EXPERIENCE OF INTRODUCING THE VIENNA DEVELOPMENT METHOD INTO AN INDUSTRIAL ORGANISATION

M.I. Jackson, B.T. Denvir
STC Standard Telecommunication Laboratories Limited
London Road
Harlow, Essex,
CM17 9NA, England.

R.C. Shaw
STC IDEC
Six Hills Road,
Stevenage, Herts,
SG1 1YB, England.

1. INTRODUCTION

Formal specification techniques are frequently advocated as a basis for improved software development methods leading to greater product quality and reduced life cycle costs. Although a great deal of research has been done into formal specification languages and techniques, relatively little experience has been gained of the impact of such approaches in industrial environments.

In order to introduce such approaches effectively, a number of difficulties have to be overcome, such as the education and training of industrial personnel, the development of appropriate standards, the phased introduction of formal specification techniques alongside established methods and practices etc.

The paper addresses some pragmatic approaches to problems such as those described above. It is based on experience gained within STC of the introduction of the Vienna Development Method (VDM) over the last two years.

2. BACKGROUND TO VDM WITHIN STC

The Systems and Software Technology Division of STC Standard Telecommunication Laboratories Ltd., has been conducting research into formal (i.e. mathematically-based) approaches to system development since 1979. Particular emphasis has been placed on improved specification techniques, since weaknesses in this area are known to cause serious design errors often discovered late in the development process with commensurately increased costs of removal.

In 1982, a major new project established in the office systems area requested advice on formal specification methods from STL. Of the methods under consideration at that time, one in particular, the Vienna Development Method (VDM), was recommended for consideration.

An attractive feature of VDM is that it provides a framework for development; formal notations are used for defining system behaviour at various levels of refinement during the development process and the necessary obligations to be satisfied for verifying that a refined description satisfies a more abstract description are clearly defined. However, a spectrum of verification techniques may be applied by the user to satisfy these obligations, ranging from totally formal proof techniques (used where high reliability is required) to informal justification of correctness (more practicable on a day-to-day basis). This flexibility to apply the disciplines of the method with varying degrees of formality within a formal framework identifies VDM as a "rigorous" (as opposed to fully formal) method and accounts to a great extent for its success in practical situations.

VDM originated in the IBM Vienna Research Laboratories and is most closely associated with the names of D. Bjorner and C.B. Jones. Our reasons for recommending it were largely pragmatic, for example:

(i) Considerable user experience of the method already existed within organisations such as IBM and the Danish Datamatik Centre. A number of case studies have been published, for example, in the recent book by Bjorner and Jones (BJO82). The strengths and weaknesses of the method are well-understood. (The reader is referred to the critical appraisal by Prehn et. al. [PRE83] for more information.)

(ii) Training and consultancy in the method were available. Prof. C.B. Jones, now of Manchester University, was willing to present an established 2-week training course for the project team and to provide continual support to STC through consultancy.

(iii) A well-written and accessible text book produced by Professor Jones, was available to support the course material [JON80].

VDM was thus seen to be among the most mature methods of system development and, therefore, suitable for industrial exploitation.

3. VDM EVALUATION

In the Summer of 1982 an evaluation exercise was conducted in order to assess the suitability of VDM as a specification vehicle for the proposed project. This exercise was organised as follows:

- A subsystem of the project was chosen by management as the case study to be used.

- A small group of analysts, having no previous familiarity with formal methods, were selected and introduced to VDM.

- Two STL staff members already familiar with VDM, were asked to support the analysts as consultants (alongside Professor Jones). Consultancy activity was intensive at the beginning of the exercise, but was conducted at a rate of a one-day review meeting every two weeks thereafter.

- An observer, independent of the analysts and consultants, was appointed to identify assessment criteria before commencement of the evaluation, to observe the conduct of the exercise and its outputs, to write the final evaluation report and to produce recommendations to management.

- The Chief System Designer was involved from time to time to answer questions regarding the requirements and to play the role of "customer".

The Evaluation exercise took as its input an English language statement of requirements. Its outputs were the VDM specification corresponding to the statement of requirements, the Evaluation Report and future plans and recommendations for VDM. The conclusions drawn can be summarised as follows:

(i) Overall VDM was found to be very effective. The identification of abstract data structures in the VDM specification was found to be a powerful concept, amenable to "brain-storming" activity. The production of the formal specification helped in identifying the best choice of operations and functions available to users of the system. A number of versions of the specification were produced and each was validated by discussion with the "customer"; this systematic, iterative approach continued until a satisfactory version, acceptable to the "customer", was produced.

(ii) VDM provided a number of useful thinking 'tools' the effects of which were evaluated as follows:

- The designer was forced and able to consider many more aspects of the requirements than with previous (non-formal) analysis techniques.
- VDM enhanced the analytical potential of the analysts by allowing them to reason about the abstract models developed in the specification process.
- VDM allowed the analysts to distinguish more easily between WHAT the system should do and HOW it should work.

(iii) VDM is capable of producing a precise, consistent and unambiguous specification. In the evaluation exercise it revealed many (unrealised) anomalies and inadequacies in the informal statement of requirements. Although no formal validation of the specification was carried out, the analysts were able to reason informally about the specification and there was a higher degree of confidence in the specification than normal.

(iv) The initial statement of requirements was found to be too brief, incomplete and ambiguous in places, to have an inadequate description of the data objects and the user operations of the system, and to include unnecessary implementation detail in places. (In spite of these faults it was considered, however, to be a fairly typical "marketing" style specification.) After producing the VDM specification, the analysts were able to derive systematically from it an English language version which was superior in all the above respects to the original and which could form a very suitable basis for marketing/customer documentation.

(v) One of the benefits of using VDM was that essential decisions had to be made during the specification phase (where they could be discussed with the customer), whereas with traditional methods these were often not made until detailed design or even implementation.

(vi) There were a number of shortcomings in the basic method which needed to be remedied before it could be used on the project. These included:

- A well-defined concrete syntax which could be input and output by the terminals and printers used on the project, so that specifications could be held within the project database and subject to normal disciplines of version control etc.

- Better structuring facilities which would allow the factoring of large specifications into modules with well-defined interfaces and tight control of the scope of names. This is necessary to support the development of large specifications by different analysts, possibly in different teams.

- More convenient facilities for defining and handling errors in a system specification, i.e. to factor the description of error behaviour from that of normal behaviour.

- The lack of an extended linguistic framework in which the VDM abstract specifications could be represented alongside more conventional pseudo-code designs developed subsequently to match the specfication.

As a result of the evaluation exercise, project management recognised the positive benefits of using VDM and decided to incorporate it as a project method. Consequently a training programme, a language development and support tools activity and a maintenance and coordination activity were initiated.

4. TRAINING PROGRAMME

The first two week VDM course, presented by Professor Jones, was held in the Autumn of 1982 and was attended by 10 students, seven of whom were from the project and three of whom were prospectivè lecturers. The seven project members were all senior staff, being project managers (1), technical consultants (3) and team leaders (3). Subsequently, further training courses were given for the benefit of other project staff. These were given by the STC staff trained as lecturers.

As a result of growing interest in VDM within the Company, a variety of courses have been developed. These comprise:

(i) A Perspective on VDM. This is a three day non-residential course aimed at project managers, team leaders, technical consultants and support staff involved in activities such as quality assurance, technical documentation or marketing and who require a sound understanding of VDM. Its objectives are to provide a broad appreciation of formal methods, the ability to read and review VDM specifications, and the ability to control the introduction and use of VDM into a project.

(ii) VDM Foundation course. This is a 10-day residential course for technical staff wishing to produce system specifications written in VDM and for those who need design systems to satisfy VDM specifications. Students completing the course

> should be able to read and write VDM specifications, demonstrate that a specification meets a requirement and demonstrate that a design meets a specification.

At the time of writing, seven Foundation courses and eight Perspective courses have been run. Eighty people, from STC and other organisations, have attended the former and about sixty-five more have attended the latter. The course programme, although developed for internal purposes, has recently been made publicly available through STC IDEC. This initiative has been supported by Professor C.B. Jones and by Mr. D. Talbot, the Director of Software Engineering for the Alvey Programme, since it is seen as highly compatible with the Alvey Strategy in software engineering.

A particular feature of the course is the Material in the Foundation course is based upon the book by C.B. Jones, "Software Development - A Rigorous Approach" [JON80]. The course concentrates on developing familiarity with the notation and confidence in its use. This is achieved through a series of lectures, exercise sessions, student presentationa and discussions in which attendees take a substantial specification problem in a workshop in the second week in small teams.

Despite many initial reservations the majority of students have left the course with the conviction that VDM provides a powerful technique for introducing formal methods into the specification phase of the software life cycle. What impact this will have on their work within projects has yet to be analysed.

On the early courses no specific entry requirements were established and this showed that students with a weak mathematical background do experience difficulty with some of the course material. Of the students who attended the first Foundation courses about 10% have encountered varying difficulties in handling the notation. This has been sufficient to impair their ability to use the notation within a wider context of ideas.

To overcome this problem a one day course covering the elements of Discrete Mathematics was set up and this is presented about one week prior to the delivery of each Foundation course. A self test procedure has now been developed which attempts to assess an individual's familiarity with elementary Discrete Mathematics.

This test will allow individuals to decide, for themselves, whether they can attend the VDM Foundation course directly or should avail themselves of either the one day introductory course in Discrete Mathematics or a more extensive treatment of the subject.

The later courses showed an improved "success rate" in this regard, approaching 100%. In order to assist those people who have attended the Foundation Course in their initial VDM applications, consultancy services are available and their use is encouraged.

5. LANGUAGE DEVELOPMENT AND SUPPORT TOOLS

Language Development activities have been concentrated in two main timescales, short-term and medium term.

In the short-term, it was recognised that there was an immediate need for facilities to allow the specification process to be carried out 'in-the-large' and for designs to be documented alongside the specifications that they implement. STC IDEC consequently undertook the development of an extended design language with the following features:

(i) It is possible to represent (almost) all of the VDM specification language taught in the programme of courses.

(ii) A "module" facility is provided which allows large specifications to be developed in self-contained units with tight control over the scopes of names and well-defined interfaces.

(iii) A programming language-like pseudo-code is provided for expressing designs within the same modules as their specifications.

This approach taken has been heavily influenced by work in IBM's Boblingen Laboratory on the SLAN-4 languge [BEI83] and is based on the assumption that, in the short-term, VDM will be used mainly as a specification aid leading into more established and less formal approaches to design. (This assumption is due mainly to two factors - firstly, that improved specification is the most significant and practicable benefit that formal methods can bring at present, secondly, that to introduce more rigorous design techniques requires more powerful automated tools than are currently available outside research establishments).

In the medium term, it was recognised that in order to extend the VDM specification language in a coherent and uniform way, and in order to build more powerful behaviourally-oriented support tools, it would be necessary to improve the standard of definition of the existing VDM specification language. Consequently, a joint activity was started by STL and the University of Manchester to define for STC a VDM "Reference Language" to be used as the standard within STC. To date, documents describing the concrete syntax, abstract syntax, type model and context conditions of the Reference Language have been issued and further work in the area of formal semantics is underway.

In the area of support tools, STC IDEC has produced a UNIX-based tool to perform syntax checking and (partial) type checking for the pseudo-code based language described above. In the medium term, it is intended to develop more extensive toolsets for VDM based around the VDM Reference Language, and a UK Alvey-funded project has just commenced for this purpose, in which STC, ICL, and Manchester University are taking part.

6. MAINTENANCE AND CO-ORDINATION ACTIVITIES

When it became apparent that there was wide interest in VDM across a number of STC units, a VDM Co-ordination and Maintenance Committee was established to support the controlled introduction of the method into STC. The membership is drawn from all actively involved units (i.e. those who are applying VDM or who have sent individuals on VDM courses). The terms of reference of the committee include:

- To develop and publish company standards for VDM and to implement change control procedures for such standards.

- To act as a qualification body for VDM support tools.

- To advise on courses, curricula and other matters related to the education and training required to introduce VDM into development centres.

- To disseminate information regarding the development and use of VDM by publishing occasional reports and by organising workshops and conferences.

- To co-ordinate interaction between STC units and external consultants with respect to VDM.

- To provide a general query/response service to established and prospective users.

The Committee has been active for about 18 months to date and has achieved positive results in all the above areas. In particular, standards have been established for the concrete and abstract syntax of the language as used within the company, and for a type model and context conditions.

7. ASSESSMENT OF VDM EXPERIENCE

In addition to training a substantial number of software engineers in the method and its language, we have now had experience of applying it to a number of typical problems in the information technology industry. These include aspects of advanced office communication systems, such as electronic mail, a data-base of personal calendars, and user access control. Case studies have also been carried out in a more conventional telecommunications problem and a real-time control system.

In general we have found the method particularly effective as an intellectual tool for analysing the problem and synthesising its specification. Deficiencies in the first statements and conception of the problem's requirements appear to be highlighted earlier in the software development process, enabling remedial action (or indeed reconsideration of the project's viability) to take place after a smaller developmental investment. Our experience is that the overwhelming majority of practising software engineers in a high technology industry can be taught the method in a practicable time-frame : here we believe the workshop component of the course to be particularly valuable.

On first adopting the VDM, different levels can be identified and progressively used by an organisation new to the method:

(i) Formulating data types, the system state, and operations with their signatures.

(ii) Formulating pre- and post-conditions of operations, and data-type system state invariants.

(iii) Identifying the obligations of proof of correctness of the specfication and its requirements.

(iv) Carrying out proofs by rigorous argument in specification reviews.

(v) Carrying out formal proofs.

The first levels serve as a good communication medium between the roles in the development process, and the later levels would increase further the confidence we may feel in the quality of the final poduct. In STC we have carried levels (i) to (iii) and to some extent (iv) also. We have not attempted level (v) to date.

Weaknesses of the method evinced by our experience so far are:

(a) For large systems and where a team of people is working on the specification, there is a need for dividing the specification up in a modular and well-structured way. The language does not provide a clear way of doing this.

(b) The language does not support the communication aspects of specifying concurrent systems or the dynamic aspects of sequential systems whose functionality changes. However, it can solve the majority of specification problems with such systems, and the concurrency aspects have to be expressed in some separate way.

We have also felt the need for some support tools, and are taking active steps regarding this.

8. CONCLUSIONS

VDM, as a method for formal specification, can be successfully introduced into an industrial organisation, if appropriate levels of investment are made in the areas of:

(i) Evaluation, by case studies, of the suitability of the method for the application area of interest. The evaluation criteria should be properly defined, monitored and assessed.

(ii) Training for staff, of a professional quality.

(iii) Consultancy, as appropriate to particular needs, by skilled and experienced personnel.

(iv) General support, for example, by developing and implementing corporate standards.

(v) Support tools in the longer term, such as a specification oriented database and a syntax-directed structure editor, which would add to the effectiveness of the method.

9. FURTHER WORK

We perceive that further work would be beneficial in the following areas:

(i) Complete the semantics definition.

(ii) Develop a means of formulating modular specifications in a semantically well defined way.

(iii) Consider how to approach the specification of concurrent systems.

10. ACKNOWLEDGEMENTS

The results described above reflect the work of a number of people in STC, including P.N. Hudson who developed the STC IDEC Design Language, and B.Q. Monahan of STL who has done substantial work on the definitions of the syntax and semantics of the VDM Reference Language. We also greatly appreciate the contribution of Professor C.B. Jones who has supported us thoughout with his advice and encouragement.

11. REFERENCES

BEI83 Beichter F, Herzog O, Petzsch H.
SLAN-4: A Language for the Specification and Design of Large Software Systems.
IBM Journal of Research and Development, Vol. 27, No. 6, November 1983.

BJO82 Bjorner D, Jones C.B.
Formal Specification and Software Development.
Prentice Hall, 1982.

JON80 Jones C.B.
Software Development: A Rigorous Approach
Prentice Hall, 1980.

PRE83 Prehn S, Hanson I.O, Palm S.U, Gobel P.
Formal Methods Appraisal: Final Report
ESPRIT Preparatory Study Report, June 1983.