INTERNATIONAL STANDARD

**ISO/IEC 13817-1**

First edition
1996-12-15

# Information technology — Programming languages, their environments and system software interfaces — Vienna Development Method — Specification Language —

## Part 1:
Base language

*Technologies de l'information — Langages de programmation, leurs environnements et interfaces logiciel système — Méthode de développement de Vienne — Langage de spécification —*

*Partie 1: Langage de base*

Reference number
ISO/IEC 13817-1:1996(E)

# Contents

## Figures



## Tables

# Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 13817-1 was prepared by Joint Technical Committee, ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces.*

ISO/IEC 13817 consists of the following parts, under the general title *Information technology — Programming languages, their environments and system software interfaces — Vienna Development Method — Specification Language*:

— *Part 1: Base language*

Additional parts will specify modules and the development method.

Annexes A to E of this part of ISO/IEC 13817 are for information only.

# Introduction

## Historical background of the Vienna Development Method (VDM)

VDM was developed at the IBM Laboratory in Vienna. The Laboratory came into existence in 1961 when Professor Heinz Zemanek of the Technical University in Vienna decided to move his whole group to an industrial home.[1)] They had previously developed a computer called Mailüfterl at the Technical University. From 1958 the group had been increasingly involved in software projects including the construction of one of the early compilers for the ALGOL 60 programming language. As time went on they found it difficult to get adequate support for their projects and eventually joined IBM. In the mid-1960s, IBM decided to develop a new programming language for which the ambition was to replace both FORTRAN and COBOL. The language, which was at first called New Programming Language (until the National Physical Laboratories in the UK objected to the acronym — the language became known as PL/I), was clearly going to be large and it was decided that it would be useful to try to apply formal techniques to its description.

Based on their own work — and influenced by research work by Cal Elgot, Peter Landin and John McCarthy — the Vienna group developed an operational semantics definition of PL/I which they called ULD-3 ('Universal Language Description'; ULD-2 was the name applied to the IBM Hursley contribution to this effort — the language itself was being developed mainly from Hursley along with the early compilers; ULD-1 was a term applied to the natural language description of the language). The description of PL/I in ULD-3 style ran through three versions. These are very large documents. Operational semantics is now seen as unnecessarily complicated as compared to denotational semantics. However, to make the principles of denotational semantics applicable to a language like PL/I with arbitrary transfer of control, procedures as arguments, complicated tasking, etc. required major theoretical break-throughs and a considerable mathematical apparatus not available at the time. The effort of the formal definition uncovered many language problems early and had a substantial influence on the shape of the language.

Towards the end of the 1960s serious attempts were made to use the ULD-3 description as the basis of compiler designs. Many problems were uncovered: in general, one could say that the over-detailed mechanistic features of the operational semantics definition considerably complicated the task of proving that compiling algorithms were correct. But again one should be clear that the work was a technical achievement: a series of papers was published that described how various programming language concepts could be mapped into implementations which could be proved correct from the description (e.g. in JL71). In addition, a series of proposals was made which could simplify the task of developing compilers from a semantic description. One of these was an early form of an exit construct (HJ70) which led to an interesting difference between the Vienna flavour of denotational semantics and that used in Oxford. Another VDM-like idea that arose at this time was Peter Lucas' *twin machine* proof (Luc68), and subsequently the observation that the ghost variable treatment in the twin machine could be replaced by retrieve functions (Jon70) as a simpler way of proving most cases of data development are correct. It is worth noting that Lucas' *twin machine* idea has been re-invented several times since: the generalization of retrieve functions to relations can be seen as equivalent to twin machines with invariants.

Hans Bekič spent some time in England with Peter Landin at Queen Mary College and was the person who first pushed the Vienna group in the direction of denotational semantics. (Until his untimely death in 1982, Hans Bekič

[1)] This is not intended to be a history of the Vienna Laboratory: citations are limited to those concerning VDM itself.

had published relatively little of his scientific research; some of his important papers were published posthumously in Bek84.) Another crucial stimulus was the visit to the Vienna laboratory by Dana Scott in 1969 (see dBS69).

During the period from 1971 to 1973, the Vienna group was diverted into other activities not really related to formal description. Cliff Jones at this time went back to the Hursley Laboratory and worked on a *functional* language description (ACJ72) and other aspects of what has become known as VDM. In particular he published a development of Earley's recogniser (Jon72) which is one of the first reports to use data reification. In late 1972 and throughout '73 and '74 the Vienna group (Cliff Jones returned and Dines Bjørner was recruited) had the opportunity to work on a PL/I compiler for what was then a very novel machine architecture. They of course decided to base their development for the compiler on a formal description of the programming language. PL/I was then undergoing ECMA/ANSI standardization (ANS76). The Vienna group chose to write a denotational semantics for PL/I (BBH+74); this is the origin of the VDM[1)] work on language description techniques.

During the same period, at the IBM Laboratory in Hursley, an investigation into the use of Meta-IV to formaliy describe five of the major languages supported by IBM was carried out. The languages were PL/I, BASIC, FORTRAN, APL and COBOL. Sketches for parts of FORTRAN and APL were written, and a full description of minimal BASIC was produced. This work ceased when the language work was moved from Hursley.

Cliff Jones and Dines Bjørner took upon themselves the task of making sure that something other than technical reports existed to describe the work that had gone on on the language aspects of VDM: Be78 is a first book-length description of that work. In ESRI, Cliff Jones also developed the work on those aspects of VDM not specifically related to compiler development and the first book on what is now generally thought of as VDM is Jon80. Both of these books have now been superceded: the language description work is best accessed in Be82 and — in its second edition — the non-language work is best seen in Jon90 and also in AI91.

Within IBM, from 1978, a number of projects used VDM (for other than language descriptions). It was during this period that specifications of several large systems were carried out at the Böblingen Laboratory. (These included a file system for DOS and an fault report tracking system; a data reification for the file system was attempted to show that the proof techniques were viable in an industrial environment.) This work was carried out as part of a technology transfer program. Later the IBM Program Product Development Centre in Rome became involved, and work was done to formally specify a hotel management system. This work showed that VDM was suitable for large-scale projects; unfortunately little has been published on this experience.

Dines Bjørner's group at the Technical University of Denmark strenuously pursued the use of VDM for language description and he and his colleagues were responsible for descriptions of the CHILL programming Language and a major effort to document the semantics of the Ada programming language (BO80).

Language work was also continued at Leicester University where a formal definition of the full Pascal language was written (AH82) and later a formal definition of the programming language Modula-2, which became a Draft International Standard (AeWO).

The non-language, specification, aspects of VDM were taken up by the STL laboratory in Harlow and, partly because of their industrial push, the British Standards Institute (BSI) was persuaded to establish a standardization activity. This activity has not been easy because of the differences between the pressures of those interested in the language description aspects of VDM and those who are more interested in pre- and post-conditions, data reification and operation decomposition. It is to the credit of both the BSI and ISO Standards committee that they have managed to bear in mind the requirements of both types of user and come up with a standard that embraces such a wide scope of technical ideas. STL was responsible for funding the first formal semantics of VDM-SL and the work that was done at Manchester University by Brian Monahan (Mon87) was used as the starting point of the formal description of the specification language. A new formal description based on this work was produced because of the necessity of merging together the two aspects of the specification language.

The outcome of the standardization effort initiated by STL through the *British Standards Institution (BSI)* was the formation of a panel (BSI IST/5/-/50) whose membership was also open to participants from non-British organizations.

[1)] It is worth getting some acronyms sorted out: VDL stands for Vienna Description Language and was a term used to describe the operational semantics (ULD-3) notation; VDM stands for Vienna Development Method and relates to the post-1973 work; the specification language part of VDM is sometimes known as 'Meta-IV'; it is now known as VDM-SL.

In 1991 the need for a VDM-SL standard was also recognized by *ISO/IEC JTC1*[2)] by the formation of a Working Group, SC22/WG19. The actual work on the standard is done by the members of the BSI Panel.

Work on the Standard encouraged the building of computer-based tools to support both the language and the development method. Adelard produced 'SpecBox', a tool that provided syntax and type checking of VDM specifications. Manchester University and the Rutherford Appleton Laboratories wrote 'mural' (JJLM91), a prototype tool that supports both the specification language and the development method, providing a proof engine to help with data reification and operation decomposition. The Technical University of Denmark and the National Physical Laboratory have configured the Cornell Synthesizer (a structured editor tool) to support VDM-SL with some type-checking facilities — this tool was used to produce the static semantics of VDM-SL. IFAD have produced a tool for syntax and type checking together with an interpreter for an executable subset to allow rapid prototyping.

During its history, the Vienna Development Method, together with its specification languages has had a profound influence on both the specification of programming languages and the specification and development of systems. The ideas in VDM have influenced several other specification languages including RAISE, COLD-K and VVSL.

[2)] Joint Technical Committee 1 of the International Standards Organization and the International Electro-technical Commission.

# Information technology — Programming languages, their environments and system software interfaces — Vienna Development Method — Specification Language —

# Part 1:
Base language

## 1 Scope

This part of ISO/IEC 13817 specifies the model based specification language VDM-SL (Vienna Development Method — Specification Language). It specifies:

- two representations: the mathematical and interchange;
- the syntax;
- the static semantics;
- the dynamic semantics;
- conformity for specifications and tools.

It does not specify:

- the proof obligations;
- the reification rules;
- the size or complexity of a specification that will exceed the capacity of any specific data processing system or the capacity of a particular tool, nor the actions to be taken when the corresponding limits are exceeded;
- the minimal requirements of a data processing system that is capable of supporting an implementation of a tool;
- the method that tools use for reporting errors.

## 2 Normative References

The following standards contain provisions which, through reference in this text, constitute requirements of this part of ISO/IEC 13817. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 13817 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 8859-1:1987[1)], *Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.*

ISO/IEC 14977:1996, *Syntactic Metalanguage — Extended BNF.*

1) Currently under revision.