

DIVISION OF COMPUTER SCIENCE

**Applying Techniques from Foreign Language Teaching to
Computer Science**

**Amanda Derrick
Carol Britton
John Sapsford-Francis**

Technical Report No.164

September 1993

Applying Techniques from Foreign Language Teaching to Computer Science.

Amanda Derrick, Carol Britton and John Sapsford-Francis.

Division of Computer Science,

University of Hertfordshire,

College Lane,

Hatfield, AL10 9AB.

email: comqjd@uk.ac.herts, comqcb@uk.ac.herts

Keywords : group activity, exchange of information, role play, peer feedback

Introduction

As increasingly large numbers of students enter higher education, it is necessary to rethink teaching strategies and the allocation of resources to cope with class sizes that are well over the three figure mark. In the Division of Computer Science at the University of Hertfordshire we have been investigating methods and techniques that are used to teach foreign languages to school children in order to see if these can be adapted to teach computer languages to students. This paper first lists the criteria that we established as essential for a useful teaching technique, and then describes our experiences in using a variation of Information Gap (a technique used in schools for teaching French) to teach the Z specification language to students.

Qualities of a useful technique for teaching computing languages

In examining techniques used in school teaching we were looking for those that satisfied all or most of the following criteria.

The technique should :

- be suitable for large class sizes
- make efficient use of resources
- provide quick and comprehensive feedback on student performance
- motivate and interest students
- develop student confidence in communicating by means of a formal language, Z
- encourage learning at different levels of expertise. This criterion arose from a study of the work of Black et al.(1988) .

Information Gap

Information Gap is a technique that is widely used in the teaching of foreign languages to school children. It involves working in small groups, where each member is given a partial set of information. The complete picture is obtained through interaction and exchange of information.

In adapting the Information Gap technique to teach the Z formal language we initially developed a specification of a library system which we then divided into two. One part of the

library specification contained mathematical Z schemas describing the membership of the library and certain system operations. These schemas were supported by explanatory narrative in English. The second part of the specification described the state of the library loans and the remaining system operations, also with English narrative.

In our experiments most of the students were in groups of two, although with weaker students groups of 3 or 4 might be preferable. When the students are working in pairs, one student has a partial specification consisting of Z schemas for certain operations and a description in English for the others. The second student has the matching incomplete specification. Each student has the task of completing their specification by writing the missing Z schemas. The partner student, who has the relevant schemas, takes on the role of tutor, answering questions and explaining difficult points. Once a schema has been completed the roles are reversed. Examples of the material given to the students are shown below. Initially we show the complementary parts of the specification that describe the data in the library system, followed by the parts that define operations on the data.

Student 1 - The Library Specification

You are going to write a very simple specification for a library system, working in pairs. You are provided with a partly written Z specification, the rest of the system being described in natural language. Your partner will have the complementary specification. You are required to build up the complete specification by writing the Z schemas for the natural language parts. Your partner (who has the Z) will help you.

The basic types in this specification are: **[Reader, Book-Id, Title]**
It has been decided to split the state schema into two schemas.

Static StateSchema

The static state schema records that the **stock** held is a set of Book-ids, that **book** is a function from Book-id to Title and that **members** of the library are a set of Readers. The static library state schema is shown below.

Static Library	
stock:	\mathbb{P} Book-Id
book:	Book-Id \rightarrow Title
members:	\mathbb{P} Reader
dom book = stock	

Dynamic State Schema

The second part of the state schema is the dynamic library. This records information about books which are out on loan and which member has borrowed the book. The books out **on-loan** are a set of Book-Ids, the books **on-shelf** are a set of Book-Ids and the function **loans** records the Book-id loaned to the Reader.

The predicates state that

- i) a reader may not take out more books than the maximum number of loans permitted
- ii) a book cannot both be out **on-loan** and held in the library
- iii) there is a relationship between **loans** and **on-loan**.

Write the dynamic state schema for the library system.
Remember to include the state invariant(s).

Student 2 - The Library Specification

(Introductory information for student 2 is the same as that for student 1)

Static State Schema

It has been decided to split the state schema into two schemas. The static state schema records that the **stock** held is a set of Book-ids, that **book** is a function from Book-id to Title and that **members** of the library are a set of Readers.

Write the static state schema. Remember to include the state invariant(s). You should consider the relationship between **stock** and **book**.

Dynamic State Schema

The second part of the state schema is the dynamic library. This records information about books which are out on loan and which member has borrowed which book. The books out **on-loan** are a set of Book-Ids, the books **on-shelf** are a set of Book-Ids and the function **loans** records the Book-id loaned to the Reader. The predicates state that :

- i) no reader may take out more books than the maximum number of loans permitted
- ii) a book cannot both be out on-loan and held in the library
- iii) there is a relationship between **loans** and **on-loan**.

The dynamic state schema is shown below

Dynamic Library
on-loan: \mathbb{P} Book-Id
on-shelf : \mathbb{P} Book-Id
loans: Book-Id \rightarrow Reader
on-shelf \cap on-loan = $\{\}$
dom loans = on-loan
$\forall r : \text{Reader} *$
$\# (\text{loans} \triangleright \{r\}) \leq \text{max-loans}$

The state of the library system has been split into the static and dynamic library schemas requiring each student in turn to attempt the declarations and state invariants required. In the class exercise the two schemas are then combined to produce the overall library schema.

Operations on the system

Partial specification for student 1

The AddBook operation

The operation to add a book to the system will update the static library system. A Book-id and a Title are inputs to the system. The schema AddBook is shown overleaf.

AddBook
Δ Static Library
\exists Dynamic Library
$b? : \text{Book-Id}$
$t? : \text{Title}$
$b? \notin \text{stock}$
$\text{book}' = \text{book} \cup \{b? \mapsto t?\}$
$\text{members}' = \text{members}$

Why is stock' not defined in the predicate part of this schema?

Explain Δ Static Library and \exists Dynamic Library

Partial specification for student 2

The AddBook operation

The operation to add a book to the system will update the static library system. A Book-id and a Title are inputs to the system.

Will this operation change the state of i) the static library, ii) the dynamic library?

What is/are the pre-condition(s) for adding a new book?

What will the new values of book and members be after the operation has taken place?

Do you need to consider the values of on-loan, on-shelf, loans?

Write the schema for the operation AddBook

The system operations are built up by each student in turn. In this way each member of the group has to construct part of the Z specification from a description in English.

Evaluation

We assessed Information Gap against the criteria set out on the first page of this paper.

The technique should be suitable for large class sizes

We wanted each student to receive as much individual attention and feedback as possible, although, with classes of fifty plus students, we realised that this could not be provided by a single lecturer. We thought that the technique would be suitable for large class sizes, that the sessions would run themselves and that we, as staff, would be largely superfluous. This was not the case initially as many of the 'tutors' wanted advice about to help their partners without telling them too much. We feel that this is more a problem of an unfamiliar technique than difficulty with the material, and the students were more confident in later sessions. Information Gap is suitable for all class sizes, once the students are familiar with the approach.

The technique should make efficient use of resources

All the students were very enthusiastic about this approach. They enjoyed playing the role of tutor and felt that explaining parts of the Z and answering questions gave them a good understanding of the language. Several students said that they felt more relaxed about asking their partner 'silly' questions than they would be in a normal tutorial situation. They found that they had to work as hard, to understand the Z they had been given and to help their partner formulate the mathematics, as they did to write their own Z. The students were using their peers as a resource and only appealing to the staff occasionally for clarification.

The technique should provide quick and comprehensive feedback on student performance

Feedback was quick and the quality improved as the 'tutors' became more familiar with their role. We found it was necessary that the tasks set were explicitly defined and that the role/responsibility of each student clearly understood. One pair of students failed to grasp the point of the exercise and raced through it, gaining little insight, by merely showing each other the Z when the going got tough.

The technique should motivate and interest students

The novel approach and the emphasis on student responsibility for the success of the exercise appeared to be motivating. We received comments such as 'invigorating' and 'it gets the adrenaline going' and there was a constant buzz of activity.

The technique should develop student confidence in communicating by means of a formal language, Z.

Our objective was to develop student confidence in communicating by means of a formal language, Z. Students should acquire not only current fluency in the language, but also comprehensive recall in the future, and a sound understanding of the underlying principles. By requiring the students to read and explain the Z schemas, they found they had to grasp the basic mathematics. We observed regular reference to the mathematical glossary supplied. This understanding was then re-inforced by writing a similar Z schema themselves. The level of understanding achieved by this group surpassed that of previous years.

The technique should encourage learning at different levels of expertise

As expertise within a specific domain increases, the representation within memory of goals and plans to achieve them becomes more complex. The following process of plan acquisition was mapped out by Black et al. (1988) when describing the learning of a human machine interaction. We argue that it is equally applicable to the learning of a formal language and that this process explains, to some extent, the suitability of the Information Gap technique to students with different levels of expertise.

Phase One: Preconceptions. With completely naive students the only plan knowledge is based upon preconceptions. These preconceptions are often inappropriate to the planning domain. The students described in this paper have developed beyond this phase. They already know how to interpret parts of a Z specification.

Phase Two: Initial Learning. With experience the student learns to overcome preconceptions by learning goals relevant to the task and by learning actions that will accomplish those goals. The learning of goals relevant to the task comes out of experience. We expect the less able students to be operating at this level of expertise during the information gap exercise.

Phase three: Plan Development. Planners now learn the frequent action combinations that will achieve general goals. This enables the students to generalize their planning knowledge. The requirements of the Information Gap approach, where each student has partial knowledge of the complete specification, mimics and is particularly compatible with the kind of problem solving behaviour required in this phase. Thus this approach facilitates the students' transition from the "Initial Learning" phase and the consolidation of the "Plan Development" phase.

Phase four: Increasing Expertise. Planners now form compound plans, where each goal may be achieved by several different plans. By forcing students to teach and explain as well as create Z specifications to their own requirements, we encourage further elaboration of the students' knowledge.

Conclusion

Our experiences with this approach lead us to believe that Information Gap is a useful and stimulating technique in the teaching of formal specification languages. We are hopeful that the technique can also be applied to the teaching of different types of computing language, and that the teaching of foreign languages in schools provides a sound basis for computer science lecturers seeking innovative and efficacious methods of teaching.

Reference

Black, J.B., Kay, D.S. & Soloway, M.E. 1988
Goal and Plan Knowledge Representations: From Stories to Text Editors and Programs,
in Carroll, J.J.(ed.) *Interfacing Thought*. Bradford books, MIT Press.