

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Scope . . . . .	5
1.3	Outline . . . . .	8
<b>2</b>	<b>Visual Languages and Pictorial Janus</b>	<b>9</b>
2.1	Visual Languages . . . . .	9
2.1.1	Visual Environments . . . . .	11
2.1.1.1	Static Visualization . . . . .	11
2.1.1.2	Animation . . . . .	13
2.1.2	Visual Programming Languages . . . . .	19
2.1.2.1	Prograph . . . . .	20
2.1.2.2	LabVIEW . . . . .	22
2.1.2.3	BridgeTalk . . . . .	23
2.1.2.4	SEA . . . . .	25
2.1.2.5	Forms/3 . . . . .	25
2.1.2.6	VisaVis . . . . .	27
2.1.2.7	Agentsheets . . . . .	28
2.1.2.8	KidSim . . . . .	31
2.1.2.9	ToonTalk . . . . .	32
2.1.2.10	VIPR and VEX . . . . .	34
2.2	Pictorial Janus . . . . .	40
2.2.1	Janus . . . . .	41
2.2.2	PJ Objects . . . . .	42
2.2.3	PJ Topological Syntax . . . . .	44
2.2.4	Examples . . . . .	48

2.2.5	Computation Model . . . . .	50
2.2.6	Animation . . . . .	51
2.2.7	Tools . . . . .	55
<b>3</b>	<b>Formal Specification and Evolving Algebras</b>	<b>57</b>
3.1	Formal Specification . . . . .	57
3.1.1	Theory . . . . .	59
3.1.2	Languages . . . . .	60
3.1.2.1	Predicate Logic-Based Languages . . . . .	60
3.1.2.2	Temporal Logic-Based Languages . . . . .	63
3.1.2.3	Process Algebras . . . . .	64
3.1.2.4	Combined Approaches . . . . .	67
3.1.3	Tools . . . . .	68
3.1.3.1	Model Checker . . . . .	68
3.1.3.2	Theorem Prover . . . . .	69
3.2	Evolving Algebras . . . . .	72
3.2.1	Overview . . . . .	73
3.2.2	Theory . . . . .	75
3.2.2.1	Basic EAs . . . . .	75
3.2.2.2	Extensions of Basic EAs . . . . .	77
3.2.2.3	Distributed EAs . . . . .	79
3.2.3	Methodology . . . . .	80
3.2.4	Specification Example . . . . .	81
<b>4</b>	<b>Definition of Pictorial HDL</b>	<b>84</b>
4.1	Overview . . . . .	84
4.2	Related Works . . . . .	89
4.2.1	VHDL-Based Design Environments . . . . .	89

4.2.2	Formal Semantics of Pictorial Janus . . . . .	91
4.2.3	Formal Semantics of VHDL . . . . .	92
4.3	PHDL's Pictorial Janus Aspect . . . . .	95
4.3.1	PJ Execution Cycle . . . . .	95
4.3.1.1	PJ Agents . . . . .	96
4.3.1.2	PJ System . . . . .	102
4.3.2	Timed PJ . . . . .	103
4.4	PHDL's VHDL'93 Aspect . . . . .	105
4.4.1	Introduction . . . . .	105
4.4.2	VHDL'93 Execution Cycle . . . . .	106
4.4.2.1	Basic Concepts . . . . .	107
4.4.2.2	User Defined Processes . . . . .	111
4.4.2.3	The Kernel Process . . . . .	116
4.5	PHDL Aspects Integration . . . . .	123
4.5.1	Visual-Textual Integration . . . . .	123
4.5.1.1	Interfacing . . . . .	124
4.5.1.2	VHDL Architecture Integration . . . . .	125
4.5.1.3	Guarded Processes . . . . .	127
4.5.2	Behavioral Semantics . . . . .	129
4.5.2.1	Modification of PJ Agents . . . . .	131
4.5.2.2	Definition of PHDL Agents . . . . .	131
4.5.2.3	PHDL Simulation Kernel . . . . .	133
<b>5</b>	<b>Applications</b> . . . . .	<b>139</b>
5.1	Bus Protocols . . . . .	139
5.1.1	Fully Interlocked Protocols . . . . .	140
5.1.2	Non Interlocked Protocols . . . . .	142
5.2	SDL Protocols . . . . .	147

5.2.1	SDL . . . . .	147
5.2.2	Translating SDL into Visual PHDL . . . . .	149
5.2.3	Example . . . . .	153
5.3	DLX RISC CPU . . . . .	157
5.4	Conclusion . . . . .	162
<b>6</b>	<b>Visual Interactive Design Environment</b>	<b>164</b>
6.1	Editor–Animator Interaction . . . . .	167
6.2	Animator–Interpreter Interaction . . . . .	168
6.2.1	Synchronization . . . . .	168
6.2.2	Interface . . . . .	169
6.3	Foreign Process Interface . . . . .	171
<b>7</b>	<b>Conclusions</b>	<b>173</b>
<b>A</b>	<b>PJ Topological Syntax</b>	<b>196</b>
A.1	Context-Free PJ Grammar . . . . .	196
A.2	Relational PJ Grammar . . . . .	196
A.3	Example . . . . .	198