan, K. and P. G. Thomas (2001). Observations of student working practices in an online distance education learning environment in relation to time. 13th Annual Workshop of the Psychology of Programming Interest Group, Bournemouth, UK.

M206 (2000). Computing: An object-oriented approach. Milton Keynes, UK, The Open University Press.

MacGregor, M., Thomas, P.G. & Woodman, M. (1999). Recording and Analysing User Actions in a Smalltalk Programming Environment, Proceedings of Technology of Object-Oriented Languages and Systems, Santa Barbara, USA, August 1999.

Paine, C. (2001). The Coach - Supporting students in the area of error reports. 13th Annual Workshop of the Psychology of Programming Interest Group, Bournemouth, UK.

Siemer, J. and M. C. Angelides (1998). A Comprehensive Method for the Evaluation of Complete Intelligent Tutoring Systems. Decision Support Systems **22**: 85-102.

Thomas, P.G. and C. Paine (2002). Monitoring distance education students' practical programming activities, Educational Technology & Society, special issue, to appear July 2002. http://ifets.ieee.org/periodical

Thomas, P.G. (2001). The Coach: Supporting students as they learn to program. Proceedings of the 6th Annual Conference on Innovation and technology in Computer Science Education. ITiCSE (2001), Canterbury, UK, June 2001.

Thomas, P. G. and C. Paine (2000). Tools for Observing Study Behaviour. 12th Annual Workshop of the Psychology of Programming Interest Group Calabro, Italy.

Thomas, P. G., C. Paine, et al. (2000). The Coach: Overview. Computing Department Research Report 2000/07, Open University.

Woodman, M., R. Griffiths, et al. (1999). Exploiting Smalltalk Modules In A Customisable Programming Environment. *Proceedings of ICSE 21, International Conference on Software Engineering*, Los Angeles.