

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Reusability for Intelligent Realtime Interactive Systems	4
1.2.1	Software Reusability and Software Reuse	4
1.2.2	Intelligent Realtime Interactive Systems	6
1.3	Problem Statement	8
1.4	Objectives	9
1.5	Structure	10
2	Related Work	13
2.1	Overview	13
2.2	Software Reusability	14
2.2.1	Definitions	14
2.2.2	Benefits and Inhibitors of Software Reuse	17
2.2.3	Facets of Software Reuse	20
2.2.4	Reuse Techniques	22
2.2.5	Measuring Reusability	26
2.2.6	Intermediate Conclusion	27
2.3	Intelligent Realtime Interactive Systems	28
2.3.1	Aspects of VE Development	30
2.3.2	VR Frameworks	31
2.3.3	Game Engines	36
2.3.4	IVE frameworks	39
2.4	Knowledge Representation	44
2.4.1	Forms of Knowledge Representation	45
2.4.2	Ontologies	47
2.4.3	Action Representation	49
2.4.4	Knowledge Representation Layers	49
2.5	Summary	51
3	A Knowledge Representation Model for IRISs	55
3.1	Semantics	55
3.2	Semantics of IRIS Frameworks	56
3.2.1	Benefits of Integrated Semantics	56
3.2.2	Aspects and Requirements	57
3.2.3	Level of Integration	61
3.2.4	Representable Elements	62
3.2.5	Summary	69
3.2.6	Adopting the Web Ontology Language	69

3.3	A Reusable Knowledge Representation Model	73
3.3.1	Relation Descriptions	73
3.3.2	Value Descriptions	74
3.3.3	Aspects and Entity Descriptions	77
3.3.4	Actions, Preconditions, and Effects	79
3.3.5	Events	81
3.3.6	Simulation Components	81
3.3.7	Choosing an Adequate Representation	82
3.3.8	Summary	85
3.4	Integration into an IRIS Framework	86
3.4.1	A Layered Approach	86
3.4.2	Integration into Program Code	87
3.4.3	Cost-Benefit Analysis	92
3.5	Summary	98
4	A Semantics-based Component Model for Reusable IRISs	99
4.1	Aspects of Reusability	99
4.2	A Semantics-based Approach	104
4.2.1	Ontological Grounding	105
4.2.2	Semantic Values	105
4.2.3	Semantic Traits	106
4.2.4	Relations	107
4.2.5	Methods	109
4.2.6	Example of Application	110
4.2.7	Annotations	112
4.2.8	Benefits and Drawbacks	113
4.3	A Uniform Access Model	114
4.3.1	State Representation & State Transitions	115
4.3.2	Uniform Access to the Simulation State	115
4.3.3	Simulation Modules	117
4.3.4	Benefits and Drawbacks	117
4.4	Concurrency: Adoption of the Actor Model	119
4.4.1	Issues to be faced	120
4.4.2	Integration	122
4.5	A Component Model for IRIS Frameworks	126
4.5.1	Semantic Values, Events, and Automatic Type Conversion	127
4.5.2	Entity and Component Descriptions	129
4.5.3	Components	133
4.5.4	Application Composition	138
4.6	Summary	139

5	Simulator X	143
5.1	Simulator X - A VR Research Platform	143
5.2	Architecture	144
5.2.1	Actors	144
5.2.2	Scala	145
5.2.3	Semantic Values	148
5.2.4	Events	149
5.2.5	State Variables	150
5.2.6	Entity Model	152
5.2.7	Relations	157
5.2.8	Components	158
5.2.9	Ontology	162
5.3	Features	165
5.3.1	Distributed Computing	166
5.3.2	Automatic Type Conversion	167
5.3.3	State History	168
5.4	Project Structure	170
5.4.1	Version Control and Project Structure	170
5.4.2	Dependency Management	171
5.5	Implemented Components	172
5.5.1	Adding new Sensors	172
5.5.2	Reasoning	173
5.5.3	Planning	175
5.5.4	Example Component: Unity	179
5.6	Summary	182
6	Validation	185
6.1	Evaluating Reusability	185
6.2	Case Study: Replacing a Rendering Component	185
6.2.1	Creating a new Component	186
6.2.2	Exchanging Components	188
6.2.3	Reuse in Simulator X	191
6.3	Developed Applications	193
6.3.1	Research Demonstrations	193
6.3.2	Exhibitions	196
6.3.3	Teaching-related Applications	197
6.4	Opinions: Questionnaires	197
6.4.1	Results: NASA TLX	199
6.4.2	Results: QUESI	202
6.4.3	User Comments	204
6.4.4	Discussion	205
6.5	Summary	206

7 Conclusion	209
7.1 Summary	209
7.2 Future Work	212
Bibliography	215
A OWL Definitions	231
B Interfaces	237
C Questionnaire Results	239