



**DESIGN OF
COMPUTER ANIMATION
LANGUAGES**

BY

Terry Thoke Trout, BSCS

A Thesis For Master of Science

**Department Of Computer Science
University of Adelaide**

January 1990

awarded 23.3.90

CONTENTS

ABSTRACT	5
DECLARATION	6
ACKNOWLEDGMENTS	7
CHAPTER 1: INTRODUCTION	8
1.1 Introduction	9
1.2 Descriptions of the Four Animation Languages	9
1.2.1 GRAMPS	10
1.2.2 ASAS	10
1.2.3 Dial	11
1.2.4 SCEFO	12
1.3 Thesis Format	12
CHAPTER 2: BACKGROUND OF ANIMATION SYSTEMS	14
2.1 Introduction	14
2.2 Objects	14
2.2.1 Object Types	15
2.2.2 Object Transformations	16
2.3 Frame to Frame Interpolation	17
2.4 Scripts	17
2.5 Displaying Individual Animation Frames	18
2.6 Input Methods	18
2.7 Control	19
2.8 Creating an Animation Sequence Quickly and Easily	20
2.9 Programming Language Versus Menu Driven System	20
CHAPTER 3: DATA STRUCTURES	21
3.1 Introduction	21
3.2 Data Structure Primitives and Constructors	22
3.3 Operators	22
3.4 Data Type Checking and Coercion	22
3.5 GRAMPS	23
3.5.1 Data Structure Primitives	23
3.5.2 Data Structure Constructors	24
3.5.3 Instantiating Objects	26
3.5.4 Operations on Objects and Interpolation	27
3.5.5 Values	28
3.6 ASAS	28
3.6.1 Data Structure Primitives	28
3.6.2 Data Structure Constructors	29
3.6.3 Predefined Data	30
3.6.4 Variables	30
3.6.5 Operators	31
3.6.5.1 Local Operators	31
3.6.5.2 Global Operators	32
3.6.5.3 Other Operators	32
3.7 DIAL	32
3.7.1 Data Structure Primitives	32
3.7.1.1 Contents of a Surface Procedure	33
3.7.1.2 Instantiating Objects	33
3.7.2 Operators	34

3.7.2.1	Operations on Objects	34
3.7.2.2	The "Change" Operator	34
3.7.2.3	Other Operators	35
3.7.3	Type Checking	36
3.8	SCEFO	36
3.8.1	Data Structure Object Template	36
3.8.2	Instantiating Objects	37
3.8.3	Operations on Objects	38
3.8.3.1	Standard Operations	38
3.8.3.2	Generic Operators, Coercion, Type Checking	39
CHAPTER 4: CONTROL STRUCTURES		40
4.1	Introduction	40
4.2	Selection	41
4.3	Iteration	41
4.4	Interpolation	42
4.5	Procedure and Macro Control Structures	42
4.6	GRAMPS	42
4.6.1	Selection and Iteration	42
4.6.2	Interpolation	43
4.6.3	Macros	43
4.7	ASAS	43
4.7.1	Iteration	44
4.7.2	Actors	45
4.7.3	Interpolation	45
4.7.4	Hierarchies	46
4.8	DIAL	46
4.8.1	Applying Operations to Objects	46
4.8.2	Execution Lines	47
4.8.3	Syntax Problems with Execution Lines	48
4.8.4	Interpolation Operations	49
4.8.5	Definite-Iteration Constructs	50
4.8.6	Procedures	51
4.9	SCEFO	51
4.9.1	Applying Operators to Objects	51
4.9.2	Interpolating SCEFO Operations	52
4.9.3	Order of Operations	53
4.9.4	Loops	54
4.9.5	Procedures	54
4.9.6	Invoking a Procedure	55
4.9.7	Hierarchies	56
CHAPTER 5: NAME STRUCTURES		57
5.1	Introduction	57
5.2	Primitive Name Structures	57
5.3	Binding Times	57
5.4	Name Structure Constructors - Scope	58
5.4.1	Static Scope	59
5.4.2	Dynamic Scope	59
5.5	Parameters	59
5.6	GRAMPS	60
5.6.1	Primitive Name Structures	60
5.6.2	Memory is Dynamically Allocated	61

5.6.3	Name Structure Constructors	61
61		
5.6.4	Passing Parameters	62
5.7	ASAS	62
5.7.1	Names Have Dynamic Scopes	62
5.7.1.1	Global Name-Binding Functions	63
5.7.1.2	Passing Parameters	63
5.7.1.3	Local Variables	64
5.7.1.4	More on Scope	64
5.7.2	Actors	65
5.7.2.1	Communication Between Actors	66
5.7.2.2	Scope Barriers Can be Bypassed	66
5.7.3	Other Functions For Binding Names	66
5.7.4	Dynamic Global Bindings are Dangerous	67
5.8	DIAL	67
5.8.1	Primitive Name Structures	67
5.8.2	Binding Times	68
5.8.3	Name Structure Constructors	68
5.8.3.1	Surface-Definition Procedure	68
5.8.3.2	“Subscript”	68
5.8.3.3	Block Structure of a Script	69
5.8.3.4	Scope Allows Transforming Unnamed Objects	70
5.8.3.5	Reference Environments are Disjoint	70
5.9	SCEFO	71
5.9.1	Primitive Name Structures	71
5.9.2	Simple Variables	71
5.9.3	Loop Variables	72
5.9.4	Names are Bound to Primitive Data When Created	72
5.9.5	Binding Time	73
5.9.6	Names Must Be Unique	73
5.9.7	Type and Value Bindings	75
5.9.8	Name Structure Constructors	75
5.9.9	Scope of Fields	76
CHAPTER 6: DESIGN OF APE - ANIMATION PROGRAMMING ENVIRONMENT		
6.1	Introduction	78
6.2	Design Method	78
6.3	Ape System	79
6.4	Commenting Scripts	79
6.5	Data Structures	80
6.5.1	Data Structure Primitives	80
6.5.1.1	“Models”	80
6.5.1.1.1	Predefined Model Figures	80
6.5.1.1.2	User-defined Models	81
6.5.1.2	Instantiation	81
6.5.1.3	Group Data Structure Constructor	81
6.5.2	Operators	82
6.5.2.1	Local Transformation Operators	83
6.5.2.2	Global Transformation Operators	85
6.5.2.3	Other Operators	85
6.5.2.4	Type Checking and Coercion	87
6.6	Control Structures	87
6.6.1	Iteration and Interpolation	87
6.6.1.1	Action Definition	87
6.6.1.2	Schedule	88
6.6.1.2.1	Automatic Schedule Filler	89

6.6.1.2.2 Grouping Schedules	91
6.6.1.2.3 Breaking Journals	92
6.6.2 Other Definite Iteration Constructs	93
6.6.3 Conditionals	94
6.6.3.1 Charting Execution of Conditionals	95
6.6.3.2 Repeating Conditionals	96
6.6.4 Indefinite Iteration	97
6.6.5 Control Structure Constructors	98
6.6.5.1 Procedures and Functions	98
98	
6.6.5.2 Actors	99
6.7 Name Structures	100
6.7.1 Primitives	100
6.7.2 Dynamic Memory Management	101
6.7.3 Name Structure Constructors	101
6.7.3.1 Scope Defined by Script Block-Structure	101
6.7.3.2 Scope Defined by Control Structures	102
6.7.3.3 Parameter Passing	102
6.7.3.4 Scope Defined by Data Structures	103
CHAPTER 7: CONCLUSION	104
7.1 The Design In Retrospect	104
7.2 Final Remarks	104
BIBLIOGRAPHY	105

ABSTRACT

This thesis analyzes four existing animation languages that are quite diverse in their designs: GRAMPS, ASAS (Actor/Scriptor Animation System), Dial (Diagrammatic Animation Language), and SCEFO (SCENE FOrmat language for describing scenes in the Brown University Animation Generation System). The analysis is performed using measurements designed from computer graphics concepts, animation requirements, and aspects and design principles of programming languages. In particular, the data structures (primitive graphics objects), control structures and name structures aspects of the four languages are thoroughly examined, as well as those design principles, slightly adapted to computer graphics and animation, found to exist in good programming languages. Other important aspects of the languages are discussed, such as how an animation sequence is represented, how any particular frame of a sequence may be viewed at any time, and how user-friendly yet controllable the languages are. Working with the findings from this analysis a new animation language is designed and partially implemented, incorporating the most positive parts of each of the four languages studied and adding other useful features.

DECLARATION

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis. I consent to the thesis being made available for photocopying and loan if applicable if accepted for the award of the degree.

Terry Thoke Trout

January 1990

ACKNOWLEDGEMENTS

I am indebted to my supervisor, Dr. Marc Berger, for his guidance and support. I am also grateful to Dr. Chris Marlin and Dr. Dave Bover for their help in the early stages of my research.

This research was supported in part by an Adelaide University Research Grant.