

Contents

Abstract	3
Zusammenfassung	4
Acknowledgements	5
Contents	7
List of Figures	11
1. Introduction	13
1.1 Continuous Software Engineering	13
1.1.1 Software Component Integration	14
1.1.2 Reengineering	16
1.1.3 Software Design	17
1.2 Problem Statement	18
1.2.1 Practical Argument	18
1.2.2 Theoretical Argument	20
1.3 Motivating Example	21
1.4 Research Overview	22
1.4.1 Objective	22
1.4.2 Deliverables	22
1.4.3 Related Publications	23
1.5 Thesis Organization	23
2. Software Architecture	27
2.1 Definition and Objectives	27
2.2 Techniques Toward an Architecture	29
2.2.1 Reference Models	29
2.2.2 Architectural Styles	34

2.2.3	Design Patterns	36
2.3	Languages to Express an Architecture	37
2.3.1	Interface Definition Languages	37
2.3.2	Architecture Description Languages	41
Components		42
Connectors		43
2.3.3	Object-Oriented Modeling Languages	46
2.4	Summary of Related Work	48
3.	Component Infrastructure Technology	51
3.1	Requirements	52
3.1.1	Development Support	52
3.1.2	Execution Support	53
3.1.3	Deployment Support	54
3.2	Sample Distributed Object Middleware	54
3.2.1	CORBA	55
3.2.2	Java/RMI	57
3.2.3	COM	58
3.2.4	Assessment and Future	59
Technology Integration		59
Middleware Services		60
Object Transaction Monitors		61
3.3	A Case Study of Infrastructure Service Design	61
3.3.1	Problem Domain	62
3.3.2	Infrastructure Service Approach	63
3.3.3	Observations	67
3.4	The Way Ahead	68
4.	A Software Component and Connector Model	69
4.1	Architectural Abstraction Concepts	69
4.1.1	Components	70
Interface Specification		72
Representation-Map		72
Representation		73
Component Characteristics		74
4.1.2	Connectors	76
Roles		76
Role Interfaces		77
Interactions		78
4.1.3	Abstract Architectures	79

4.1.4	Concrete Architectures	81
4.2	Modeling Example	82
4.2.1	Components	84
	Interface Specifications	84
	Representations and Rep-Maps	87
4.2.2	Continuous Engineering	89
4.2.3	Connectors	90
4.2.4	Abstract Architectures	91
4.2.5	Concrete Architectures	93
	Object Binding and Resolution	94
	Event Notification	95
	Transaction Processing	98
4.3	Summary of Key Principles	102
	Separation of Modeling Concerns	104
	Pattern-Oriented	105
	Component-Oriented	105
	Anticipation of Change	106
	Technology Relationship	106
5.	Service Connectors	107
5.1	Modeling CORBA-based Systems	107
5.2	Modeling Formalism	109
5.3	Dynamic Object Binding	112
5.3.1	The CORBA Naming Service	112
5.3.2	Connector "Object Binding and Resolution"	113
	Roles	113
	Role Interfaces	113
	Interactions	115
5.4	Event-Based Communication	117
5.4.1	The CORBA Events Service	117
5.4.2	Connector "Event Notification"	118
	Roles	118
	Role Interfaces	118
	Interactions	121
5.5	Distributed Transaction Processing	125
5.5.1	The CORBA Object Transaction Service	125
5.5.2	Connector "Transaction Processing"	127
	Roles	127
	Role Interfaces	127

Interactions	130
5.5.3 Modeling Example	135
Abstract Architecture	135
Concrete Architecture	136
5.6 Object Caching	139
5.6.1 Load Balancing, Replication, and Object Caching	139
5.6.2 Connector “Object Caching”	141
Roles	141
Role Interfaces	141
Interactions	144
5.6.3 Modeling Example	149
Abstract Architecture	150
Concrete Architecture	152
Performance Test	155
5.7 Object Groups	157
5.7.1 Object Groups and Server Pools	157
5.7.2 Connector “Server Group”	158
Roles	158
Role Interfaces	158
Interactions	161
5.8 Summary and Discussion	165
6. Design Method	167
6.1 Design Process	167
6.1.1 Forward Design	168
6.1.2 Reverse Design	169
6.1.3 Continuous Design	171
6.2 Design Considerations	172
6.2.1 Distributed Interface Design	172
6.2.2 Wrapper Design	173
6.2.3 Distributed System Configuration	174
7. Conclusion	177
7.1 Summary of Results	177
7.2 Future Work	179