DIVISION OF COMPUTER SCIENCE

COMPUTER SCIENCE AND PROBLEM SOLVING: ENCOURAGING SELF-DIRECTED LEARNING WITH LARGE CLASSES

John Sapsford-Francis
Carol Britton
Juliet Brown

Technical Report No.168

August 1993

Computer Science and Problem Solving: encouraging self-directed learning with large classes.

John Sapsford-Francis, Carol Britton and Juliet Brown

Division of Computer Science,
University of Hertfordshire,
College Lane,
Hatfield AL10 9AB.

email: comqjs@uk.ac.herts, comqcb@uk.ac.herts, comqjab@uk.ac.herts

Introduction

Our current students will be professionals during a period of massive change. As Engel (1991) points out, for such students self-directed learning is likely to be a necessity. Higher education is itself in a state of flux. The development of less labour intensive educational methods has become a priority as student numbers rise. In this paper we describe a problem solving course which has been devised and run in the Division of Computer Science at the University of Hertfordshire. The course enables Computer Science degree students to develop a range of skills and to explore, understand and apply a range of concepts, that underly the discipline of Computer Science.

The course is student centred and activity based encouraging independent self-supporting learning strategies. Students work on a carefully selected range of problems that provide intensive information-rich opportunities to develop their skills and understanding.

Such student centred courses are found in many academic disciplines, however they are still rare in the teaching of Computer Science and the application of such approaches to the education of large numbers of students is very seldon attempted. This paper describes the adaptation of a student-centred learning approach to the needs of 185 first year undergraduate students. The main concerns in the design of the new course were:

- the provision of a problem solving folder
- the planning of student time both in contact with staff and on their own
- the development of information-rich learning activities
- the provision of a skeleton "Problem Solving Toolkit".
- the provision of adequate information resources

The problem solving folder

At the start of the course all students were provided with a problem solving folder. This folder consisted of four main sections that will be referred to throughout this paper:

Section 1: "About the course" introduced the course and its objectives and explained how we expected students to work on the course and develop their problem solving expertise. It also contained the week by week timetables (see the section on the planning of student time).

Section 2: "Your toolkit" contained material on objective setting for the course, an "attitudes to problem solving questionnaire" that helped students diagnose their main learning requirements for the Problem Solving course, a section on reference models of problem solving and useful rules of thumb and two skeleton sections for techniques and representations for problem solving.

Section 3: "Reference material" contained papers from a number of areas within computer science and generic problem solving.

Section 4: "Task specification and problem sheets" was a storage area for tasks and problems that students had completed on the course.

The planning of student time

Each student attended a two hour activity session, run for 30 students, once a fortnight, and, on average through the year, attended one one hour lecture per week. Each student was also expected to undertake a programme of reading and practical tasks outside class time; many but not all of these tasks required students to work in groups.

As we were expecting students to put in about two-thirds of their total effort on the course outside class time, and also expecting them to co-ordinate a complex mixture of reading and tasks, we provided a very detailed week by week timetable. The week by week timetable was kept in section 1 of the problem solving folder. It included the dates and times of all classroom sessions, and the start and end dates and approximate time estimates for each task we required them to do outside class.

The development of information-rich learning activities

As described above the course was organised around a number of key activity sessions. Tasks undertaken in the two hour activity sessions may require preparatory or follow-up work, or both, and are typically supported by some reading, one or more lectures, and one or more other tasks set for students to do outside class time.

In the 1992/3 session the "Speaking Wristwatch" problem was one such key activity. The students, in groups of four or five, were asked to specify a speech interface to a wristwatch. Skills required, and we hope developed, included the ability to interpret a fairly open problem brief, to abstract and decompose problems, and to choose and use appropriate representations in documenting problems and proposed solutions. During the activity session they were interrupted for two minutes at ten minute intervals, and used this time to construct an "interrupted protocol" of their problem solving process. After the session, they had to produce a report which defined the problem as they interpreted it, gave the criteria against which they would evaluate their solution, documented the solution, and evaluated the solution against their criteria.

The "Speaking Wristwatch" activity was supported in advance by a lecture and a task to be done outside class. The lecture was about protocol creation and analysis in general, and some of its applications in computer science. The support task required them to construct of a concurrent protocol for a simple problem.

The activity was supported by further follow-up tasks. The reports the students produced were handed in, and a selection were duplicated and distributed to each group. They then had to draw up a list of criteria for a good **report**, and evaluate each of the duplicated reports against their criteria. Once they had done this we distributed our own evaluations. The students' next task was to compare their criteria, and their rating of the reports, with ours. To follow up and consolidate what they had learnt about protocol analysis, their final task was to add a "technique summary" for this to their toolkit folder.

This activity sequence illustrates many of the main elements of the problem solving course: provision of access to sufficient knowledge to enable students to carry out an activity; a reasonably challenging group activity that requires students to explore their growing expertise; a means for students to monitor their problem solving performance; an opportunity for students to reflect on their performance and consider objectives for future problem solving activities.

The provision of a skeleton "Problem Solving Toolkit"

The problem solving toolkit was the second main section of the problem solving folder. The toolkit that was supplied was in skeleton form only. From the very beginning of the course all the students were encouraged to customize this skelton toolkit and eventually to build up their own version of it. This toolkit was to be constructed from the students' experience of and observations on solving a variety of problems. Throughout the course, it would be applied, tested and gradually refined. (This pattern of learning corresponds to Kolb's (1984) experiential learning cycle).

One of the aims of the course is to enable the students to build-up useful problem solving knowledge abstractions. Gick and Holyoak (1983) see the formation of such problem solving schemata as occuring through the successful mapping of one problem solution onto another. We sought to draw students' attention to similarities between problems and to enable students to generalise their learning, by striving to develop their own abstractions. The problem solving toolkit was intended to act as a focus for these abstractions.

An important step in enabling the students to build their own problem solving toolkit is to provide a broad framework for their thinking about the process of problem solving. One such framework, that we made available to students, was Bransford and Stein's (1984) Ideal problem solving model.

From the start of the course, then, the students were introduced to the concept of a problem solving toolkit. This comprised a set of personal objectives, problem solving concepts, representations, techniques and case studies intended to support them in their problem solving activities. The toolkit included material on approaches to such varied problem solving activities as problem perception, information gathering, problem representation, objective setting, developing new ideas, hypothesis testing , time management and group dynamics.

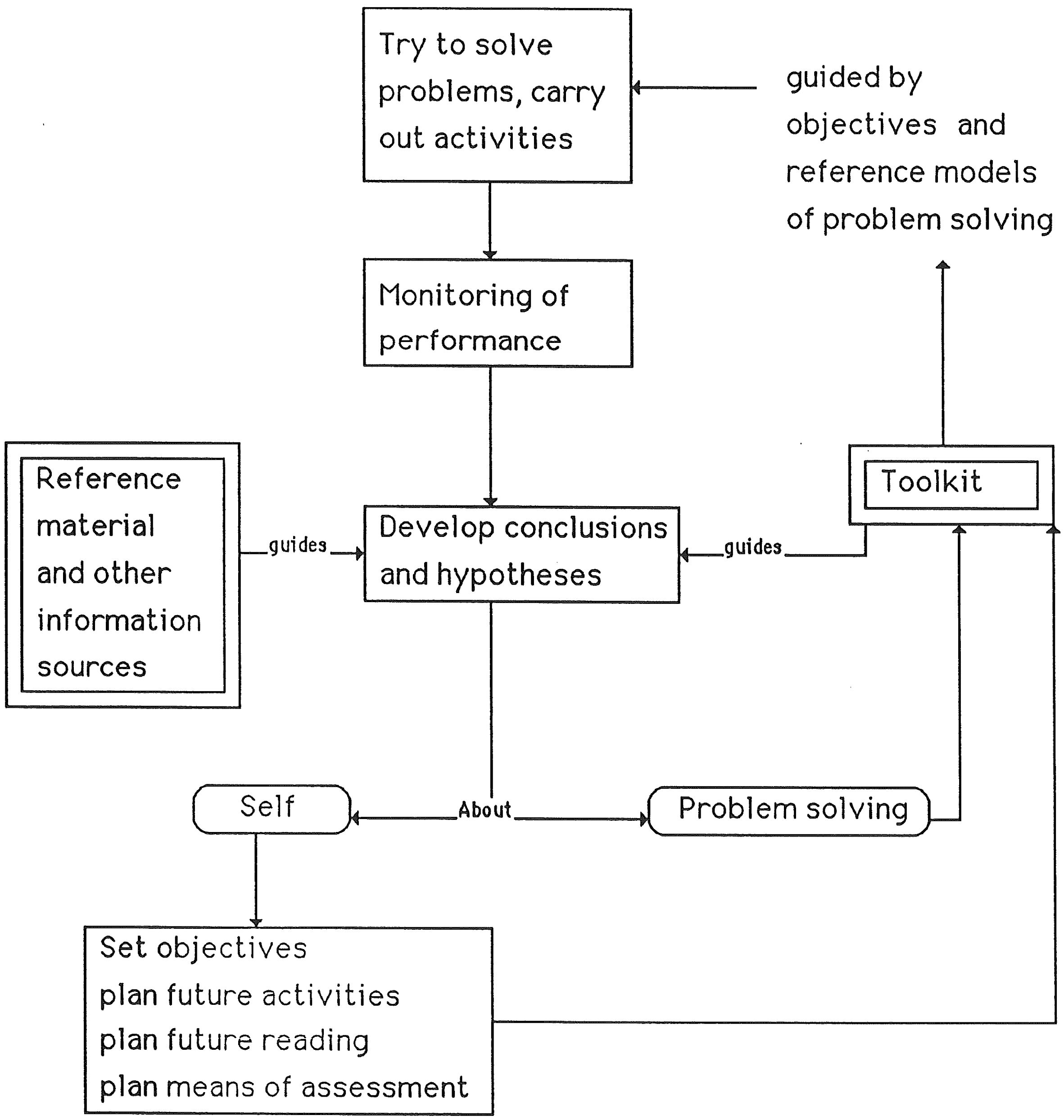

Figure 1 How the toolkit supports experiential learning

The way that the toolkit is used in the course is shown diagramatically in figure 1 above.

Performance on problems and monitoring and reflection on such performance are all informed by the skeleton toolkit. Learning from each of these processes is made concrete through customisation of the toolkit. Sudents are encouraged to see the problem solving course as an opportunity for building and refining their own problem solving toolkits.

An activity centred session typically follows the pattern described by the figure and notes below.

- Problem solving activity . This is usually carried out in groups and involves some kind of observation, either by the problem solvers themselves or by an external observer.

- Monitoring of performance. This takes the form of a feedback session and general discussion. Questions covered may include:
  What have we learned from this activity?
  How can we usefully apply this knowledge?
  How well did we manage the problem solving process?
  What goals could we set for future problem solving activities?
  What implications are there for our problem solving model?
  What is the relevance of this to Computer Science?

- Develop conclusions and hypotheses. Students are always expected to reflect on the performance of the activity and the following discussions. Students' conclusions should be based not only on their experience of the problem solving activity, but also on relevant reference material and other sources of information. The conclusions and hypotheses drawn from the activity, discussion contribute to the enhancement of the student's own problem solving ability and his or her toolkit of relevant techniques and skills.

- On the basis of what has been learnt from the process, the students are encouraged to set their own objectives for problem solving, to plan their activities and reading, and to consider how all this can be assessed, monitored and controlled. In this way, the students are gradually taught to manage their own learning.

Preliminary evaluation of the course indicates that these innovations have substantially mitigated the negative effects of increased student numbers. As the problem solving course already incorporates some parts of the traditional computer science syllabus (eg. "structured techniques" such as data flow diagramming, entity relationship modelling and simple systems analysis) it is apparent that this approach can effectively be applied to more specific subject areas within Computer Science.

The provision of adequate information resources

Directed reading forms an important part of the Problem Solving course. Section 3 of the problem solving folder is a reference section containing articles, papers and other materials that have been carefully selected from a wide range of disciplines to help students make connections and learn about fundamental concepts and activities in problems solving.

At present the reference section contains the following material:

The Design Philosophy Behind Motorola's MC68000. Starnes, T.W. (1983)
A model for improving problem solving. Bransford, J.D. and Stein, B.S. (1984)
Critical Thinking Skills. Ennis, R.H. (1987)
The problem with Software. Ince, D. (1988)

Systems and their Models
Scientific Method
Knowledge Performance and Dysfunction
Opening Up and Closing down (All written by the course team)

Students are directed to read, apply and answer questions on these reference materials as part of the week by week timetable.

Conclusions: Problems, Pragmatics and Lessons

Attempting a student centred approach with so many students is not without its problems. To gather information about these problems we have monitored student attitudes and performance on the course in a number of ways:

- through student performance on both assessed and unassessed course activities
- through the completion of student attitude questionnaires
- through informal feedback from students either directly or via their personal tutors

The feedback we have gathered suggests some student concern about coping with self managed study, feedback on their performance on the course and difficulty in transferring their learning on the course to other areas. I was also clear that the weaker students did little work on building up their toolkits.

Self managed study

A large number of the students seem to expect and want passive learning. This is an enduring and persistent mind set that our students seem to have. They expect to attend lectures and to receive detailed feedback on their progress. They do not expect to have to manage their own learning and find their own feedback. Yet this is our ultimate target. To adress the problem: we approach this goal, gradually, trying to show them the value of autonomous learning and self-evaluation. We also encourage students to plan, monitor and control their own route through the problem solving course. The problem solving toolkit provides essential support and guidance while the students are learning to do this.

The provision of adequate feedback

At this stage in the students learning feedback from the course team is extremely valuable. By concentrating our efforts on providing a comprehensive information resource (the problem solving folder) and on intensive fortnightly activity sessions we have been able to increase the amount of effort put into the provision of feedback. For the 185 students on the course there were the following opportunities for feedback:

diagnostic writing assignment at the begining of the course
report writing exercise on the speaking wristwatch activity (see above)
several voluntary essays
a mock practical exam with group report and individual essays (this occurred mid year)
the practical exam with group report and individual essays (which occurred at the begining of the summer term)

In addition ample opportunities for individual discussion were provided in the summer term.

Transfer of learning on the course

Getting students to transfer their learning to other situations is often difficult. We encourage students to do this by setting a variety of activities where students look at problems in their own lives or on other courses and then present an analysis of them. Problems which the students are set during the course cover a wide range of activities: from library research to code breaking and from project planning to designing a new pathway layout for the university. Many researchers (eg. Tulving et al. 1964, Morton 1964, Meyer 1975, all in the area of word recognition) have emphasised the importance of context as a memory retrieval cue. By providing a multiplicity of problem solving contexts we provide a rich set of cues to the students' abstract knowledge of problem solving.

Encouraging weaker students to build up their toolkits

It is possible for students to gain quite a lot from the course without developing their problem solving toolkits. This may explain the reluctance, on the part of the weaker students, to involve themselves in this part of the course. From looking at the work of the better students, however, it is clear that building up their own toolkit was a valuable course activity. Therefore we will encourage work on the toolkit by making references to it, in the week by week timetables, much clearer and more frequently. We will also make it clear that we expect this work to be done by referring back to it at various points throughout the course.

Rules of thumb for a successful student centred course

The problem solving course has been designed to meet clear educational needs. It has been built on sound educational and psychological theory. The design and running of the course has presented us with a serious challenge, which we feel we have met. The effect of the course on students' problem solving and management skills is hard to measure objectively, although attitude surveys and student performance measures have yielded favourable results.

Certain factors have been critical in achieving this success:

- **detailed planning of students' time.** With comparatively little contact with students one cannot rely on traditional staff-student contacts to motivate the students and to drive the course. It must be absolutely clear to the students what they are expected to do, when they are expected to do it and how long it should take. The best way that we have found to convey this information is to provide a very detailed programme as described in the section on the planning of student time (above).

- **attention should be paid to group dynamics.** All activities on the problem solving course are carried out in groups, usually made up of between three and five students. The composition of the groups may be changed for each activity, so that the students do not get used to working with a particular team of people. They constantly have to adapt to new group situations. The importance of group work, both in Computer Science and in management, is stressed throughout the course and individual groups are monitored by staff during activities to identify any problems.

- **designing highly interconnected learning experiences.** If learning experiences are part of a unified whole there is a better chance that students will see their relevance and that they will be motivated to participate in them. The sequence of lectures, readings, tasks and activities surrounding the "Speaking Wristwatch activity" described above is a good example of this principle. We assert that although such sequences are hard to plan, and somewhat prone to damage if adequate redundancy is not built into the programme, they are perceived as important by students, they tend to provide strong motivation for students and they are more likely to result in transferrable learning.

- **activities should be concrete, where possible, and relevant to the intended area of application.** It is our experience that students find activities that involve the direct perception and manipulation of real world objects (whether those objects are lego bricks, eggs, balloons, or pathways around the University) particularly motivating and significant. There seems to be an especially valuable kind of learning that derives from direct contact with a problem that cannot be found in a theoretical treatment of the same problem.

- **include self monitoring and reflection in problem solving activities, but also allow "fluent" performance.** It is important that students should be able to observe and report on group problem solving behaviours. They should also be able to listen to and take on board observation of their own behaviours. This is one of the main means we have of focusing students' attention on particular types of behaviour in problem solving. Almost all of the course activities involve some form of observation. While the activity is being carried out, observation may be achieved by appointing a member of each group to act as an external observer, or by means of interrupting the activity so that the students can record precisely what they were doing at that moment. After the activity is over, retrospective self-observation is promoted during discussion and feedback sessions. The students are also expected to show an awareness of their own problem solving behaviour in their written reports. These kinds of observation act as forcing functions (Norman

1991), to some extent disrupting students' automatic problem solving behaviour and forcing them to pay attention to the way they habitually deal with problems. Awareness of habitual behaviour is the begining of change. However excessive observation and the intrusion of self-reflection on the problem solving process can be extremely frustrating for students. At times it is important to allow them to try out their developing problem solving abilities in settings that are free of observation and introspection.

References

Bransford, J.D. and Stein, B.S. 1984 The IDEAL problem solver. New York: W.H. Freeman and Company

Engel, C.E. 1991 Not Just a Method But a Way of Living, in Boud, D. and Feletti, G. (Eds) 1991, the Challenge of Problem Based Learning. Kogan Page.

Ennis, R.H. A Taxonomy of Critical Thinking Dispositions and Abilities in Baron, J.B. & Sternberg, R.J. (1987) Eds. Teaching Thinking Skills: Theory and Practice. WH Freeman and Co.

Gick, M.L. and Holyoak, K.J. 1983 Schema Induction in Analogical Transfer, Cognitive Psychology, 15, 1-38

Ince, D. 1988, Software Development: Fashioning the Baroque. Oxford Science Publications.

Kolb, D.A. 1984 Experiential Learning: Experience as the source of learning and Development, Englewood Cliffs, New Jersey: Prentice Hall

Meyer , D.E., Schvaneveldt, R.W. and Ruddy, M.G. 1975, Loci of contextual effects on visual word recognition, in Rabbitt P.M.A.1and Dornic, S. (1975) Attention and Performance V, Academic Press, New York.

Morton , J. 1964, The effects of context on the visual duration threshold for words, British Journal of Psychology, 55, pp 165-80

Norman D.A.1991 Cognitive Artifacts , in Carroll, J.M. (Ed.) 1991 Designing Interaction: Psychology at the Human Computer Interface, Cambridge University Press.

Starnes, T.W. 1983 Design Philosophy Behind Motorola's MC68000. BYTE. April 1983

Tulving, E., Mandler, G. and Baumal, R. (1964) Interaction of two sources of information in tachistoscopic word recognition, Canadian Journal of Psychology, 18,pp 62-71

Appendix 1

University of Hertfordshire Problem Solving Course: Aims and Objectives

Aims

- To introduce general problem solving techniques, and show how there are essential to Computer Science
- To introduce particular intellectual tools (eg methods and notations) that are commonly used by computer scientists for solving particular classes of problems.
- To discuss the concept of a problem and its relation to domain specific knowledge.
- To provide experience of working in groups, explore the dynamics of groups interaction and develop appropriate communication skills.

Objectives:

On completion of the course, the student will, at an appropriate level, be able to:

- Identify sensible boundaries for simple problems.
- Analyse situations to extract a precise formulation of a problem.
- Utilise a variety of techniques for expressing problems and their solutions.
- Utilise a variety of techniques for planning, monitoring and controlling the management of the problem solving process.
- Discuss the group and cognitive factors that facilitate and inhibit problem solving and their applications in general problem solving and management of the problem solving process.
- Participate fully, and constructively, in team problem solving.
- Elicit and model user requirements for simple systems by applying appropriate communication skills and modelling techniques.
- Contibute to technical documentation and presentations relating to a suitable simple system.
- Show an awareness of the existence of alternative strategies for the management of the problem solving process, and discuss, at an appropriate level, the assumptions upon which these are based.