

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

161

DIANA An Intermediate Language for Ada

Revised Version

Edited by G. Goos, W.A. Wulf
A. Evans, Jr., and K.J. Butler



Springer-Verlag
Berlin Heidelberg New York Tokyo 1983

Editorial Board

D. Barstow W. Brauer P. Brinch Hansen D. Gries D. Luckham
C. Moler A. Pnueli G. Seegmüller J. Stoer N. Wirth

Editors

Gerhard Goos
Universität Karlsruhe, Institut für Informatik II
7500 Karlsruhe 1, FRG

William A. Wulf
Arthur Evans, Jr.
Kenneth J. Butler
Tartan Laboratories, Inc.
477 Melwood Ave., Pittsburgh, PA 15213, USA

DIANA was maintained and revised by TARTAN Laboratories Inc. for the ADA Joint Program Office of the Department of Defense under contract number MDA903-82-C-0148 (expiration date: 1983 February 28). The Project Director of DIANA Maintenance for TARTAN was Arthur Evans, Jr.

ISBN 3-540-12695-3 Springer-Verlag Berlin Heidelberg New York Tokyo
ISBN 0-387-12695-3 Springer-Verlag New York Heidelberg Berlin Tokyo

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to "Verwertungsgesellschaft Wort", Munich.

© by Springer-Verlag Berlin Heidelberg 1983
Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2145/3140-543210

TABLE OF CONTENTS

Abstract	1
Acknowledgments	3
Preface	5
1. Introduction	7
1.1. Design Principles	8
1.1.1. Original Design	9
1.1.2. Principles Governing Changes	11
1.1.3. What is a 'DIANA User'	12
1.1.4. Specification of DIANA	14
1.2. Structure of the Document	16
1.3. Attribution Principles of DIANA	17
1.3.1. Static Semantic Information	18
1.3.2. What is 'Easy to Recompute'?	19
1.3.3. Other Principles	20
1.3.4. Examples	20
1.4. Notation	21
1.4.1. Example of the IDL Notation	25
1.4.2. Specification of Representations	25
1.4.3. Example of a Structure Refinement	28
1.4.4. DIANA Notational Conventions	29
2. Definition of the Diana Domain	31
3. Rationale	79
3.1. Comparison with the Abstract Syntax Tree	80
3.1.1. Semantic Distinctions of Constructs	80
3.1.2. Additional Concepts	81
3.1.3. Tree Normalizations	81
3.1.4. Tree Transformation According to the Formal Definition	82
3.1.5. Changes to the AST	82
3.2. Consequences of Separate Compilation	83
3.2.1. Forward References	84
3.2.2. Separately Compiled Generic Bodies	84
3.3. Name Binding	85
3.3.1. Defining Occurrences of Identifiers	85
3.3.2. Used Occurrences of Identifiers	86
3.3.3. Multiple Defining Occurrences of Identifiers	87
3.3.4. Subprogram Calls	87

3.4. Treatment of Types	89
3.4.1. Machine Dependent Attributes	90
3.4.2. Type Specifications	90
3.4.2.1. Structural Type Information	90
3.4.2.2. Subtype Specifications	91
3.4.2.3. Derived Types	93
3.4.2.4. Incomplete and Private Types	96
3.4.2.5. Anonymous Array Types	96
3.4.2.6. Anonymous Derived Types	96
3.4.3. Type Specifications in Expressions	96
3.4.3.1. Examples for Constraints of Expressions	97
3.4.3.2. Type Specifications for Names	98
3.5. Entities with Several Declaration Points	102
3.5.1. Type Declarations	103
3.5.1.1. Incomplete Type Declarations	103
3.5.1.2. Private Types	104
3.5.1.3. Discriminant Parts	105
3.5.2. Deferred Constants	106
3.5.3. Subprograms	106
3.5.3.1. Declaration and Body in One Declarative Part	107
3.5.3.2. Declaration and Body in Different Compilation Units	108
3.5.3.3. Subprogram Bodies as subunits	109
3.5.3.4. Entries and Accept Statements	110
3.5.3.5. Subprogram Formals	110
3.5.4. Packages	111
3.5.5. Tasks	112
3.5.5.1. Task Types and Task Bodies	113
3.5.5.2. Single Tasks and Task Bodies	113
3.5.6. Generic Units	114
3.6. Treatment of Instantiations	115
3.6.1. Instantiation of Subprograms	116
3.6.2. Instantiation of Packages	118
3.7. Treatment of Renaming	118
3.7.1. Renaming of Subprograms	118
3.7.2. Renaming of Packages	118
3.7.3. Renaming of Tasks	121
3.8. Implementation Dependent Attributes	121
3.8.1. Evaluation of Static Expressions	121
3.8.2. Representation of Identifiers and Numbers	122
3.8.3. Source Positions	122
3.8.4. Comments	122
3.8.5. Predefined Operators and Built-In Subprograms	123
3.9. Equality and Assignment	123
3.10. Summary of Attributes	124
3.10.1. Lexical Attributes	124
3.10.2. Semantic Attributes	125
3.10.3. Code Attributes	128
3.10.4. Unnecessary Attributes	128

4. Definition of the Diana Operations	129
4.1. The DIANA Operations	129
4.2. DIANA's Use of Other Abstract Data Types	130
4.3. Summary of Operators	130
4.4. General Method for Deriving a Package Specification for DIANA	131
4.5. Deriving a Specific ADA Package	132
4.6. The DIANA Package In ADA	133
5. External Representation of DIANA	145
6. Implementation Options	153
I. The Predefined Environment	157
I.1. Universal Types	157
I.2. The Predefined Language Environment	157
I.3. Attributes	158
I.4. Pragmas	158
II. The Abstract Parse Tree	161
III. Reconstructing the Source	165
III.1. General Principles	165
III.2. Examples	166
III.3. Normalizations of the Source	167
III.4. Comments	168
IV. Diana Summary	171
V. Diana Names	185
VI. Diana Attributes	193
VI.1. Structural Attributes	193
VI.2. Lexical Attributes	194
VI.3. Semantic Attributes	195
VI.4. Code Attributes	196
Index	199

The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official Department of Defense position, policy, or decision, unless designated by other official documentation.

LIST OF FIGURES

Figure 1-1:	Typographic Conventions used in this Document	21
Figure 1-2:	Example of IDL Notation	26
Figure 1-3:	Some Trees in ExpressionTree	27
Figure 3-1:	Example of a Necessary Tree Transformation	83
Figure 3-2:	Call of implicitly-Defined Inequality	89
Figure 3-3:	Float constraint created by DIANA	92
Figure 3-4:	DIANA Form of type/subtype Specification	94
Figure 3-5:	An Example for Derived Enumeration Types	95
Figure 3-6:	Constraints on Slices and Aggregates	99
Figure 3-7:	Constraints on Slices and Aggregates	100
Figure 3-8:	Constraints on String Literals	101
Figure 3-9:	Example of an Incomplete Type	104
Figure 3-10:	Example of a Private Type	105
Figure 3-11:	Example of a Deferred Constant	107
Figure 3-12:	Subprogram Structure	108
Figure 3-13:	Subprogram Declaration and Body in Different Compilation Units	109
Figure 3-14:	Example of a Subprogram Body as a subunit (I)	110
Figure 3-15:	Example of a Subprogram Body as a subunit (II)	111
Figure 3-16:	Example of a Package Body as a subunit	112
Figure 3-17:	Example of a Task Type and Body	113
Figure 3-18:	Example of Single Tasks	114
Figure 3-19:	Example of a Generic Body as a subunit	115
Figure 3-20:	Instantiation of a Generic Procedure	117
Figure 3-21:	Instantiation of a Generic Package	119
Figure 3-22:	Renaming a Procedure	120
Figure 3-23:	Renaming a Package	120
Figure 4-1:	Sketch of the DIANA Package	134
Figure 5-1:	External DIANA Form	146
Figure 5-2:	Example <i>ExpressionTree</i> of IDL Notation	147
Figure 5-3:	Example <i>AnotherTree</i> of IDL	149
Figure I-1:	Example of a Pragma	160