

A Software Architecture for Modeling and Distributing Virtual Environments

Rycharde Hawkes

Doctor of Philosophy

The University of Edinburgh

1996

In memory of

Martyn “Rocket” Howard

Abstract

The simulation of a Virtual Environment (VE) is an intensive process which is severely limited if restricted to one machine. Through distribution it is possible to increase the size and accuracy of the simulation, thus permitting multiple users to interact with each other and the VE.

Existing distributed VE systems have been designed to target a specific level of distribution. This level is dictated by the geographical distance over which the systems must operate and the communications medium connecting them. The system requirements on a tightly-coupled multiprocessor system are not the same as those of a system operating over a Wide Area Network (WAN). Consequently, the solution for any given level does not scale well to larger or smaller system configurations.

VE modeling has its heritage in Computer-Aided Design (CAD) and has evolved unchecked into its present state. As the amount of information required in a VE increases, so the current modeling techniques and tools are put under added stress to cope with the extra load. Most modeling techniques are driven by the structure of the system upon which the model must execute, rather than capturing the structure of the information it should represent.

This thesis questions the motives behind VE modeling, examines the problems of distributing a VE and details the various solutions that have been employed. An analysis of the methods used leads to the selection of techniques which may be combined to provide a solution unified over all levels of distribution. The proposed solution is also integrated with and actively supports the modeling process, thus providing a powerful environment for VE designers and participants alike.

The architecture of this system is presented complete with a description of a prototype implementation that demonstrates the key aspects. The thesis concludes with an evaluation of the prototype.

Acknowledgements

There are many people that I would like to thank, but this thesis is long enough already. Therefore I would like to specifically mention just a few: my supervisor, Eric McKenzie; Martin Reddy and David Stephenson for actually reading the whole thesis; Pierre duPont for answering many questions; my friends and family for their understanding and encouragement over the years; my invisible friend Bobo the pink flying elephant, who prevented me from losing my mind; and last, but not least, Frederike van Wijck, without whose support I would surely have never finished.

This thesis was undertaken on a part-time basis whilst the author was employed on the JCI/MRC Grant G9212693: Visual Control of Steering.

Declaration

I declare that this thesis was composed by myself, and that the work contained therein is my own, except where explicitly stated otherwise in the text.

Rycharde Hawkes

Contents

1. INTRODUCTION.....	1
1.1 MOTIVATION.....	1
1.2 A MODELING TIME-LINE	2
1.2.1 Past	2
1.2.2 Present	3
1.2.3 Future.....	4
1.3 DISTRIBUTING SIMULATIONS	4
1.4 INTERACTIVITY	4
1.5 THESIS PREVIEW.....	5
1.6 SUMMARY	6
2. DESIGN ISSUES FOR DISTRIBUTED VIRTUAL ENVIRONMENT SYSTEMS.....	7
2.1 DISCRETE EVENT SIMULATION HERITAGE	8
2.1.1 <i>Optimistic versus Conservative</i>	8
2.1.2 <i>Time Warp</i>	8
2.1.3 <i>Discrete Event Simulation Summary</i>	9
2.2 ISSUES	10
2.2.1 <i>Real-time</i>	10
2.2.2 <i>Communications</i>	10
2.2.3 <i>Data Management</i>	13
2.2.4 <i>Computation Management</i>	15
2.2.5 <i>VE Modeling</i>	18
2.2.6 <i>Time Management</i>	18
2.2.7 <i>Fault Tolerance</i>	19
2.2.8 <i>Security</i>	20
2.2.9 <i>Issues Summary</i>	21
2.3 IMPLEMENTATIONS	21
2.3.1 <i>SIMulation NETworking System (SIMNET)</i>	21
2.3.2 <i>Distributed Interactive Simulation (DIS)</i>	22
2.3.3 <i>Naval Postgraduate School Networked Vehicle Simulator IV (NPSNET-IV)</i>	24
2.3.4 <i>Minimal Reality (MR) Toolkit</i>	26
2.3.5 <i>Distributed Interactive Virtual Environment (DIVE)</i>	29
2.3.6 <i>Distributed Virtual Environment System (dVS)</i>	30
2.3.7 <i>Waterloo Virtual Environment System (WAVES)</i>	33
2.3.8 <i>AVIARY</i>	35
2.4 SUMMARY	37
2.4.1 <i>Communication Organisations</i>	37
2.4.2 <i>Transport Mechanisms</i>	39
2.4.3 <i>Bandwidth Implications</i>	39
2.4.4 <i>Distribution & Scaleability</i>	40
2.4.5 <i>Time</i>	41
2.4.6 <i>Fault Tolerance</i>	42
2.4.7 <i>Security</i>	42
2.4.8 <i>Modeling</i>	43
2.4.9 <i>System Summaries</i>	43
2.4.10 <i>A Universal Architecture</i>	44

3. MODELING AND DISPLAYING VIRTUAL ENVIRONMENTS	45
3.1 A NEW MODELING PARADIGM	45
3.1.1 <i>Definition of a Model</i>	46
3.1.2 <i>An Ecological Approach</i>	46
3.1.3 <i>Tools of the Trade</i>	47
3.1.4 <i>Virtual Environment Taxonomies</i>	49
3.1.5 <i>An Abstract Model</i>	53
3.1.6 <i>Representations</i>	54
3.2 THE MODELING PROCESS	55
3.2.1 <i>Model Construction</i>	56
3.2.2 <i>The Design Process</i>	58
3.3 REAL-TIME VIRTUAL ENVIRONMENT DISPLAYS	58
3.3.1 <i>Problems with Variable-Rate Systems</i>	59
3.3.2 <i>The Variable-Rate Paradigm</i>	62
3.3.3 <i>The Fixed-Rate Paradigm</i>	63
3.3.4 <i>Conclusions</i>	64
3.4 SUMMARY	65
4. A UNIVERSAL SIMULATION SYSTEM.....	66
4.1 SYSTEM REQUIREMENTS	67
4.2 DESIGN RESTRICTIONS	68
4.2.1 <i>Finite Memory</i>	68
4.2.2 <i>Finite Computational Power</i>	68
4.2.3 <i>Finite Communications Bandwidth</i>	48
4.2.4 <i>Limited Transport Mechanisms</i>	69
4.3 DISTRIBUTED REAL-TIME SYSTEM IMPLICATIONS	69
4.3.1 <i>Computation Management</i>	70
4.3.2 <i>Memory Management</i>	72
4.3.3 <i>Locating Resources</i>	73
4.3.4 <i>Location of Backing Store</i>	73
4.3.5 <i>Fault Tolerance</i>	73
4.3.6 <i>Summary</i>	74
4.4 A UNIVERSAL MODELING LANGUAGE	74
4.4.1 <i>Language Requirements</i>	75
4.4.2 <i>Candidate Languages</i>	76
4.4.3 <i>Proposed Language</i>	80
4.4.4 <i>Summary</i>	91
4.5 SYSTEM ARCHITECTURE	91
4.5.1 <i>Universal Simulation Node</i>	91
4.5.2 <i>Universal Simulator System</i>	92
4.5.3 <i>System Organisation</i>	94
4.5.4 <i>System Operation</i>	96
4.5.5 <i>Time Management</i>	110
4.5.6 <i>Fault Tolerance</i>	111
4.5.7 <i>Access Control</i>	112
4.5.8 <i>Feature Summary</i>	113
4.6 SUMMARY	114
5. A PROTOTYPE USS	116
5.1 REAL-TIME IN THE REAL WORLD	116
5.1.1 <i>Real-Time Displays</i>	117
5.1.2 <i>Conclusions</i>	121
5.2 TARGET PLATFORMS	122
5.2.1 <i>IBM Personal Computer Compatibles</i>	122
5.2.2 <i>Cray T3D</i>	122
5.2.3 <i>Sun SPARCcenter</i>	123
5.2.4 <i>SGI RealityStation</i>	123
5.2.5 <i>Implementation Language</i>	123
5.3 NETWORKING	124

5.3.1 IPC Mechanisms	125
5.3.2 Addresses	125
5.3.3 Messages	126
5.3.4 Hardware Differences	127
5.3.5 Layer Implementation	129
5.3.6 Networking Summary	133
5.4 CONFIGURATION CONTROL	133
5.4.1 Universal Configuration Language (UCL)	133
5.4.2 System Configuration	134
5.5 A UML INTERPRETER	135
5.5.1 Overall Structure	136
5.5.2 Interpreting the Data Definition	136
5.5.3 Instancing	137
5.5.4 Component Dependencies	140
5.5.5 Interpreting Instruction Code	141
5.5.6 Interpreter Embedding	142
5.5.7 Persistence	142
5.6 UNIVERSE MANAGER	142
5.6.1 Node Initialisation	143
5.6.2 System Initialisation	143
5.6.3 Managing Processes	145
5.6.4 Processing Service Requests	147
5.6.5 Entity Migration	147
5.6.6 System Interaction	148
5.6.7 System Termination	148
5.7 RESOURCE MANAGER	148
5.7.1 Resource Consumption	148
5.7.2 Initialisation	149
5.7.3 Services	149
5.8 ENTITY LIBRARY	150
5.8.1 Initialisation	151
5.8.2 Service Requests	151
5.9 MANAGER LIBRARY	152
5.9.1 Initialisation	152
5.9.2 Simulation Loop	153
5.10 VISUAL MANAGER	153
5.10.1 Initialisation	153
5.10.2 Simulation Loop	154
5.10.3 Entity Enhancement	154
5.10.4 VIS Summary	154
5.11 CONSOLE	154
5.12 FURTHER IMPROVEMENTS	154
5.12.1 Configuration	154
5.12.2 Multi-part Messages	155
5.12.3 State Encoding	155
5.12.4 Persistence	155
5.12.5 Message Elimination	155
5.12.6 Entity Synchronisation	156
5.12.7 Function Access	156
5.13 SUMMARY	156

6. PROTOTYPE EVALUATION.....	158
6.1 SYSTEM ANALYSIS	158
6.2 TESTING METHODOLOGY.....	159
6.2.1 CPU Performance.....	159
6.2.2 Computation and Communication	162
6.2.3 Memory	162
6.2.4 Instrumentation	164
6.3 UML	164
6.3.1 Code Size.....	164
6.3.2 Primitive Types	165
6.3.3 Component Sizes	165
6.3.4 Interpretation	169
6.3.5 State Management.....	171
6.3.6 Summary of UML Analysis.....	174
6.4 PML	175
6.4.1 Transmission	175
6.4.2 Reception	179
6.4.3 Throughput.....	182
6.4.4 TCP/IP Performance.....	184
6.4.5 PML Summary	185
6.5 SIMULATION EXECUTION	186
6.5.1 Single Node	188
6.5.2 Two Nodes.....	191
6.5.3 Three Nodes	195
6.5.4 Entity Migration.....	198
6.5.5 Process Activity.....	201
6.5.6 Simulation Execution Summary	203
6.6 IMPROVING PERFORMANCE	204
6.6.1 Message Elimination.....	204
6.6.2 Shared-Memory IPC	205
6.6.3 Multicast	206
6.6.4 Accounting for Latency.....	207
6.6.5 Increased Bandwidth	208
6.7 SUMMARY	209
6.7.1 Living with TCP/IP	209
6.7.2 Resource Management.....	210
6.7.3 Scalability.....	210
6.7.4 Distribution at a Price	212
6.7.5 Conclusions.....	213
7. CONCLUSION	214
7.1 THESIS REVIEW	214
7.2 USS CLASSIFICATION	216
7.2.1 Communications	216
7.2.2 Data and Computation Management.....	217
7.2.3 VE Modeling	218
7.2.4 Time Management.....	218
7.2.5 Fault Tolerance.....	219
7.2.6 Security	219
7.3 IMPORTANT FEATURES	219
7.3.1 Real-Time.....	219
7.3.2 Scalability.....	219
7.3.3 Bandwidth Reduction	220
7.3.4 Modeling	220
7.3.5 Flexibility	221
7.4 AREAS FOR INVESTIGATION	222
7.4.1 Reliable Multicast.....	222
7.4.2 Guaranteed Bandwidth	222
7.4.3 Time Synchronisation.....	222

7.4.4 Real-Time Operating Systems	223
7.4.5 Shadowing the User	223
7.5 OUTLOOK	223
7.5.1 Internet.....	223
7.5.2 Virtual Reality Modeling Language.....	224
7.5.3 Java.....	224
7.5.4 Consequences.....	224
7.6 SUMMARY	225
 APPENDIX A	226
 APPENDIX B	231
 APPENDIX C	237
 APPENDIX D	238
 APPENDIX E	239
 GLOSSARY	240
 BIBLIOGRAPHY	244

List of Figures

FIGURE 2.1	EXAMPLE INTERACTION MATRIX.	17
FIGURE 2.2	A CLOCK SYNCHRONISATION FORMULA.....	28
FIGURE 2.3	DIS PERFORMANCE WITH DIFFERENT DEAD-RECKONING ACCURACIES.	23
FIGURE 2.4	MR TOOLKIT COMPONENT STRUCTURE.....	27
FIGURE 2.5	dVS SYSTEM ARCHITECTURE.	31
FIGURE 2.6	BASIC WAVES ARCHITECTURE.....	33
FIGURE 2.7	AVIARY COMPONENT SCHEMATIC.	35
FIGURE 3.1	CONFECTION OF ARTIFICIAL AND REAL ENVIRONMENTS.	49
FIGURE 3.2	TYPE OF TIME AND SPACE.....	49
FIGURE 3.3	FUNDAMENTAL ELEMENTS OF A VIRTUAL ENVIRONMENT.	52
FIGURE 3.4	A CLASSIFICATION SCHEME FOR VIRTUAL ENVIRONMENTS.....	52
FIGURE 3.5	UNIVERSE HIERARCHY TREE SHOWING POSSIBLE ENTITY MIGRATION PATHS.	57
FIGURE 3.6	THE EFFECT OF UPDATE RATE ON TIME-TO-CONTACT (TTC).	61
FIGURE 4.1	A TAXONOMY OF REAL-TIME SCHEDULING ALGORITHMS.	70
FIGURE 4.2	BACKUS-NAUR FORM DESCRIPTION SHOWING RELATIONSHIPS BETWEEN UML COMPONENTS.	81
FIGURE 4.3	EXAMPLE TOP-LEVEL UML DESCRIPTION.	82
FIGURE 4.4	A POSSIBLE DEFINITION FOR THE VISUAL ELEMENT.....	83
FIGURE 4.5	DEFINING A UNIVERSE BY INHERITANCE.....	84
FIGURE 4.6	INHERITING FROM AN ELEMENT.....	85
FIGURE 4.7	EXPLICIT/IMPLICIT INVOCATION OF A CONVERTER.....	85
FIGURE 4.8	DEFINITION OF AN ENTITY.	86
FIGURE 4.9	METHODS FOR ACCESSING MEMBER PROPERTIES IN ELEMENTS.....	87
FIGURE 4.10	USER AND SYSTEM FUNCTION CALL EXECUTION.	88
FIGURE 4.11	ATTEMPTING TO CONVERT AN ELEMENT WITHOUT A CONVERTER.	89
FIGURE 4.12	CALCULATING A TIME DELTA USING STATE INDEXING.	89
FIGURE 4.13	IMPORTING A MODULE.	90
FIGURE 4.14	EXAMPLE STRUCTURE OF A UNIVERSAL SIMULATION SYSTEM.	92
FIGURE 4.15	EXAMPLE ORGANISATION OF A USS COMPLETE WITH POPULATED USNs.....	93
FIGURE 4.16	HIERARCHICAL STRUCTURING OF USSs.	95
FIGURE 4.17	ORDER OF EVENTS FOR A SIMULATION UPDATE.....	97
FIGURE 4.18	POSSIBLE COMMUNICATION PATH TAKEN BY A MESSAGE SENT FROM AN ENTITY TO ALL MANAGERS.	98
FIGURE 4.19	PROCEDURE FOR REGISTERING INTEREST IN A UML COMPONENT.....	99
FIGURE 4.20	SEQUENCE OF EVENTS DURING ENTITY CONSTRUCTION.	101
FIGURE 5.1	SIMULATION CYCLE SCHEDULING.....	118
FIGURE 5.2	PSEUDO-CODE FOR THE FIXED FRAME RATE, WORST-CASE SIMULATION CYCLE.	119
FIGURE 5.3	IMPROVED SIMULATION CYCLE SCHEDULING.	120
FIGURE 5.4	EXAMPLE NETWORK CONFIGURATIONS OF THREE USSs.....	124
FIGURE 5.5	POSITION OF THE PROCESS MANAGEMENT LAYER WITHIN THE SYSTEM SOFTWARE.....	125
FIGURE 5.6	SEND-RECEIVE-REPLY PROCEDURE FOR SENDING MESSAGES UNDER QNX.	130
FIGURE 5.7	MESSAGE TRANSMISSION SEQUENCE USING TCP/IP.....	131
FIGURE 5.8	STRUCTURE OF A LOGICAL PROCESS CONSISTING OF TWO PHYSICAL PROCESSES.	132
FIGURE 5.9	THE BASIC ELEMENTS OF UCL.	134
FIGURE 5.10	EXAMPLE USS CONFIGURATION FILE.	135
FIGURE 5.11	CORE UML C++ CLASS HIERARCHY.	136
FIGURE 5.12	UML CODE FRAGMENT AND THE INTERNAL DATA STRUCTURE USED TO REPRESENT IT..	138
FIGURE 5.13	INSERTION OF VECTOR PROPERTY INTO ELEMENT.	139

FIGURE 5.14 UMLCOMPONENT STRUCTURE WITH DEPENDENCY.	140
FIGURE 5.15 STRUCTURE OF THE INFORMATION HELD FOR EACH PROCESS.	146
FIGURE 5.16 A STATE UPDATE USES A DIFFERENT MONITOR ID WHEN SENT TO EACH DEPENDENT.	146
FIGURE 5.17 RESOURCE CONSUMPTION REPRESENTATION.	149
FIGURE 5.18 EXAMPLE NODE RESOURCE CONFIGURATION USED BY A RM.	150
FIGURE 5.19 STRUCTURES USED TO KEEP TRACK OF ENTITIES AND THEIR COMPONENT DEPENDENCIES.	152
FIGURE 5.20 EXAMPLE USE OF THE PRIVATE KEYWORD TO REDUCE FUNCTION ACCESS THROUGH SCOPE.	156
FIGURE 6.1 BASIC RELATIONSHIPS BETWEEN UML COMPONENTS AND THEIR MEMORY USAGE.....	166
FIGURE 6.2 BASIC INTERPRETER OVERHEADS FOR THREE PRIMITIVE TYPES.	170
FIGURE 6.3 COST OF ADDING AND REMOVING A DEPENDENCY ON A UML COMPONENT.	171
FIGURE 6.4 FUNDAMENTAL STATE OPERATIONS ON AN INTEGER/REAL/BOOLEAN AND THEIR COST ON EACH PLATFORM.	171
FIGURE 6.5 STATE OPERATION COSTS BASED UPON STATE SIZE (PENTIUM).....	172
FIGURE 6.6 STATE OPERATION OVERHEADS FOR AN ELEMENT WITH ZERO AND ONE PROPERTIES (PENTIUM).	172
FIGURE 6.7 STATE OPERATIONS ON ELEMENTS WITH ONE TO TEN PROPERTIES (PENTIUM).	173
FIGURE 6.8 COSTS OF STATE SIZING/PACKING/UNPACKING A BOOLEAN/INTEGER/REAL ON ALL THE TEST PLATFORMS.	173
FIGURE 6.9 STATE ENCODING OPERATION OVERHEADS FOR A STRING OF 40 CHARACTERS.	174
FIGURE 6.10 RELATIVE OVERHEADS IMPOSED BY A PIPE ON EACH TEST PLATFORM.	175
FIGURE 6.11 DECONSTRUCTED PML OVERHEADS FOR SENDING A MESSAGE UNDER QNX (SERVER).	176
FIGURE 6.12 PML MESSAGE TRANSMISSION LATENCY BETWEEN LOCAL PROCESSES ON THE QNX PLATFORMS.	176
FIGURE 6.13 PML MESSAGE TRANSMISSION LATENCY BETWEEN REMOTE PROCESSES.	177
FIGURE 6.14 PML MESSAGE TRANSMISSION TIMES FOR TCP/IP.....	178
FIGURE 6.15 EFFECT OF PROTOCOL ON LOCAL PML TRANSMISSION TIME (GATEWAY).	179
FIGURE 6.16 BREAKDOWN OF A PML MESSAGE RECEIVE UNDER QNX (GATEWAY).....	180
FIGURE 6.17 COMPARISON BETWEEN RECEIVING MESSAGES FROM LOCAL AND REMOTE PROCESSES (PENTIUM).	180
FIGURE 6.18 PML MESSAGE RECEPTION USING TCP/IP.	181
FIGURE 6.19 MAXIMUM LOCAL THROUGHPUT WITHIN EACH NODE USING QNX IPC.	182
FIGURE 6.20 MAXIMUM QNX NETWORK THROUGHPUT BETWEEN NODE PAIRS.	183
FIGURE 6.21 MAXIMUM TCP/IP MESSAGE PASSING THROUGHPUT FOR EACH PLATFORM.	183
FIGURE 6.22 PML MESSAGE RECEPTION TEST FOR TCP/IP ON REALITY WITH TIME-WAIT BUILD-UP.....	184
FIGURE 6.23 ACTIVITY BREAKDOWN OF A UM WHEN THERE IS ONE MONITORED COMPONENT.	187
FIGURE 6.24 ACTIVITY BREAKDOWN OF A UM WHEN THERE ARE TWO MONITORED COMPONENTS. ...	189
FIGURE 6.25 EFFECTS OF VARIOUS FACTORS ON SIMULATION RATE.....	190
FIGURE 6.26 PERCENTAGE IDLE TIME IN THE UM FOR EACH TEST CASE (PENTIUM).	191
FIGURE 6.27 COMPARISON OF PENTIUM AND GATEWAY BASELINE PERFORMANCE.	191
FIGURE 6.28 ACTIVITY BREAKDOWN OF UMS IN A MASTER-SLAVE CONFIGURATION WITH NO MANAGERS.....	192
FIGURE 6.29 ACTIVITY BREAKDOWN OF UMS IN A MASTER-SLAVE CONFIGURATION WITH 1 MANAGER ON THE MASTER NODE.	193
FIGURE 6.30 ACTIVITY BREAKDOWN OF UMS IN A MASTER-SLAVE CONFIGURATION WITH 1 MANAGER ON THE SLAVE NODE.....	194
FIGURE 6.31 ACTIVITY BREAKDOWN OF UMS IN A MASTER AND 2 SLAVES CONFIGURATION WITH NO MANAGERS.....	196
FIGURE 6.32 ACTIVITY BREAKDOWN OF UMS IN A MASTER AND 2 SLAVES CONFIGURATION WITH A MANAGER ON SLAVE NODE SERVER.	197
FIGURE 6.33 SINGLE ENTITY MIGRATION WITHIN A TWO NODE SYSTEM (PENTIUM/SERVER).....	198
FIGURE 6.34 MULTIPLE ENTITY MIGRATION WITHIN A TWO NODE SYSTEM (PENTIUM/SERVER).....	199
FIGURE 6.35 MULTIPLE ENTITY MIGRATION WITHIN A THREE NODE CONFIGURATION.	200
FIGURE 6.36 CONSTRUCT/UPDATE/DESTRUCT TIMES FOR THE ENTITY USED DURING BENCHMARKING.....	201
FIGURE 6.37 AVERAGE TASK BREAKDOWN FOR A SINGLE ENTITY.....	202

FIGURE 6.38 ACTIVITY BREAKDOWN FOR THE BENCHMARK MANAGER.	202
FIGURE 6.39 POTENTIAL PERFORMANCE INCREASES WITH A MULTICAST UPDATE (PENTIUM).	204
FIGURE 6.40 INCREASES IN SIMULATION RATE WHEN ELIMINATING BOTH UPDATE CONTROL MESSAGES.	205
FIGURE 6.41 COMPARISON OF ESTIMATED REALITY PERFORMANCE WITH SHARED MEMORY IPC AND PENTIUM USING QNX IPC.	205
FIGURE 6.42 ACCOUNTING FOR MESSAGE LATENCY REDUCES SIMULATION CYCLE DURATION.	207
FIGURE 6.43 PENTIUM TO SERVER TRANSMISSION TIMES WHEN USING THE CURRENT ETHERNET LINK AND A PREDICTED FAST ETHERNET LINK.	208
FIGURE A.1 YACC DESCRIPTION OF UML GRAMMAR.	230
FIGURE B.1 UML DEFINITION USED IN THE PROTOTYPE EVALUATION.	231
FIGURE B.2 BENCHMARK ENTITY SOURCE CODE.	234
FIGURE B.3 BENCHMARK MANAGER SOURCE CODE.	236

List of Tables

TABLE 2.1 KLEINROCK DISTRIBUTION CLASSIFICATION SCHEME.....	13
TABLE 2.2 DEGREES OF FAULT TOLERANCE.....	20
TABLE 2.3 DISTRIBUTED VE SYSTEM FEATURE CLASSIFICATION SUMMARY.....	38
TABLE 3.1 COMMON SENSES AND THEIR SENSITIVITY/RESOLUTION.	48
TABLE 4.1 NETWORKING MEDIUM PROPERTIES.	69
TABLE 4.2 INTERPRETER RESOURCE EVALUATION.....	79
TABLE 5.1 MESSAGE ADDRESS STRUCTURE.....	126
TABLE 5.2 MESSAGE HEADER STRUCTURE.	126
TABLE 5.3 DESCRIPTION OF THE NINE PHYSICAL MESSAGE STRUCTURES.	127
TABLE 5.4 SUMMARY OF MESSAGE TYPES AND THEIR USE.	128
TABLE 5.5 SAMPLE ENTITY DISTRIBUTION OVER THREE NODES.	144
TABLE 6.1 RESOURCE RATINGS FOR EACH TEST PLATFORM.	160
TABLE 6.2 MINIMUM MEMORY USAGE OF USS COMPONENTS.	163
TABLE 6.3 EFFECTS OF TECHNIQUES TO REDUCE CODE SIZE.	165
TABLE 6.4 MEMORY CONSUMPTION OF THE FOUR PRIMITIVE UML TYPES.	165
TABLE 6.5 APPROXIMATE MEMORY USAGE FOR UML COMPONENTS.	167
TABLE 6.6 EXAMPLE OF HOW MUCH MEMORY IS ALLOCATED TO REPRESENT A UML DEFINITION AND HOLD ITS INSTANCE DATA UNDER QNX.	169
TABLE 6.7 FUNDAMENTAL INTERPRETER OPERATIONS TIMINGS FOR EACH TEST PLATFORM.	169
TABLE 6.8 OPERATION OVERHEADS PER INTEGER ARRAY ELEMENT.	172
TABLE 6.9 TIME PENALTIES INCURRED WHEN RM MONITORS CPU USAGE.	203
TABLE 7.1 COMPARISON OF DISTRIBUTED VE FEATURE CLASSIFICATIONS INCLUDING USS.	217