MODUS

The Modula-2 Users Assotiation

Invites you to participate

The Directors:

Randy Bush
Pacific Systems Group

601 South 12th Court
US-Coos Bay, OR 97420
503 / 267 - 69 70

Tom DeMarco
Atlantic Systems Guild

353 West 12th Street
US-New York, NY 10014
212 / 620 - 42 82

Jean-Louis Dewez
Laboratoire de
Micro Informatique, CNAM
2, Rue Conté
F-75003 Paris
01 / 42 71 24 14

Swend Erik Knudsen
Rechenzentrum
ETH Zürich
Clausiusstrasse 55
CH-8092 Zürich
01 / 2 56 34 87

Heinz Waldburger
ERDIS SA

Petit Clos 4
CH-1800 Vevey 2
021 / 52 61 71

The Administrators:

USA
George Symons
MODUS
PO Box 51778
US-Palo Alto, CA 94303
415 / 322 - 05 47

Rest of the World
Aline Sigrist
c/o ERDIS SA
PO Box 35
CH-1800 Vevey 2
021 / 52 61 71

Modula-2 News

Issue # 0

October 1984

Modula-2 News is the quarterly, informal publication of Modus, the Modula-2 Users Association.

Content:

Revisions and Amendments to Modula-2

N. Wirth, ETH Zürich, 1.2.84 / 14.5.84

On November 21, 1983, a meeting was held with participants from several firms who had implemented Modula-2. Numerous features and facilities were proposed for addition or correction. The following subset was agreed upon. These rules should be regarded as revisions of Modula-2. Future implementors are encouraged to comply with these revisions, and existing compilers should be adapted. Although any change in a language is subject to resentment, the number of changes adopted here is very small and, I believe, each one is a genuine improvement.

1. Restrictions and Clarifications

1.1 The types of a formal VAR-parameter and that of its corresponding actual parameter must be *identical* (i.e. not merely compatible). This rule is relaxed in the case of a formal parameter of type ADDRESS, which is also compatible with all pointer types, and in the case of the type WORD, where the compatible types are specified for each implementation.

1.2 The types of the expressions specifying the starting and limiting values of the control variable in a for statement must be *compatible* (i.e. not merely assignment compatible) with the type of the control variable.

1.3 A process initiated in a module at priority level n must not call a procedure declared in a module at priority level m < n. Calls of procedures declared without priority are allowed.

1.4. Pointer types can be exported from definition modules as opaque types. Opaque export of other types may be subject to implementation restrictions. Assignment and test for (in)equality are applicable to opaque types.

1.5 All modules imported to the main module are initialized *before* the importing module is initialized. If there exist circular references, the order of initialization is not defined.

1.6 If a module identifier is imported, this does *not* imply that the identifiers of objects of this module become visible. However, those which are exported in qualified mode can be accessed by prefixing them with the module identifier.

2. Changes

2.1 *All* objects declared in a definition module are exported. The explicit export list is discarded. The definition module may be regarded as the implementation module's separated and extended export list.

DefinitionModule = DEFINITION MODULE ident ";" {import} {definition} END ident "." .

2.2 The syntax of a variant record type declaration with missing tag field is changed from

FieldList = | CASE [ident ":"] qualident OF ...

to

FieldList = | CASE [ident] ":" qualident OF ...

The fact that the colon is always present makes it evident which part was omitted, if any.

2.3 The type PROCESS in module SYSTEM is deleted. Its place is taken by the type ADDRESS.

3. Extensions

3.1 The syntax of the case statement and the variant record declaration is changed from

```
case = CaseLabelList ":" StatementSequence.
variant = CaseLabelList ":" FieldListSequence.
```

to

```
case = [CaseLabelList ":" StatementSequence].
variant = [CaseLabelList ":" FieldListSequence].
```

The inclusion of the empty case and the empty variant allows the insertion of superfluous bars similar to the empty statement allowing the insertion of superfluous semicolons.

3.2 A string consisting of n characters is said to have *length* n. A string of length 1 is assignment compatible with the type CHAR.

3.3 The syntax of the subrange type is changed from

```
SubrangeType = "[" ConstExpression ".." ConstExpression "]".
```

to

```
SubrangeType = [ident] "[" ConstExpression ".." ConstExpression "]".
```

The optional identifier allows to specify the base type of the subrange. Example: INTEGER [0 .. 99]

3.4 Elements of sets have been restricted to be constants. This restriction is now relaxed. The syntax of sets and factors changes to

```
ConstFactor = ... | ConstSet | ... .
ConstSet = [qualident] "{" [ConstElement {"," ConstElement}] "}".
ConstElement = ConstExpression [".." ConstExpression].

factor = ... | set | ... .
set = [qualident] "{" [element {"," element}] "}".
element = expression [".." expression].
```

3.5 The character "~" is a synonym for the symbol NOT.

3.6 The identifiers LONGCARD, LONGINT, and LONGREAL denote standard types (which may not be available on some implementations)..

3.7 The type ADDRESS is compatible with all pointer types and with either CARDINAL or LONGCARD. The interpretation of addresses as numbers depends on the implementation.

3.8 The standard functions MIN and MAX take as argument any scalar type (including REAL). They stand for the type's minimal resp. maximal value.

File: Modula.Revision.DOK

MODULA-2 IN THE PUBLIC EYE

CONTENTS

Modus members may use this material for non-commercial purposes.

1. BOOKS, MANUALS, NEWSLETTERS & MAGAZINES

a) Volition Systems
"Modula-2 User's Manual", Release 0.3, 26 August 1983.
"Implementation Guide for the Sage", Release 0.3e, 2 April 1983.
"Implementation Guide for the IBM PC and XT", 22 May 1984.
"Implementation Guide for the 8080/Z80", 9 November 1983.
"Implementation Guide for the Apple II", 15 November 1982.
"Implementation Guide for the Apple ///", 22 August 1983.

b) Lilith Handbook
"A Guide for Lilith Users and Programmers", L. Geissmann, J. Hoppe Ch-J. Jacobi, S. E. Knudsen, W. Winniger, and N. Wirth, Institute fuer Informatik, ETH Zuerich, 1982.

c) Wirth
"Programming in Modula-2", (first edition), Springer-Verlag, 1982.
"Programming in Modula-2", (second corrected edition), Springer-Verlag, 1983.

d) Structured Language World: quarterly newsletter focussing on Pascal, Modula-2 and Ada.

e) Journal of Pascal Ada and Modula-2: bimonthly magazine

f) Logitech
"Modula-2/86", Software Engineering Library Users Guide, 199 pages 1984.

g) Diser
"Modula Computer", user's manual, May 21, 1982, first available from Diser May, 1983 (?)

h) MRI "Modula-2 Handbook" (PC Version; 55 pages of revised ETH documentation with 5 pages of PC specific preface material).

i) Wiener & Sincovec
"Software Engineering in Modula-2 and Ada", John Wiley and Sons, February 1984.

j) AustMUG News: irregular newsletter from the Australian Modula-2 users distributed by John Gough, Queensland Institute of Technology.

1: "Modula-2, An Overview", p. 2-4, Received Dec. 21, 1983.
3: "The Structure and Use of Procedure Variables" by K.J. Gough, p. 5-16.
"2-DEBNF Form Modula-2", by A. H. J. Sale, Department of Information Sciences, University of Tasmania, p. 17-23.

2. REPORTS AND MONOGRAPHS (ETH reports)

1978(?), ETH Report No. 27, "Modula-2".

1979, "The Module: A System Structure Facility in High-Level Programming Languages", by N. Wirth.

March 1980, ETH Report No. 35, "A Simple Nucleus Written In Modula-2, Preventing Storage Overflows In High-Level Languages".

March 1980, ETH Report No. 36, "Modula-2" (second edition, Dec. 1980).

Oct. 18-22, 1982 "BugNet: A Debugging System for Parallel Programming Environments" by R. Curtis and L. Wittie, pg. 394-399, The 3rd International Conference on Distributed Computing Systems.

Feburary 1983, Modula/R Report (Lilith Version), Editor: Manuel Reimer.

July 1983, ETH Report, No. 54, "The Grid File: A Data Structure Designed to Support Proximity Queries on Spatial Objects", K. Hinrichs, J. Nievergelt.

September 1983, ETH Report No. 55, "Database Techniques for Professional Workstations", Carl August Zehnder (Ed.).

Data Management can be Learned (an Introduction to Database Techniques for Professional Workstations) by Carl August Zehnder.
LIDAS - The Database System for the Personal Computer Lilith.
Implementation of the Database Programming Language Modula/R on the Personal Computer Lilith, by Manuel Reimer.
Gambit: An Interactive Database Design Tool for Data Structures, Integrity Constraints and Transactions.
Hiquel: An Interactive Query Language to Define and Use Hierarchies, by Peter Ursprung, Carl Zehnder.
Integrating Database and Program Descriptions using an ER-Data Dictionary, by Robert Marti.
Database Services for Personal Computers linked by a Local Area Network.

September 1983, ETH Report No. 56, "System Programming In Modula-2: Mouse and Bitmap Display", J. Gutknecht.

"The Runtime Structure of Modula-2", by John Gough, Queensland Institute of Technology. December, 1983(?).

3. FEATURE ARTICLES

"A Sample Nucleus Written in Modula-2: A Case Study", Dr. J. Hoppe, Software Practive and Experience 10, p. 697-706, December 1979.

"Lilith: A Personal Computer for the Software Engineer", N. Wirth, Keynote from Proceedings of the 5th International Software Engineering Conference, p. 2-15, March 1981.

"Modula-2: Why It Matters", Tom DeMarco, The Yourdon Report, p. 1-2, March/April 1981.

"Modula-2", Dr. R. Henry, Microprocessor Software Quarterly 3, SWURCC, Bath, May 1981.

"Modula-2 Adds Concurrency to Structured Programming", N. Wirth & M. Schindler, Electronic Design, p. 117-122, July 23 1981.

"Modula-2 -- A Language for the Eighties", Volition Systems, November 1981.

"Modula-2 Moves From the Lab to the Market", Mini-Micro Systems, p. 153-156, May 1982,

"Ambiquities and Insecurities in Modula-2", D. Spector, SIGPLAN Notices, p. 43, August 1982.

"Modula-2 -- A Solution to Pascal's Problems", SIGPLAN Notices, p. 28, September 1982.

"Modula-2 - A Worthy Successor to Pascal", Joel McCormack Mini/Micro Computer Conference and Exhibition, Session 10/4, September 1982.

"Ada, Modula-2, and Modern Software Engineering", Mary & Bob Hofkin, Journal of Pascal and Ada, p. 17-22 & 36, Nov/Dec 1982.

"An Introduction to Modula-2 for Pascal Programmers", by Lee Jacobson & Bebo White, The Best of the Computer Faires, Vol. VII p. 343-350 (Computer Faire), Mar. 18-20, 1983.

"The Simplicity of Modularity", Allen Munro, Softalk, p. 209-219, March 1983.

"Modula Matters", Mary K. Hofkin, Journal of Pascal and Ada, p. 25, March/April 1983.

"Modula-2 - A Worthy Successor to Pascal", Joel McCormack & Richard Gleaves, BYTE, p. 385-395, April 1983.

"Modula-2: Steps Beyond Pascal" Northcon/83, #7 (Professional Program Session Record):
"A brief Comparison of Pascal and Modula II - An Implementor's Point of View", by David Coar, pg. 1-3.
"Teaching Modula-2 -- In The Spirit of Pascal", by James O. Baker, pg. 1-5.

"Modula Matters" by Mary K. Hofkin, Journal of Pascal and Ada, p.20, May/June 1983.

"Using Modula-2", by David Carlisle, Journal of Pascal and Ada, p. 44-48, May/June 1983.

"Modula-2: Alternative to Ada?" by Allen Munro, Softalk, p. 109-117, Vol 2, July 1983.

"Modula Matters" by Mary Hofkin, Journal of Pascal & Ada, p. 20-21, July/Aug 1983.

"A Message-Passing Operating System For Real-Time Applications", by A. H. Anderson, presented at IEEE Workshop on Real-time Systems, Niagria Falls, NY, August 11-12, 1983.

"Structured Languages Such as Modula-2 Boost Programmer Output", Computer Technology Review, p. 337-341, (rec'd Sept, 1983), Summer 1983.

"Structured language eases systems-programming tasks", Electronics, p. 131-134, Sept. 22, 1983.

"Modula-2: Pascal's powerful heir", Mini-Micro Systems, p. 183-186, Sept. 1983.

"Lexing and Parsing Modula-2" by David Spector, SIGPLAN Notices, p. 25-32 Vol. 18 No. 10 October 1983.

"Die Programmiersprache Modula-2", Informatik Bulletin, Nummer 39/Okt. 1982, (in German) rec'd Oct. 3, 1983.

"Experience With Modula-2", Professional Program Session Record 12, (Mini/Micro West '83), Nov 8-11:
"A Novice Programmer's View of Modula-2" by John M. Craig.
"Object Oriented Design with Modula-2" by Richard Wiener & Richard Sincovec.
"The Module Library - Providing Modula-2 Program Portability Across Operating Systems" by Joel J. McCormack.

"Dynamic Multidimensional Arrays in Modula-2" (Implementing general purpose matrix operations), by Dr. Richard S. Wiener, Journal of Pascal and Ada, p. 43-45, Nov/Dec 1983.

"Hermes: An Operating System For A Modula-2 Environment", by Eric Hirschy, Presented at 1983 Conference on Small and Personal Computers (ACM SIG SMALL) San Diego, CA December 7-9, 1983.

"Modula-2, An Overview", John Gough, AustMUG News, p. 2-4, received Dec. 21, 1983.

"The Structure and Use of Procedure Variables" by K.J. Gough, AustMUG News, p. 5-16, received Dec. 21, 1983.

"2-DEBNF Form Modula-2", by A. H. J. Sale (Department of Information Sciences, University of Tasmania), AustMUG News, p. 17-23, received Dec. 21, 1983.

"Modula-2: Part II", part of a 5 part series by Robert C. Platt, Washington Apple Pi, p. 26-27, Feb. 1984.

"Modula-2 Modules Similar to Ada Packages", by A. Winsor Brown, p.18-20 Software News, Feb. 1984.

"The Making of Modula-2", by Edward Joyce, (Languages), PC Magazine, p. 177-181, April 3, 1983.

"Comparing Pascal and Modula-2 as Systems programming languages", P. H. Hartel, Programming Languages and System Design, ed. J. Bormann, North Holland, 1983.

"All This and Modula-2", Mike Parr, Soft, p. 58-63, January 1984.

"Modula Matters: Information hiding in Modula-2", Mary K. Hofkin, Journal of Pascal and Ada, p. 20-21, Jan/Feb 1984.

"Modula-to-Ada Type Translation", J. W. L. Ogilvie, Journal of Pascal and Ada, p. 33-32, Jan/Feb 1984.

"Generic Sorting in Modula-2", Dr. R. S. Wiener, Journal of Pascal and Ada, p. 33-36, Jan/Feb 1984.

"Modula-2 for Pascal Programmers" by Edward Joyce, (Languages), PC Magazine, p. 193-197, April 3, 1984.

"An Introduction to Modual-2 for Pascal Programmers", by Hugh McLarty and David W. Smith, Dr. Dobb's Journal, p.22-27, May 1984.

"An Introduction to Modula-2" (A program for making and taking quizzes with Modula-2) by Stuart B. Greenfield, Journal of Pascal, Ada & Modula-2, p. 36-40, May/June Vol.3, No.3 1984.

"Modular Software Construction and Object-Oriented Design Using Modula-2", (A case study illustrating object-oriented desing in Modula-2) by Richard S. Wiener and Richard F. Sincovec, Journal of Pascal, Ada & Modula-2, p.41-47, May/June Vol.3, No.3 1984.

4. ARTICLES (Containing significant info on M2)

"First Commercial Modula-2 Compiler Eclipses Pascal", Electronic Engineering Times, p. 1, December 7, 1981.

"Implementation of Modula Language Runs on Apple IIs", IEEE MICRO.

"First Full-Scale Implementation of Modula-2 Aimed at Apple II Micro, Has 64kBytes RAM", Electronic Engineering Times, p. 18, March 15, 1982.

"It's Better Than Ada", Datalink, p. 1, April 1, 1982.

"Alternative to Apple II Pascal Problems", Computer Design Monthly, May 1982.

"A Prospectus", Journal of Pascal and Ada, p. 8, Sept/Oct 1982.

"Available Systems", Journal of Pascal and Ada, p. 12, Sept/Oct 1982.

"Release of Modula-2 Compiler", Journal of Pascal and Ada, p. 13, Sept/Oct 1982.

"First Commercial Implementation of Modula-2 Language Announced", Microtechnics, No. 2, 1982.

"Modula/R Report" Lilith Version, editor Manuel Reimer, LIDAS Memo 021-83 February 1983, rec'd oct. 3, 1983.

New Modula-2 Version Faster, Easier To Use, Pascal News, p. 24, Number 25, April 83.

"Users Column" by Jerry Pournelle, Byte, p. 298-324, May 1983.

"The Ideal Architechture for Modula-2" by Rod C. Schiffman, Journal of Pascal and Ada, p.41-42 & 48, May/June 1983.

"The User Goes to the Faire", by Jerry Pournelle, Byte, p. 306-330, June 1983.

"User's Column", by Jerry Pournelle, Byte, p. 411, June 1983.

"Life in Modula-2" by John Ogilvie, Journal of Pascal & Ada, p. 33-35 July/Aug. 1983.

"Jenseits von Pascal - Modula-2, eine neue Programmiersprache", Computer Personenlich, Aug. 24, 1983.

"Analysis: Modula-2 versus Pascal", Structured Language World, by Volition Systems, p. 6 Issue 5, Aug. '83

"The Family Tree Of Computer Languages", by Bernard Cole, Popular Computing, p. 82-88, September 1983.

"Computer Languages Of The Future", by Jonathan Amsterdam, Popular Computing, p.136-151, September 1983.

"Using procedure type variables", (Modula Matters), by Mary K. Hofkin, p. 20-21, Nov./Dec. 1983.

"Ada's Future in Industry and Education", (Ken Bowles Talks About Ada, Part II), p. 40-41, Nov./Dec. 1983.

"Buddy, Can You Spare a Door Latch?", (User's Column) by Jerry Pournelle, Byte, p. 59-76, December 1983.

"Pascal's Design Flaws Modula-2 Solutions and Pascal Patches" by Mark C. Johnson & Allen Munro, Byte, p. 371-388, March 1984.

5. M2 REVIEWS

Journal of Pascal and Ada, "Using Modula-2", by David Carlisle, p. 44-48, May/June Vol.2 No.3 1983.

Personal Computer News (UK), (PCN Pro-Test Software) "Modual-2: son of Pascal", June 10-15, 1983.

Sandy Apple Press, "Modula-2", by John W. Baxter, p. 2 & 19, Aug/Sept 1983, Vol.3 No.8.

InfoWorld, (Software Reviews) "Voliton Systems' Modual-2 programming language", p.55-58, Vol. 5, #38, Sept. 19, 1983.

AustMUG News 2, "Volition Systems' Modula-2 - an Informal Review", by Geoff Foster, p. 2-12, Received Dec. 21, 1983.

PC Magazine, "What's New with Modula-2?", by Joel Pitt (Languages), p. 187-191, April 3, 1984.

"Volition Systems Modula-2", Peter Ware, Soft, p. 67-72, March 1984.

A+ Magazine, "Modula-2 on the Apple Too", (programming) by Ken Glickfeld, p. 67-76, Vol.2, Issue6, June 1984.

Byte, "Volition Systems' Modula-2" by Eric Eldred,(Softeware Review), p. 353-364, June 1984.

(Last Revised 26-Jun-84, work partly supported by Volition Systems)

Modula-2 News

Issue # 1

January 1985

Modula-2 News is the quarterly, informal publication of

Modus, the Modula-2 Users Association.

CONTENT

The MODUS Quarterly

Issue # 2

April 1985

Modula-2 News for MODUS, the Modula-2 Users Association.

CONTENT

The MODUS Quarterly

Issue # 3

July 1985

Modula-2 News for MODUS, the Modula-2 Users Association.

CONTENT

MODUS QUARTERLY ISSUE NO. 4 NOVEMBER 1985

The MODUS Quarterly

Issue # 4 >> YOUR LAST ISSUE <<

November 1985 until you renew.

Modula-2 News for MODUS, the Modula-2 Users Association.

CONTENT

The State of Modus

Welcome to a new year! Modus has successfully completed its first year of publishing newsletters, and except for a slight delay with the final issue, I think we did an excellent job. I would like to thank all of you who contributed to the newsletter, and to encourage those of you who have not to please submit an article or a letter if you feel the subject is important to you.

Modus has also managed to sponsor a number of meetings during the year. Two of these were in the UK and Europe, with the final one being held in the USA. I only attended the meeting in the US, but from the reports I received, all of these meetings were quite successful and deserve a repeat performance sometime during this coming year.

During our initial year, the organization evolved in many ways, and grew to a size larger than our original projections. For those of you who are not familiar, Modus is a diverse international organization. We have headquarters in Palo Alto, California and Vevey, Switzerland. The organization shares the newsletter between the two headquarters as well as many mutual directors. But legally, there are two separate organizations, with the US portion being a California Corporation.

The membership of Modus has grown from just a handful at the beginning of last year to over 350 in the US and over 250 in Europe with the number growing every day. We have received a lot of cooperation from the companies producing and selling Modula-2 compilers which has helped spread information on the organization without requiring advertising by Modus. Modus does not plan to advertise in order to increase membership. We are interested in providing a means of communication for serious users and implementors of Modula-2, and information for those interested in the language. To meet this goal, compiler vendors notices, and word of mouth from you gets Modus information to potential members.

As far as the financial situation, I only have figures for the California Corporation. The Corporation is solvent and even slightly profitable. With income from Memberships at about $7,000 and payment from the European group for Newsletters that were sent over in bulk, there was a $950 profit at the end of our fiscal year (Sept. 30). There are also about 400 copies of each issue of the Newsletter left over for new members who want them.

I would like to take this opportunity to thank all of the people who have given their time to help Modus. Modus has been run completely by volunteers. Up to this time the only services we have paid for are our legal, accounting and printing services. I would like to give special thanks to Randy Bush and Tom DeMarco who spearheaded the effort on the US side to make the organization a reality. I would also like to thank all of the people at Pacific Systems Group who gave their time to make the make sure the membership database was in order, and the first three issues of the Newsletters were mailed out. I would also like to thank Logitech and Charmaine Bennett for taking over the job of mailing out the Newsletter. And most of all, I would like to thank Dick Karpinski for being the editor of this Newsletter. [Being is easy, doing is hard. rhk

I hope all of last years members enjoyed the Newsletters and received information that was useful to you. I also hope that the information that will be upcomming in this and future issues will be at least valuable.
But remember, that depends on you.

George J. Symons, Secretary USA

extract of [illegible]

The MODUS Quarterly

Issue # 5

February 1986

Modula-2 News for MODUS, the Modula-2 Users Association.

CONTENT

Computer Sciences Laboratory
Xerox Corporation
Palo Alto Research Center
Palo Alto, California 94304
415 494-4415

XEROX

31. July 85

Mr. John D. O'Meara
8900 42nd Ave. NE
Seattle WA 98115

Dear Mr. O'Meara,

This is in reply to you observation that Modula-2 lacks *open array parameters* with more than a single index.

I thank you for your suggestion and agree with you that this is a serious handicap when dealing with matrices. Fortunately the extension of the open array concept to more than one dimension is straightforward and lies entirely within the framework of Modula's concept. In fact, the only change required in the Report affects the syntax, namely (1) the last line on page 160 (3rd Ed, page 156 in 2nd Ed) in *Programming in Modula-2* and (2) rule 78 on page 172 (3rd Ed, rule 82, page 168, 2nd Ed) changes from

FormalType = [ARRAY OF] qualident.

to

FormalType = {ARRAY OF} qualident.

The associated interpretation (semantics) is quite obvious. One might add a sentence in section 10.1 on formal parameters after the syntax, explaining that the number of "ARRAY OF"s in the formal parameter specification must be identical to the number of index expressions in the actual parameter; but I think it is actually quite superfluous.

Note that this does not include the changes in the text of my book preceding the Report; but there are similarly few, and it is not the ultimate language specification anyway.

If this is felt to be essential, I should be happy to include this generalization in the next edition of the book. But you should be aware that individual implementors will still be free to update their compilers or to leave them with a new "restriction".

Sincerely yours,

N. Wirth

ETH EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE ZÜRICH

Institut für Informatik
Fachgruppe Computer-Systeme

Clausiusstrasse 55
Durchwahlnummer 01 256 22 26
Sekretariat 01 256 22 27

Postadresse:
Institut für Informatik
ETH-Zentrum
CH-8092 Zürich

1610

Dr. Richard Karpinski
Editor, MODUS Quarterly
MODUS, Modula-2 User's Assoc.
6521 Raymond St.

Oakland, CA 94609 / U S A

January 17, 1986

Dear Dr. Karpinski,

Thank your for your letter! No, I don't cover the topics on array parameters and integer arithmetic elsewhere, and you are free to reprint it. Perhaps the memo on arithmetic could be expanded into the following note:

Integer Arithmetic

The definition of the DIV and MOD operators in Modula-2 (and most other languages) is a persisting problem, and I believe that the standardization group and implementors should consider it seriously. In the report, x MOD y is undefined for negative y, and for negative x the definition is in contradiction with mathematics. This should not be!

The trouble is that most computers implement arithmetic on integers in the sense of Euler, whereas modulo arithmetic is the one that is relevant in mathematics and - I believe - in computing too. Unfortunately they differ. Since it hardly made much sense to define a programming language with basic operations differing from those of practically all available hardware, the definition of the MOD operator in Modula is valid for positive operands only, where the two arithmetics are the same.

Fortunately, there is now at least one computer available whose designers recognized the issue: the NS32000. It offers instructions for both arithmetics. Our new compiler reflects this situation, and I suggest to adopt this solution generally.

In Euler's arithmetic, let the quotient be denoted by Q = x/y and the remainder by R = x REM y. If $y \neq 0$, Q and R always satisfy the equation

$$Q*y + R = x$$

and either $0 \leq R < y$ or $0 \geq R > y$. Operations are symmetric with respect to zero, i.e. $(-x)/y = x/(-y) = -(x/y)$.

Modulo arithmetic is based on the idea of equivalence classes of integers. Each class can be identified by a specific member, for example its least non-negative member. For a given modulus y > 0, we create y classes, each class containing all integers q*y+r for arbitrary q, and its least, non-negative, identifying member being r, where $0 \leq r < y$. Hence we define q = x DIV y, and r = x MOD y, satisfying the equations

$$q*y + r = x \text{ and } 0 \leq r < y$$

Examples:

31/10 = 3	31 REM 10 = 1
-31/10 = -3	-31 REM 10 = -1
31 DIV 10 = 3	31 MOD 10 = 3
-31 DIV 10 = -4	-31 MOD 10 = 9

An example of the use of modulo arithmetic is the cyclic buffer of size n with indices in for the location of the next input and out for the next output. The number of filled slots is always (in-out) MOD n (and not (in-out) REM n).

If negative integers are represented by their two's complement a right shift of x by k positions represents the operation x DIV 2^k (and not $x/2^k$).

(end of note)

Concerning multi-dimensional open arrays, I do not have anything further to say.

With best regards,

N. Wirth

Prof. Niklaus Wirth

encl.

The MODUS Quarterly

Issue # 6

November 1986

Modula-2 News for MODUS, the Modula-2 Users Association.

CONTENT

The MODUS Quarterly

Issue # 7

February 1987

Modula-2 News for MODUS, the Modula-2 Users Association.

CONTENT

Without Delay Renew Today!

New Products

Modula-2

For IBM 370 VM/CMS. Developed by the Computer Systems Group at the University of Waterloo. Contact Sandra Ward, WATCOM Products Inc., 415 Phillip Street, Waterloo, Ontario, Canada, N2L 3X2, (519) 886-370, Telex: 06-955458.

EXE2LNK

For users of the Logitech compiler on IBM PC machines. Converts .EXE files to .LNK files used by the Logitech Modula-2 Linker. Used to bind assembly language routines with Modula-2 programs. Contact Leif Ibsen, Blommevangen 15, DK-2760 Maalov, Denmark.

M23

Third Edition Modula-2 for RT11SJ, RT11XM, and TSX+., plus other performance improvements. Contact: Dr. K. John Gough, School of Computing Studies, Queensland Institute of Technology, G.P.O. Box 2434, Brisbane, Queensland, Australia, 4001, Telex: 44699.

Modula-2PC

Modula-2 for IBM PC/PCjr/PCXT/PCAT or compatible. Contact: Peter Collier, PCollier Systems Inc., Suite 390, 7925-A North Oracle Road, Tucson, Arizona 85704, (800) 522-2060.

MacMeth

Modula-2 for 512K Macintosh or Macintosh Plus. Contact: Modula Corporation, 950 North University Avenue, Provo, Utah 84604, (800) 545-4842.

GEFI Modula-2

An adaptation of the ETH-Zuerich SMILERX, 68000 Modula-2 compiler to the Macintosh environment. Contact: Chantal Fauconnet, GEFI Service, 71 rue de la Victoire, 75999 Paris France, Tel: (1) 39 85 44 43.

Modula-2/RTS

Modula-2 for RT11 and SHAREplus. Contact: Guenter Dotzel, ModulaWare GmbH, Wilhelmstrasse 17A, D-8520 Erlangen, West Germany, Tel: 09131 208395.

Modula-2 for OS/MVS

A fast single pass compiler for IBM Mainframes. Generates native 370 code for OS linker and loader. Contact: G. Blaschek, University of Linz, Institut fuer Informatik, Altenbergerstrasse 69, A-4040 Linz, Austria, Tel: (0732)-232381-447

Modula-2/68
Modula-2/68-CD

A Modula-2 language system for use with the MC68000 family of computers. Native Modula-2 development on many of the popular 68000 based systems. Many host systems are supported. CD is Modula-2/68 cross development on VAX computers. CD runs on VAX/VMS, Ultrix and BSD 4.2/4.3 operating systems. Contact: Stan Osborne, Djavaheri Bros., P.O. Box 4759, Foster City, California 94404-0759, Tel: (415) 341-1768, Telex: 4949940.

Modula-2 VM/CMS (IBM 370)

Developed at TU Berlin. Contact: Thomas Habernoll, TU Berlin, Informatik Rechnerbetrieb, Sekr. FR 5-3, Franklinstr. 28/29, 1000 Berlin 10, West Germany.

Modula-2

Modula-2 and Advanced Systems Editor from Pecan Software Systems, Inc. Contact: Y. A. Lifschutz, 1410 39th Street, Brooklyn, NY 11218, Tel: (718) 851-3100.

The MODUS Quarterly

Issue # 8

May 1987

Modula-2 News for MODUS, the Modula-2 Users Association.

CONTENT

The MODUS Quarterly

Issue # 9

October 1987

Modula-2 News for MODUS, the Modula-2 Users Association

CONTENT

MODUS Conference June 15 & 16, 1987

Seven Hills Conference Center, San Francisco, California.

Monday, June 15, 1987

Introduction, George Symons, Stan Osborne
Standardization Panel, Moderator: Dick Karpinski
A discussion of Opaque Types, Chuck Bilbe
Testing Floating Point Implementations, Dick Karpinski
Converting Big UCSD Pascal Applications to M2, Jon Bondy
Writing Portable Applications,Morris Djavaheri
Building Large Applications and Support Tools, Ed de Young
Adding High Level Concurrency Features, Mike Meehan

Tuesday, June 16, 1987

Modula-2 for NS32000 Embedded Systems, Peter Ashenden
Modula-2/370, Experiences with M2 as a portable Systems Language, Jeffrey Savit
ISO WG13 Proposed Concurrent Programming Libraries, Stan Osborne
Design Language From Modula-2, "MODEL", Frode Odegard
Form-maker, a screen/form generator for M2, David Rhoads
Building a Network Simulator, Paul Labbe'
Using the ISO Networking model for communication, Andy Bierman

Technical Demos

Andy Bierman, Two IBM PC/AT's communicating by serial cable.

Workman & Associates, IBM PC & Atari ST demos.

Abstracts

High Level Language Concurrency Features for Distributed Computation

Mike Meehan, The University of Alabama in Huntsville.

The analysis and formulation of concurrent programming constructs suitable for implementation in high level languages targeted for distributed programming environments is discussed. Concurrent programming abstractions in common use in high level languages are analyzed. An alternative formulation called the dynamic monitor is given. The dynamic monitor concept is demonstrated through incorporation into a high level programming language, M3. M3 is derived from Modula-2. M3 is a superset of Modula-2 designed for distributed computation in loosely coupled network environments. A compiler, for this language, generating instructions for a local area network of micro-computers is presented.

Modula-2 for NS32000 Embedded Systems

Peter Ashenden, Department of Computer Science, University of Adelaide, Australia.

The development of a cross-support system for the Modula-2 language is discussed. The cross-compiler runs under the VAX/VMS operating system, generating code for the NS32000 processor; it was used in the development of embedded software for the QDS-1000 workstation. Code generated by the Modula-2 cross-compiler can be run either stand-alone or under the EXEC run-time executive supplied by National Semiconductor. The development of the latter version provided some examples of situations where EXEC strongly supported the kind of run-time environment required for Modula-2, as well as giving rise to some difficulties because of interference between EXEC and the Modula-2 run-time organization.

Modula-2/370, Experiences with Modula-2 as a portable systems language.

Jeffrey Savit, Savvy Computing, Inc., Upper Montclair, New Jersey.

A discussion of the Modula-2/370 implementation and the decisions made during development. Also discussed are aspects of the language and common programming practices that assist or impede portability.

MODEL - a Modula-2 Design Language

Frode Odegard, Modula-2 CASE Systems A/S, Jessheim, Norway.

Introduces MODEL, a notation which lets designers divide projects into subsystems. Each subsystem consists of modules. The language lets you make modules private to their subsystem, prevent unwanted dependencies and set up forced dependencies on the module level. The main point is to help designers enforce further rule upon the implementors, other than those offered by Modula-2. Finally, a MODEL-based database used in the PEM system is described.

Building a Network Simulator

Paul Labbe', Communications Research Centre, Ottawa, Ontario, Canada.

Building a network simulator for mobile platforms provides a valuable experimental environment for the control and observation of an event-driven simulation. To cope with the evolution of the communications systems being simulated a simulator was needed that could be easily modified. The use of abstract data types, information hiding, and separate compilation allows building the components of the simulator concurrently. Tools were developed to ease building and testing the simulator. These tools and the Modula-2 compiler are used to guarantee the consistency of the simulator during its development.