



A first course in formal specification: design principles and experience

D. Lightfoot

*School of Computing and Mathematical Sciences,
Oxford Brookes University, Oxford, OX3 0BP, UK*

The course

Modular degree courses at Oxford Brookes University consist of modules, each of which takes one term to complete. The term currently consists of nine weeks of teaching, followed by a reading week and an eleventh, final week in which the students take examinations. Although there are prerequisite dependencies between modules, it is best that as much as possible modules are self-contained. These are the constraints on the amount of material that can be covered in each module and the style in which it is taught.

A one-term module introducing formal specification using the Z notation has now run for four years. The principles put into practice in constructing this module and the experiences of running it are discussed in this paper.

Design principles

The design of the module was based on the following principles:

- The constructive nature of the Z notation would be stressed.

400 Software Engineering in Higher Education

- The mathematical treatment would be interleaved with examples to illustrate its use and to motivate the learner.
- The treatment would not be more mathematical in style than necessary.
- The module would not contain too much material, and yet would be coherent.

These stemmed from the module-designer's own experience of learning formal specification and from practical considerations such as the time constraints and the expected abilities and prior knowledge of the learners.

Constructive style

The fact that the Z notation is *constructive* means that learners can apply what they have already learnt to learning newly introduced constructs. For example, when studying functions learners use what they know about relations, just as when learning about relations they used what they knew about sets. Starting with elementary treatment of (typed) sets and logic that is little more than the Boolean algebra known to many students of computing, the remainder of the notation is built up. This pertains consistently in the Z notation and is a considerable aid to the learner. It was one of the reasons for choosing Z as the notation to teach, since it is not so true for some other formal specification notations.

Examples

Examples are interleaved with the mathematical treatment, providing illustration of features covered so far and showing the need for further features. A complete, though necessarily somewhat trivial example of a formal specification can be shown after only one section of formal material and its introduction helps the students to understand what formal specification is *for*.

Style of treatment

If formal specification is to become generally used it cannot remain the preserve of mathematicians. It is important that the topic be presented in a way that makes students' learning as straightforward as possible. Although successful completion of a first-year module in discrete mathematics is a prerequisite of the formal specification module, it is

sensible to revise the ideas of discrete mathematics and to take care to make the notation of Z very clear.

The module has been successfully completed by students who had not been able to study the discrete mathematics module.

A particular point considered was whether to use the conventional, correct mathematical symbols of the Z notation, thus incurring some difficulties in the preparation of texts, or whether to replace them by plain-text, "ASCII" equivalents. For example, should one use the $\in$ symbol or spell it (say) "elem"? Since part of the objective of the course was that the student should be able to *read* a formal specification document it was decided to use real mathematical symbols. This has proved to be practical, since most of the required symbols are now widely available in "Symbol" fonts of modern text-processors.

A further decision was the extent to which *proof* was to be covered. Although it is a very important topic, it was felt that the students were not primarily mathematicians and would find proof difficult. In the event the emphasis has been on using mathematics and the Z notation as a means of expression, and this has proved successful.

Self-contained

It is clear than with only nine weeks of teaching available the material must be somewhat truncated. It was important to include enough to make learning the subject worthwhile, without overloading the students. In the event, the time allowed treatment of sets, logic, schemas and schema calculus, predicate calculus with quantifiers, relations, functions and sequences. The ideas of functions cover not just functions of the form:

$$f: X \rightarrow Y$$

but also:

$$f: X \rightarrow (Y \rightarrow Z)$$

As mentioned above, little emphasis is given to proof. A further omission is bags, which, although often convenient, can be represented by functions.

402 Software Engineering in Higher Education

Assessment

In addition to exercises done by the students to practise notations and their use, the module is assessed by work done by the students during the course. This consists of two parts:

The first of these is a straightforward exercise done early in the course, which acts mainly as a means of boosting the students' confidence and carries relatively few marks.

The second, main assessment is a complete formal specification document written by the student on a topic of their own choice. There have been some very interesting examples done by students, although some students find that they need to have topics suggested to them.

The overall assessment is 70% for the examination and 30% for the coursework.

Teaching materials

When the module was first delivered there were only a few books available on formal specification in Z. Most notable and useful were Hayes [1] and Spivey [2]. Hayes is a collection of tutorial examples of Z and is very useful as illustrative material, though it cannot stand alone as a teaching text for beginners. Spivey is (to date) the definitive reference manual for Z, but although it contains an excellent tutorial chapter, it could not be used as sole teaching material for a module such as this.

Before developing the module, the module-designer had begun to write a textbook on formal specification in Z and so overhead transparencies, student notes, and exercises and solutions for the module were developed. The expanded notes and exercises and solutions have been published as a textbook (Lightfoot [3]).

The textbooks are supplemented by exercises and solutions, both for individual work and for work in groups. The group work is not assessed and leads to lively and productive discussion and co-operation.

Experience

History

The material of the module was originally developed on Higher National Diploma and Higher National Certificate courses, where the small student numbers made development and production of teaching materials easier. The module has now been run once a year for four years. The number of students choosing it has risen continuously over this period.

Results

The success rate of students is very good, with some students each year obtaining very high scores in the assessment (coursework and examination). In the case of very high scores, scripts are always sent to external examiners for confirmation that the level of attainment expected is reasonable. In all cases so far the external examiners have confirmed the grading of the internal examiners and assessors.

Each year however there have been some students who, for one reason or another, have not been successful in the module. It is hard to generalise about these students, but they seem also to be weak in other modules of the course. Overall the module has an average score in line with other modules of the modular degree course.

That some students find the module immediately useful is indicated by the fact that several have gone on to use Z as part of their final-year projects.

Feedback

Students give formal feedback after all modules of the course and feedback from this module indicates that the majority of students find it enjoyable and worthwhile.

Books

Since the module first ran, several further introductory texts on Z have appeared. Advice to the students has been to buy either Lightfoot or another introductory textbook of their choice and to buy Spivey as a reference text.

404 Software Engineering in Higher Education

Software tools

So far the module has been run without the aid of specific software tools, though most students use text-processors to prepare their specification documents. Although software tools are available, it is important that any tools chosen should be quick to learn and convenient to use. This is particularly true given the constraints that apply to the module. Tools that involved marking-up text for a pre-processor seemed not to offer the desired convenience.

The ideal software tool should offer "what-you-see-is-what-you-get" type-setting as well as syntax checks on the Z. It would be preferable that such a software tool would allow the Z sections to be exported to a regular text-processor, since in a Z document the mathematics is embedded in natural-language commentary.

Courses for industry

The module has been developed into a short course for industry and has been presented very successfully on several occasions.

Future developments

Further topics

Since the module was developed certain new features have been introduced to Z. One of these is the "dot" notation for qualifying the names of components of schemas. Furthermore, greater emphasis is now given to the concept of "promotion", which allows a limited specification to be readily generalised.

The use of both of these will be introduced shortly, though they will not have a dramatic effect on the style of the module.

Standardisation

The Z notation is currently being standardised under the auspices of the International Standards Organization (ISO). The work of the panel is being followed and will in due course be reflected in modifications to the course materials where necessary.

Object-orientation

Various object-oriented versions of Z have been proposed and work is continuing on these. The possible introduction of an object-oriented version of Z to the module will be considered. It may be that the ideas

and notations of object orientation can readily be introduced to the module without requiring the omission of current material, or they may form the basis of a further module.

Follow-up course

The process of developing a program from its specification had been deliberately omitted from the module. The treatment of this process; the formal development of software, forms the basis of a new advanced module for the computing courses, which will run for the first time in the academic year beginning 1994.

Courses for industry

A short course on applying formal techniques to the development of software has already been developed and successfully presented for local industrial clients. The experience of designing and presenting this course has been applied to the development of the degree module on formal software development.

Conclusions

The experience at Oxford Brookes University is that formal specification can be taught successfully to students who do not have specific mathematical skills or knowledge.

It is expected that the impact of formal methods on computing degree courses will increase and further modules are being made available and being considered.

The requirements of industry for training in formal methods are also being addressed by a programme of short courses.

The use of appropriate, well-designed software tools would assist in the students' learning and in their preparation of specification documents.

References

1. Hayes, I.H. *Specification Case Studies*, Prentice Hall (2nd Edition) 1994
2. Spivey, J.M. *The Z Notation*, Prentice Hall (2nd Edition) 1993
3. Lightfoot, D.E. *Formal Specification Using Z*, Macmillan, 1991