

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
SUMMARY	xvii
I INTRODUCTION	1
1.1 Overview of Software Tools	5
1.1.1 Middleware with Copy-Restore Semantics	5
1.1.2 Program Generation for Distribution	6
1.1.3 Automatic Partitioning	7
1.2 Thesis Statement	8
1.3 Contributions	9
1.4 Overview of Distribution Concerns	10
1.5 Overview of Dissertation	11
II NRMI	12
2.1 Introduction	12
2.2 Background and Motivation	15
2.3 Supporting Copy-Restore	20
2.4 Discussion	23
2.4.1 Copy-Restore vs. Call-by-Reference	23
2.4.2 DCE RPC	24
2.5 NRMI Implementations	26
2.5.1 A Drop-in Replacement of Java RMI	27
2.5.1.1 Programming Interface	27
2.5.1.2 Implementation Insights	29
2.5.2 NRMI in the J2EE Application Server Environment	31
2.5.3 Introducing NRMI through Bytecode Engineering	34
2.5.3.1 User View: NRMIzer	35
2.5.3.2 Implementation Specifics: Backend Engine	36
2.6 Conclusion	37

III	GOTECH	39
3.1	Introduction	39
3.2	The Elements of Our Approach	42
3.2.1	NRMI	42
3.2.2	AspectJ	42
3.2.3	XDoclet	43
3.3	The Framework	44
3.3.1	Overview	44
3.3.2	Framework Specifics	45
3.3.2.1	Middleware	46
3.3.2.2	GOTECH Annotations	46
3.3.2.3	GOTECH XDoclet Templates	47
3.3.3	Discussion of Design	50
3.4	Advantages and Limitations	51
3.4.1	Advantages of our approach	51
3.4.2	Limitations	54
3.4.2.1	Entity Bean support	54
3.4.2.2	Conditions for applying rewrite	55
3.4.2.3	Making types serializable	55
3.4.2.4	Exceptions, construction, field access	57
3.5	Conclusions	58
IV	J-ORCHESTRA	59
4.1	Introduction	60
4.2	User View of J-Orchestra	63
4.3	The General Problem and Approach	66
4.4	Classification Heuristic	70
4.5	Rewriting Engine	77
4.5.1	General Approach	77
4.5.2	Call-Site Wrapping for Anchored Modifiable Code	82
4.5.3	Placement Policy Based On Creation Site	87
4.5.4	Object Mobility	89
4.6	Dealing with Concurrency and Synchronization	93
4.6.1	Overview and Existing Approaches	94
4.6.2	Distributed Synchronization Complications	95
4.6.3	Solution: Distribution-Aware Synchronization	98
4.6.4	Benefits of the Approach	104
4.6.4.1	Portability	104
4.6.4.2	The Cost of Universal Extra Arguments	106

4.6.4.3	Maintaining Thread Equivalence Classes Is Cheap	110
4.6.5	Discussion	111
4.7	Appletizing: Partitioning for Specialized Domains	113
4.7.1	Static Analysis for Appletizing	116
4.7.2	Profiling for Appletizing	117
4.7.3	Rewriting Bytecode for Appletizing	119
4.7.4	Runtime Support for Appletizing	124
4.8	Run-Time Performance	126
4.8.1	Indirection Overheads and Optimization	126
4.8.1.1	Indirection Overheads	126
4.8.1.2	Local-Only Optimization	128
4.9	Java Language Features And Limitations	129
4.9.1	Unsafety	130
4.9.2	Conservative classification	131
4.9.3	Reflection and dynamic loading	131
4.9.4	Inherited limitations	132
4.10	Conclusions	133
V	APPLICABILITY AND CASE STUDIES	135
5.1	Applicability of the Translucent Approach	135
5.2	Applicability of NRMI: Usability Call-by-Copy-Restore vs. Call-by-Copy	137
5.3	Applicability of GOTECH: What are the Distribution Concerns and Can They Be Separated?	140
5.3.1	Semantics	140
5.3.2	Performance	141
5.3.3	Conventions	141
5.4	Applicability of J-Orchestra: Conditions for Successful Partitioning	143
5.5	NRMI Case Studies	146
5.5.1	NRMI Low-Level Optimizations	147
5.5.2	Description of Experiments	148
5.5.3	Experimental Results	152
5.6	The GOTECH Case Study	156
5.7	J-Orchestra Case Studies	161
5.7.1	Appletizing Case Studies	161
5.7.1.1	JBits	162
5.7.1.2	JNotepad	165
5.7.1.3	Jarminator	167
5.7.1.4	Discussion	168
5.7.2	Kimura Case Study	169
5.7.3	Other J-Orchestra Case Studies	177

VI GENERALIZING THE J-ORCHESTRA INDIRECTION MACHINERY	.179
6.1 Introduction	179
6.2 User-Level Indirection Techniques	180
6.3 Transparency Limitations	182
6.3.1 Beyond Java Conventions: Native Code in .NET	187
6.4 Weak Assumptions of J-Orchestra Classification	188
6.4.1 Type-Based Analysis + Weak Assumptions	188
6.4.2 More Sophisticated Type-Based Analysis	193
6.5 Validating The Assumptions and Analysis	195
6.5.1 Impact on Real Applications	197
6.5.2 Accuracy of Type Information	199
6.5.3 Testing Correctness	201
6.6 Conclusions	203
VII RELATED WORK	.205
7.1 Directly Related Work	205
7.1.1 NRMI	205
7.1.1.1 Performance Improvement Work	205
7.1.1.2 Usability Improvement Work	206
7.1.2 GOTECH	209
7.1.3 J-Orchestra	211
7.2 Related Research Areas	214
7.3 Beneficiaries of This Research	216
VIII FUTURE WORK AND CONCLUSIONS	.219
8.1 NRMI Future Work	219
8.2 GOTECH Future Work	223
8.3 J-Orchestra Future Work	224
8.4 Merits of the Dissertation	230
8.5 Conclusions	231
REFERENCES	.233
VITA	.244