Copyright © 1995 by HOOD User’s Group

K 10.1.1 Client Side <ObjectName> structure -- ER code Illustration . . A113

K 10.1.2 Request Broker <ObjectName>_RB structure -- SER code illustration A114

K 10.1.3 Object_Server code Illustration A117

K 10.1.4 VN IMPLEMENTATION ILLUSTRATION A119

K 10.2 C++code illustrations A120

K 10.2.1 Client Side <ObjectName> structure -- ER code Illustration in C++ A120

K 10.2.2 / Request Broker <ObjectName>_RB structure -- SER code illustration in C++ A121

K 10.2.3 Object_Server code Illustration in C++ A124

K 10.3 OSTM Illustration of Stack A125

K 2.2 HRTS.FSMS sample code A69
K 2.2.1 G_FSM Ada Code Illustration A69
K 2.2.2 G_FSM C++ Code Illustration A71
K 3 OBJECT OBCS A80
K 3.1 Object GENERIC OBCS IS A81
K 3.2 HRTS.OBCS Ada sample code A82
K 3.2.1 package TQPool A82
K 3.2.2 Class TClientObcs in Ada A84
K 3.2.3 package TServerObcs in Ada A85
K 3.3 HRTS.OBCS C++ sample code A86
K 3.3.1 Class TIOStream in C++ A86
K 3.3.2 Class TRtnQ - C++ return queue for IPC communication A89
K 3.3.3 Class TErQ -Requets Queue for IPC A90
K 3.3.4 Class TClientObcs in C++ A92
K 3.3.5 Class TServerObcs in C++ A94
K 4 OBJECT TMSG A96
K 4.1 Class Object TMsg IS A96
K 4.2 HRTS.TMsg C++ sample code A97
K 4.3 HRTS.TMsg Ada sample code A98
K 5 OBJECT VNCS A101
K 5.1 VNCS sample code A101
K 6 OBJECT HRTS_PE A102
K 6.1 Object HRTS_PE IS A102
K 6.2 HRTS.HRTS_PE sample code in Ada A103
K 6.3 HRTS.HRTS_PE sample code in Ada A104
K 7 OBJECT SEMAPHORES A105
K 7.1 Object SEMAPHOREs IS ACTIVE A105
K 7.2 HRTS.SEMAPHORE sample code A106
K 8 OBJECT TIMER A107
K 8.1 Object TIMER IS ACTIVE A107
K 8.2 HRTS.TIMER sample code A108
K 9 OBJECT HRT_SCHEDS A109
K 9.1 Object HRT_ SCHEDs IS A111
K 9.2 HRT_ SCHEDs Sample Code A112
K 10 TARGET CODE ILLUSTRATIONS FOR PROTOCOL CONSTRAINED OPERATIONS A113
K 10.1 Ada code illustrations A113

Copyright © 1995 by HOOD User's Group

G 4 PRAGMA FOR TESTING SUPPORT A19
G 4.1 Pragma OP_TEST A19
G 4.2 Pragma OTS A20
G 5 PRAGMA ODS_ANNOTATION A22
G 6 PRAGMA GRAPHICAL_ANNOTATION A22
G 7 PRAGMA LOCATION A22
G 8 PRAGMA PROTECTED A24
G 9 PRAGMA IPC A24
H STANDARD INTERCHANGE FORMAT A25
H 1 GENERAL A25
H 2 LEXICAL ELEMENTS A25
H 2.1 Character Set A25
H 2.2 Lexical Elements, Separators, and Delimiters A26
H 2.3 Identifiers A26
H 2.4 Numeric Literals A27
H 2.5 String Literals A27
H 2.6 Comments A27
H 2.7 Reserved Words A27
H 2.8 Pragmas A27
H 2.9 Free text and target dependent fields A28
H 2.10 Annotations A28
H 2.11 Allowable Content of Sections A28
H 3 SYNTAX SUMMARY A28
H 4 EXAMPLE OF YACC IMPLEMENTATION A42
I ODS DEFINITION A51
J OSTM AND OSTD DESCRIPTIONS AND SYNTAX A55
J 1 OSTD SYNTAX A55
J 2 OSTM SYNTAX A57
J 2.1 OSTM Code Illustration in Ada A57
J 2.2 OSTM code illustration in C++ A58
K HOOD RUN TIME SUPPORT LIBRARY A59
K 1 OBJECT EXCEPTIONS A60
K 1.1 Object EXCEPTIONS IS A61
K 1.2 HRTS.EXCEPTIONS sample code A62
K 2 OBJECT FSMS A68
K 2.1 Object FSMs IS A68

Copyright © 1995 by HOOD User's Group

D 6 GENERIC A11
D 7 CONSTRAINED OPERATION A11
D 8 CONTROL FLOW A11
D 9 DATA FLOW A12
D 10 DESIGN PROCESS A12
D 11 ENVIRONMENT OBJECT A12
D 12 EXCEPTION FLOW A12
D 13 FORMAL PARAMETERS A12
D 14 HDT (HOOD DESIGN TREE) A12
D 15 INCLUDE RELATIONSHIP A12
D 16 INSTANCE A12
D 17 INTERNAL OPERATION A13
D 18 NON-TERMINAL OBJECT A13
D 19 OBJECT DESCRIPTION SKELETON - ODS A13
D 20 OBJECT CONTROL STRUCTURE - OBCS A13
D 21 OBJECT STATE TRANSITION DIAGRAM - OSTD A13
D 22 OPERATION CONTROL STRUCTURE - OPCS A13
D 23 OP_CONTROL OBJECT A13
D 24 OPERATION_SET A13
D 25 PASSIVE OBJECT A14
D 26 PHYSICAL NODE A14
D 27 PROVIDED INTERFACE A14
D 28 REQUIRED INTERFACE A14
D 29 ROOT OBJECT A14
D 30 SYSTEM CONFIGURATION A14
D 31 TERMINAL OBJECT A14
D 32 USE RELATIONSHIP A14
D 33 VIRTUAL NODE A14

E ABBREVIATION LIST A15

F RESERVED WORD LIST A17

F 1 HOOD RESERVED WORDS A17
F 2 HOOD AND ADA RESERVED WORDS A18

G HOOD PRAGMAS A19

G 1 PRAGMA TARGET_LANGUAGE A19
G 2 PRAGMA NOMUTEX A19
G 3 PRAGMA HCS A19

Copyright © 1995 by HOOD User's Group

18.5.1.1 Modular Refinement and BreakDown / Include Refinement 145
18.5.1.2 ADTs Identification and Interface Refinement 146
18.5.1.3 Class Refinement through Attribution and Inheritance 148
18.5.1.4 Combining ADTs with USE and INCLUDE 149
18.5.2 Implementation Refinement 151
18.5.3 Technology Component Architecture Definition 152
18.5.4 Generic Architecture for Information Systems 153
18.5.5 Development Approach for Complex Systems 153

A APPENDIXES A1

B REFERENCES A1

C HOOD GRAPHICAL SYMBOLS A5

C 1 PASSIVE OBJECT A5
C 2 ACTIVE OBJECT A5
C 3 CONSTRAINED OPERATION A5
C 4 OP_CONTROL A5
C 5 PASSIVE CLASS A5
C 6 ACTIVE CLASS A5
C 7 GENERIC OBJECT A6
C 8 GENERIC CLASS A6
C 9 PASSIVE OBJECT INSTANCE A6
C 10 PASSIVE CLASS INSTANCE A6
C 11 ACTIVE OBJECT INSTANCE A6
C 12 MULTIPLE INSTANCE OF PASSIVE OBJECTS OR CLASSES A7
C 13 MULTIPLE INSTANCE OF ACTIVE OBJECTS OR CLASSES A7
C 14 VIRTUAL NODE A7
C 15 USE RELATIONSHIP A7
C 16 DATAFLOW A8
C 17 EXCEPTION FLOW A8
C 18 INCLUDE RELATIONSHIP (CLIENT_SERVER VIEW) A8
C 19 INCLUDE RELATIONSHIP (STRUCTURAL VIEW) A9
C 20 OSTD GRAPHICAL FORMALISM A10

D GLOSSARY A11

D 1 ACTIVE OBJECT A11
D 2 CLASS A11
D 3 ABTRACT CLASS A11
D 4 ACTUAL PARAMETERS A11
D 5 GDT (GENERIC DESIGN TREE) A11

Copyright © 1995 by HOOD User’s Group

17.2.3.2.1 OPCS_ER for HSER constraint : 111
17.2.3.2.2 OPCS_ER for LSER or RLSER constraints. 111
17.2.3.2.3 OPCS_ER for ASER , RASER constraints, ... 111
17.2.3.2.4 OPCS_ER for LSER_TOER constraint 112
17.2.3.2.5 OPCS_ER for HSER_TOER constraint : 112
17.2.3.2.6 OPCS_ER for ASER_BY_IT constraint : 112
17.2.3.3 OPCS _SER Specifications 113
17.2.3.3.1 OPCS_SER for OBJECTS. 113
17.2.3.3.2 OPCS_SER for Classes. 114
17.2.3.3.3 <OBJECT>_RB Target code outline 114
17.2.4 Active Class Implementation Support 116
17.3 VIRTUAL NODE IMPLEMENTATION 117
17.4 HOOD PRAGMAS 120
17.5 ADA TARGETS 120
17.5.1 Introduction 120
17.5.2 Kernel Extraction Rules 120
17.6 C++ AND C TARGETS 125
17.6.1 Introduction 125
17.6.2 Kernel Extraction Rules 126
17.7 OTHER TARGETS 131
18 THE HOOD DESIGN PROCESS 132
18.1 SYSTEM TO DESIGN 132
18.2 SYSTEM CONFIGURATION 133
18.3 THE BASIC HOOD DESIGN PROCESS 136
18.3.1 Problem definition 137
18.3.1.1 Activity: Understand the problem to solve before jumping on a solution 137
18.3.1.2 Activity Outputs 137
18.3.2 Elaboration of an Informal Solution Strategy 138
18.3.2.1 Activity: Refine and work out a solution 138
18.3.2.2 Activity Outputs 138
18.3.3 Formalisation of the strategy 138
18.3.3.1 Activity: Refine and work out the selected solution outline . 138
18.3.3.2 Activity Outputs 138
18.3.4 Formalisation of the solution 139
18.3.4.1 Activity: Formalise the reviewed solution in the ODSs 139
18.3.4.2 Activity Outputs: ODSs 139
18.3.5 Analysis of the solution 140
18.3.5.1 Activity: review and justify all design decisions 140
18.3.5.2 Activity Outputs 140
18.4 THE OVERALL HOOD DESIGN PROCESS 141
18.5 DESIGN INTEGRATION APPROACH FOR MULTI-TECHNOLOGY DEVELOPMENTS 144
18.5.1 Modular and ADT Refinement Principles 145

14.11 GENERIC INSTANCE ODS 77
14.12 TERMINAL VN ODS 78
14.13 NON TERMINAL VN ODS 79
14.14 DESCRIPTION OF TYPES, CLASSES, CONSTANTS AND DATA 80
14.14.1 Specification of Types 80
14.14.2 Specification of Classes 80
14.14.3 Description of Constants 81
14.14.4 Description of Data 81
14.15 DESCRIPTION OF OPERATION_SET 82
14.16 DESCRIPTION OF EXCEPTIONS 82
14.17 DESCRIPTION OF OBCS 83
14.17.1 Description of OBCS Visible Part 83
14.17.2 Description of OBCS Internal Parts 84
15 DESIGN CHECKING, SCOPING, VISIBILITY AND HOOD RULES . . 86
15.1 VISIBILITY AND SCOPING 86
15.2 HOOD RULES 88
16 HOOD RULES 91
16.1 GENERAL DEFINITIONS 91
16.2 INCLUDE RELATIONSHIP 93
16.3 USE RELATIONSHIP 94
16.4 INHERITANCE AND ATTRIBUTE RELATIONSHIPS 94
16.5 OPERATIONS 95
16.6 PROVIDED INTERFACE 95
16.7 SCOPING AND VISIBILITY 96
16.7.1 Naming 96
16.7.2 Visibility 96
16.8 CONSISTENCY AND COMPLETENESS 97
16.9 REQUIRED INTERFACE 99
16.9.1 General 99
16.9.2 GENERIC definition 100
16.9.3 GENERIC instance 100
17 IMPLEMENTATION FOR TARGET ENVIRONMENTS 101
17.1 CODE GENERATION PRINCIPLES 101
17.2 MULTI-TARGET CONSTRAINED OPERATION SUPPORT 102
17.2.1 Implementation of State and Concurency constrained operations 103
17.2.2 Implementation of Protocol and Time constraints 104
17.2.3 OPCS TARGET CODE PARTs Specifications 105
17.2.3.1 State and Concurency Constraint Support 105
17.2.3.1.1 OPCS HEADER specification 107
17.2.3.1.2 OPCS FOOTER specification 108
17.2.3.2 OPCS_ER specification 109

9.1 HOOD CLASS DEFINITION 34
9.1.1 Class Graphical Representations 35
9.1.2 Type-use Relationships 35
9.1.3 Class Inheritance Definition 36
9.1.4 Attribution Definition 36
9.1.5 Graphical representation of Class libraries 38
9.2 CLASS INSTANCE DEFINITION 40

10 GENERICS 41
10.1 GENERIC DEFINITION 41
10.2 GENERIC INSTANCE DEFINITION 43

11 REAL TIME SYSTEMS 45
11.1 DEFINITIONS 45
11.2 REAL TIME SYSTEMS VS HOOD OBJECT MODEL 46
11.2.1 Resources and Passive Objects 47
11.2.2 Real Time Architecture and HOOD Design 48

12 VIRTUAL NODES 49
12.1 DISTRIBUTED SYSTEMS 49
12.2 HOOD CONCEPTS 50
12.2.1 VN Definition 51
12.2.2 VN Implementation Principles 52
12.2.2.1 Surrogate Objects 52
12.2.2.2 Multi-Target Code Structure 53
12.2.2.3 Target Code Optimization Principles 55
12.3 DEFINING VN HIERARCHIES 57
12.3.1 Partitioning Process 57
12.3.2 Partitioning Rules 59
12.3.3 Configuration Rules 59

13 CONCEPTS SUMMARY 61

14 TEXTUAL FORMALISM 62
14.1 DESCRIPTION 62
14.2 TERMINAL ODS ILLUSTRATION 63
14.3 NON TERMINAL ODS 65
14.4 ENVIRONMENT ODS 65
14.5 ROOT ODS 65
14.6 OP_CONTROL ODS 65
14.7 CLASS ODS-C++ ILLUSTRATION 66
14.8 CLASS ODS ADA ILLUSTRATION 69
14.9 GENERIC ODS 74
14.10 GENERIC DESCENDANT ODS 76

ABSTRACT . . . . **II**

ACKNOWLEDGEMENTS . . . . **IV**

OBSERVATION REPORTS . . . . **VI**

OR FORM . . . . **VII**

1 INTRODUCTION **1**

2 HOOD OVERVIEW **3**

2.1 BASIC CONCEPTS 3
2.2 SYSTEM DESIGN 5

3 OBJECT AND OPERATION CONCEPTS **6**

3.1 STATIC PROPERTIES 6
3.2 DYNAMIC PROPERTIES 7
3.2.1 Communication between Objects 7
3.2.2 Control Flow and Dynamical Behaviour 8
3.2.3 Constrained Operations 8
3.2.3.1 State Constraints Description 10
3.2.3.2 Concurency constraints descriptions 12
3.2.3.3 Protocol constraints descriptions 13
3.2.3.4 Time constraints descriptions 15
3.2.4 Control Structures and Target Models 16
3.2.4.1 Definitions 16
3.2.4.2 Implementation Principles 16
3.2.5 ACTIVE and PASSIVE Objects 19
3.3 OBJECT DESCRIPTION SKELETON 20

4 THE USE RELATIONSHIP **23**

5 THE INCLUDE RELATIONSHIP **24**

5.1 HOOD DESIGN TREE 24
5.2 OPERATION BREAKDOWN 26
5.2.1 One to one Mapping 26
5.2.2 One to many Mapping 26
5.2.3 Overloaded Operations 27
5.2.4 Unconstrained/Constrained Operation Mapping 27
5.2.5 Operation_sets 28
5.2.6 Internal Operations 29
5.3 TYPES AND CLASS REFINEMENT 29

6 DATA FLOWS **30**

7 EXCEPTION FLOWS **31**

8 ENVIRONMENT OBJECT **32**

9 CLASSES **33**

Copyright © 1995 by HOOD User's Group

OR FORM

OBSERVATION REPORT (OR) From

Document under review : *HOOD REFERENCE MANUAL* *Release 4.*	**Para:**	**Page:**	**No:**

Subject:

Discrepancy:

Recommendation: **Urgency:** □ L □ M □ H

OBSERVATION REPORTS

This book is a **Reference Manual** having as goals to describe unambiguously the HOOD concepts and their implementation primarily on ADA and C++ targets. We have tried to illustrate much of the concepts by taking the well known STACK example, which is refined in several variants along the book, from a basic Abstract Data Type implementation to a client-server distributed class implementation.

We have a HOOD DESIGN PROCESS description in the core of the manual, whereas, Appendixes now contain most of the stuff regarding toolset implementors or Quality Assurance. A number of target code illustrations are also given, with most of the examples being compiled. Full illustration of VN code generation is still missing because of not yet available infrastructure tools; we hope to add in the final release of this manual for publication within the next months.

Finally, the work and experience gathered in this book is the result of a common effort supported by all HTG members and some university people. Merging all the proposals and concepts in an orthogonal consistent construct is a huge task, and I apologize for remaining inconsistencies. Therefore submission of critical comments for modification or extension to this HOOD 4 definition, would be highly appreciated. However we would like them being sent via e-mail to the following address **before November 15tht 1995:**

heitz@cisi.cnes.fr

If you do not have e-mail access, please send the comments using the special Observation Report (OR) form given in section: OR FORM on page vii below, to HOOD TECHNICAL GROUP at the following address:

Attn HTG-Chairman

Maurice HEITZ,
CISI
13, rue Villet
31400 TOULOUSE - FRANCE
fax :33 - 61.17.66.96
email : heitz@cisi.cnes.fr

NB : The HUG is setting up a HOOD FORUM on the Internet with sponsorship from ESA/ESTEC, and possibly a MOSAIC repository located at ESTEC. As soon as the HOOD FORUM will be accessible on the internet information on how to deliver comments on HRM4 shall be available in the header MOSAIC page.

Reviewers of HRM included :

Pierre DISSAUX	[TNI]
Serge BRIERE	[CAP_SESA]
Jean Marie WALLUT	[CNES]
Jean Pierre GERME	[3IP]
Olivier ROTH	[Sema]
Jerome DESQUILBET	[RATIONAL]
Sylvain WALLEZ	[ALCATEL]
Gilles MOTET	[INSA]
Miche PLANTIE	[EERIE]
Theodor TEMPELMEIER	[FH ROSENHEIM]
Marco BARRERA	[ALENIA-Spazio]
Jean BEZIVIN	[Univ NANTES
Philippe PERRIN	[EDF]
Khader SAHRAOUI	[LAAS]
Bernard COULETTE	[ENSEEIHT]

- and several others HOOD teachers (IRIT, YORK, EPFL, ENST) to be completed on final version

M. HEITZ Editor

ACKNOWLEDGEMENTS

The assistance provided for meetings of the HOOD TECHNICAL GROUP by CISI, CNES, CRI, DORNIER, ESA, INTECS, MATRA MARCONI SPACE, SEMA GROUP, TNI and TRASYS is gratefully acknowledged : without their support this work could not have been undertaken. We are also grateful for the support of the employers of the HOOD TECHNICAL GROUP members and other organisations that provided valuable feedback for the HOOD definition.

Particular thanks are due to CNES and CISI, whose support made possible the production of this book.

On a more personal note, we are especially grateful to the authors of observation reports on earlier definition of the HOOD method, and to all members of the HOOD TECHNICAL GROUP who contributed with lots of technical notes and multiple and detailed discussions. Especially we want to thank Pierre DISSAUX from TNI for his constructive contributions and discussions.

Attendee of HOOD TECHNICAL GROUP included :

Edouard ANDRE	[Sema Group]	
Jorge.AMADOR-MONTEVERDE	[ESTEC]	
Bernard DELATTE	[CNES]	
Rainer GERLICH	[DORNIER]	
Maurice HEITZ	[CISI]	(HTG chairman)
Tim OTSKOV	[CRI]	
Patrice MICOUIN	[STERIA]	
Jean François MULLER	[MMS]	(HTG secretary)
Alain Paul ANDRY	[TRASYS]	
Giancarlo SAVOIA	[INTECS]	

We also acknowledge gratefully the support of Jardine BARRINGTON-COOK from LOGICA, chairman of the HOOD STEERING GROUP, and the HSG secretary Guy PAQUET from SPACEBEL, who provided the necessary liaison between the HOOD TECHNICAL GROUP and the HOOD USER GROUP. Finally we acknowledge gratefully early reviewers who provided valuable feedback for improving the writing and understanding of this manual.

Initially developed for Ada83 program developments, HOOD is now targeting Ada95 as well as more classical and object oriented languages and systems. HOOD puts the emphasis on *interface and behaviour* definition and mastering, some issues that were rather neglected by other design methods. Moreover HOOD appears more and more as the framework for mastering and integrating the development of software components which may be developed with different target language and technologies (Man Machine Interface generators, Data Base Management System interface generators, rule systems and Knowledge Based System generators).

HOOD is thus supporting complex programming and development in the large, relying on code generator technology from high level and when possible, formal notations.

As a result HOOD fills primarily the needs of prime contractors and integrators. Providing a **standard inter-change format**, the HOOD method addresses also pragmatic **reuse**, tool **inter-operability** and **design perenniality**.

HOOD is thus the method of choice for large, long-lived projects where reuse, reliability and maintenability are key issues.

HOOD™ is a registered Trademark of the HOOD User Group. This fact must be stated in any publication referencing the name of HOOD in the context of the HOOD method as the basis of the publication.

ABSTRACT

This document defines the HOOD method for Architectural Design, Detailed Design and coding for software to be developed in programming languages such as Ada, C, or FORTRAN, as well as in object oriented languages such as C++ , Ada95 or Eiffel.

HOOD is a method of hierarchical decomposition of the design into software units based on identification of objects, classes and operations reflecting problem domain entities or more abstract objects related to digital programming entities.

The HOOD method comprises textual and associated diagrammatic representations allowing formal refinement, automated checking, user customizable documentation generation and target language source code generation.

The HOOD method was developed in 1987 under European Space Agency (ESA) contract A0/1-1890/86/NL/MA by a consortium of CISI, CRI A/S and Matra Marconi Space. HOOD has been selected by ESA projects as the design method for the Architectural Design phase. Since, HOOD is being more and more selected by large, complex or long lived project from aerospace, defence and industry.

Since 1989 the HOOD Manuals have been developed, in response to user experience, by the HOOD Working Group comprising representatives of ESTEC, Columbus and Hermes projects and later under control by the HOOD USER GROUP (HUG).

In 1991, the HUG was setup as a non profit organisation aiming to provide support for sharing experience and to control the evolution of the method. The HUG is organized in a STEERING GROUP (HSG) in charge of administrative issues, and in a TECHNICAL GROUP (HTG) in charge of all technical issues possibly delegating work to specific WORKING GROUPs.The HUG is based at:

C/O Spacebel Informatique,

11,rue Colonel Bourg

B-1140 BRUSSELS, Belgium

Tel: (32) 2 730.46.11

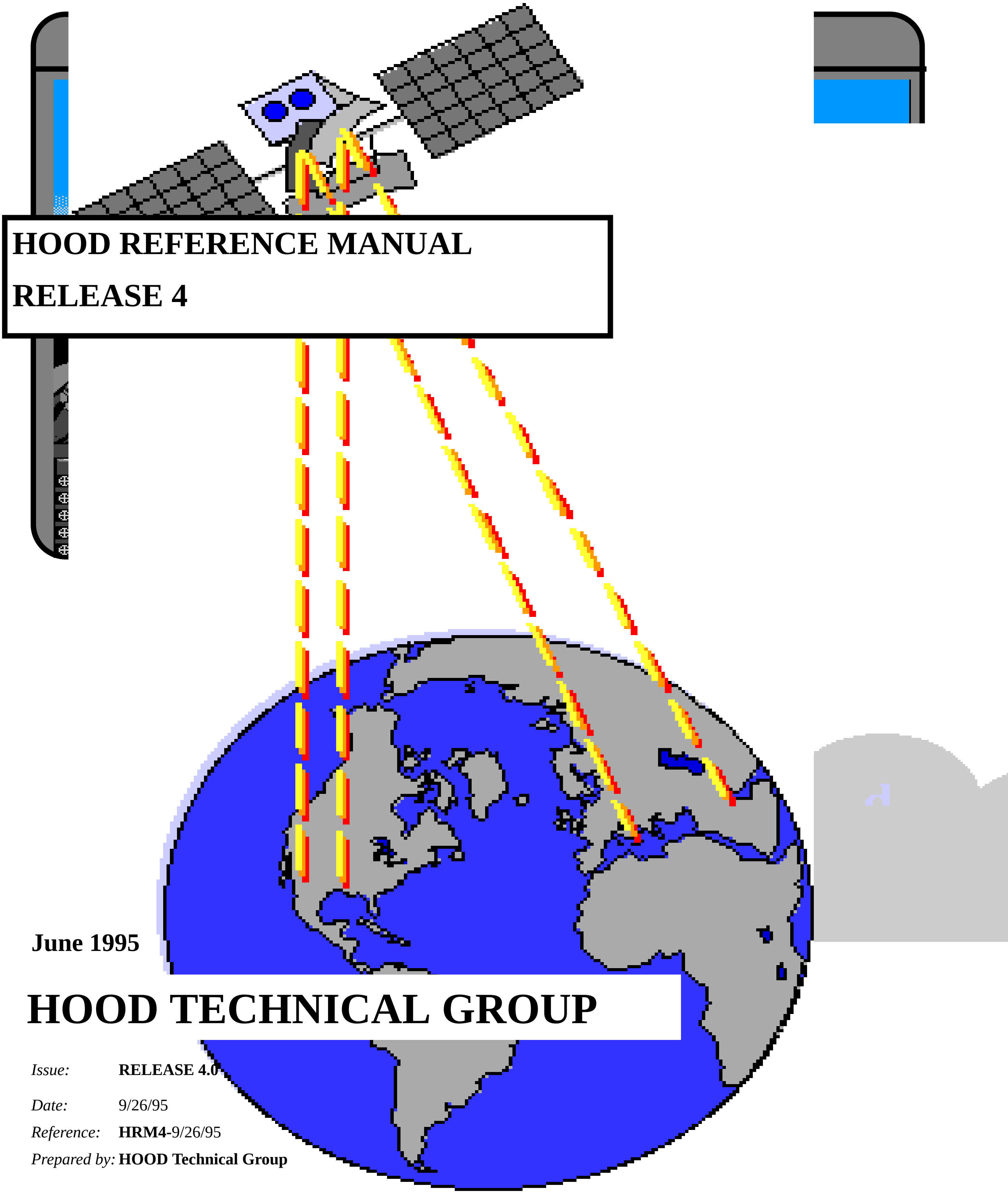
Fax: (32) 2 736.80.13

e-mail : hug@spacebel.be

The HOOD REFERENCE MANUAL Issue 3.1 was further developed in 1991 by the HOOD TECHNICAL GROUP and approved for two years by vote by the HOOD USER GROUP at the Pisa (Italy) april 3rd 1992 HUG meeting. The present document HOOD REFERENCE MANUAL Issue 4 was further developed in 1994 by the HOOD TECHNICAL GROUP and approved for two years by vote by the HOOD USER GROUP at the annual HUG meeting in February 1995.

Copyright © 1995 by HOOD User’s Group

HOOD REFERENCE MANUAL
RELEASE 4



June 1995

HOOD TECHNICAL GROUP

Issue: **RELEASE 4.0**

Date: 9/26/95

Reference: **HRM4**-9/26/95

Prepared by: **HOOD Technical Group**

Copyright © 1995 by HOOD User's Group