

Class and Object Modularity
Description and Measurement

Christine McClean

Doctor of Philosophy in Engineering

2006

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgements

Thank you to my supervisors John Leaney and Keiko Yasukawa for their support, valuable suggestions and comments regarding this research. In particular, I would like to express my very special thanks and gratitude to Keiko Yasukawa for her help and support throughout the process of revising this thesis. “I couldn’t have done it without you!”

Many thanks also to Keiko Yasukawa, Noel McClean and John Leaney for their assistance in proof reading this thesis document and suggesting improvements.

Thank you to the anonymous assessors of this thesis for their helpful comments and insights that have contributed to many improvements to this research work.

Thank you to Peter Petocz for suggesting the calculation of an aggregation of modularity as an appropriate data analysis technique.

Thank you to Adrian Richards for offering insights into the process of scientific measurement.

Thank you to Scientific Toolworks Inc. for providing a copy of the Understand For C++ code analysis application.

Finally, last but not least, thank you to my husband Noel and our girls Catie and Merryn for their support, patience and understanding throughout this research.

Table of Contents

1. INTRODUCTION	1
1.1. APPROACH TAKEN TO SYSTEMATIC MEASURE DEVELOPMENT PROCESS DEFINITION	2
1.2. KEY CONCEPTS	3
1.3. MEASURE DEVELOPMENT PROCESS	6
1.4. MEASURE VALIDITY	11
1.4.1 Content validation	14
1.4.2 Construct validation.....	16
1.5. MEASURE RELIABILITY	18
1.6. LEVEL OF MEASUREMENT	18
1.7. COMMENTS	19
1.8. STRUCTURE OF THIS THESIS.....	20
2. LITERATURE REVIEW	22
2.1. PROCESSES OF SOFTWARE MEASURE DEVELOPMENT	23
2.2. PROCESSES CURRENTLY USED TO DEVELOP DESCRIPTIVE SOFTWARE MEASURES	27
2.3. THEORETICAL BASIS OF DESCRIPTIVE SOFTWARE MEASURE DEVELOPMENT	33
2.4. CONCLUSION.....	35
3. CONCEPTUAL DEFINITION	37
3.1. STAGE 1 OF MEASURE DEVELOPMENT PROCESS - CONCEPTUAL DEFINITION	37
3.1.1 Prerequisites to the conceptual definition stage	40
3.1.2 Performance of the conceptual definition stage.....	41
3.1.3 Products of the conceptual definition stage	42
3.1.4 Practical Considerations.....	42
3.2. CONCEPTUAL DEFINITION OF C++ CLASS AND OBJECT MODULARITY.....	44
3.2.1 Prerequisites	44
3.2.2 Sub-characterisation of object-oriented software modularity.....	44
3.2.2.1 Direct mapping	46
3.2.2.2 Information Hiding	47
3.2.2.3 Few Interfaces.....	50
3.2.2.4 Explicit Interfaces.....	52
3.2.2.5 Small Interfaces	54
3.2.3 Conceptual definition of modularity sub-characteristics	59
3.2.3.1 Interface dependence sub-characteristic	60
3.2.3.2 External relationships sub-characteristic	62
3.2.3.3 Connection obscurity sub-characteristic	64
3.2.3.4 Dependency sub-characteristic	67
3.2.4 Validation of the sub-characterisation.....	69
3.3. CONCLUSION.....	69
4. ENTITY MODELLING	71
4.1. STAGE 2 OF MEASURE DEVELOPMENT PROCESS - ENTITY MODELLING	71
4.1.1 Prerequisites to the entity modelling stage	74
4.1.2 Performance of the entity modelling stage.....	74
4.1.3 Products of the entity modelling stage	75
4.1.4 Assessing mathematical model validity	76
4.1.5 Practical Considerations.....	76
4.2. ENTITY MODELLING OF C++ CLASS AND OBJECT MODULARITY	78
4.2.1 Prerequisites	79
4.2.2 Selection of the mathematical model.....	79
4.2.3 C++ class modularity	81
4.2.3.1 C++ class module model.....	81
4.2.3.2 Interface dependence sub-characteristic of C++ class modularity	84
4.2.3.3 External relationships sub-characteristic of C++ class modularity	86
4.2.3.4 Connection obscurity sub-characteristic of C++ class modularity.....	88
4.2.3.5 Dependency sub-characteristic of C++ class modularity	92
4.2.3.6 Class modularity sub-characteristics sharing identical feature descriptions	95

4.2.4	<i>C++ object modularity</i>	96
4.2.4.1	C++ object module model.....	96
4.2.4.2	Interface dependence sub-characteristic of C++ object modularity	102
4.2.4.3	External relationships sub-characteristic of C++ object modularity	106
4.2.4.4	Connection obscurity sub-characteristic of C++ object modularity	109
4.2.4.5	Dependency sub-characteristic of C++ object modularity	116
4.2.4.6	Summary: object mathematical model completeness.....	119
4.2.4.7	Object modularity sub-characteristics sharing identical feature descriptions	121
4.3.	CONCLUSION.....	122
5.	OPERATIONAL MEASURE DEFINITION	124
5.1.	STAGE 3 OF MEASURE DEVELOPMENT PROCESS - OPERATIONAL DEFINITION	124
5.1.1	<i>Prerequisites to operational definition stage</i>	126
5.1.2	<i>Performance of operational definition stage</i>	126
5.1.3	<i>Product of operational definition stage</i>	126
5.1.4	<i>Assessing measure level of measurement and validity</i>	128
5.1.5	<i>Practical Considerations</i>	128
5.2.	OPERATIONAL DEFINITION OF C++ CLASS AND OBJECT MODULARITY	129
5.2.1	<i>Prerequisites</i>	130
5.2.2	<i>Mathematical measure definition notation</i>	130
5.2.3	<i>Measures of C++ class modularity</i>	132
5.2.3.1	Measures of C++ class interface dependence	133
5.2.3.2	Measures of C++ class external relationships	139
5.2.3.3	Measures of C++ class connection obscurity	141
5.2.3.4	Measures of C++ class dependency	145
5.2.4	<i>Measures of C++ object modularity</i>	149
5.2.4.1	Measures of C++ object interface dependence	149
5.2.4.2	Measures of C++ object external relationships	154
5.2.4.3	Measures of C++ object connection obscurity	156
5.2.4.4	Measures of C++ object dependency	163
5.2.5	<i>Level of measurement of modularity measures</i>	168
5.2.6	<i>Content validation of modularity measures</i>	169
5.2.6.1	Example of a content validation	170
5.3.	CONCLUSION.....	173
6.	MEASUREMENT INSTRUMENT IMPLEMENTATION	175
6.1.	STAGE 4 OF MEASURE DEVELOPMENT PROCESS - MEASUREMENT INSTRUMENT IMPLEMENTATION	175
6.1.1	<i>Prerequisites to the measurement instrument implementation stage</i>	178
6.1.2	<i>Performance of the measurement instrument implementation stage</i>	178
6.1.3	<i>Products of the measurement instrument implementation stage</i>	179
6.1.4	<i>Assessing implemented measure level of measurement and validity</i>	179
6.1.5	<i>Reliability of software based measurement instruments</i>	180
6.1.6	<i>Practical Considerations</i>	181
6.2.	IMPLEMENTATION OF C++ CLASS AND OBJECT MODULARITY MEASURES	182
6.2.1	<i>C++ basic software model implementation</i>	184
6.2.1.1	Natural language C++ basic software model	184
6.2.1.2	Mathematical C++ basic software model.....	185
6.2.1.3	Implementation of C++ basic software model	187
6.2.2	<i>Software measurement model implementation</i>	189
6.2.2.1	Derivation of C++ software modularity measurement model	189
6.2.2.2	Implementation of C++ software modularity measurement model	190
6.2.3	<i>Software modularity measures implementation</i>	192
6.2.3.1	Class external relationships measure implementation.....	195
6.2.3.2	Object external relationships measure implementation.....	196
6.2.3.3	Object connection obscurity measure implementation.....	197
6.2.3.4	Object dependency measure implementation.....	200
6.2.4	<i>Level of measurement and content validity of implemented measures</i>	201
6.2.4.1	Level of measurement.....	201
6.2.4.2	Content validity	201
6.3.	CONCLUSION.....	202

7.	APPLICATION OF C++ CLASS AND OBJECT MODULARITY MEASURES.....	204
7.1.	DATA REDUCTION TECHNIQUES.....	206
7.1.1	Calculation of an Aggregate of Modularity.....	206
7.1.1.1	Preliminary calculations.....	207
7.1.1.2	Class and object modularity aggregate calculation.....	209
7.1.1.3	Class and object modularity weighted aggregate calculation.....	213
7.1.2	Calculation of a Modularity Distance.....	215
7.2.	CONTENT VALIDATION – EMULEPLUS SOFTWARE SYSTEM.....	216
7.2.1	eMulePlus object interface dependence.....	218
7.2.2	eMulePlus object external relationships.....	219
7.2.3	eMulePlus object connection obscurity.....	219
7.2.4	eMulePlus object dependency.....	222
7.3.	EXAMPLE OF A CONSTRUCT VALIDATION – EMULEPLUS SOFTWARE SYSTEM.....	224
7.4.	MODULARITY MEASUREMENT CASE STUDIES.....	236
7.4.1	Case study 1 - modularity of the eMulePlus software system.....	237
7.4.1.1	Aims.....	237
7.4.1.2	Application.....	237
7.4.1.3	Analysis and interpretation.....	238
7.4.1.4	Discussion.....	245
7.4.2	Case study 2 - interface dependence of CPartFile class and object modules.....	247
7.4.2.1	Aims.....	247
7.4.2.2	Application.....	247
7.4.2.3	Analysis and interpretation.....	248
7.4.2.4	Discussion.....	266
7.5.	CONCLUSIONS.....	266
8.	DISCUSSION AND SUGGESTIONS FOR FURTHER WORK.....	269
8.1.	RESEARCH OUTCOMES.....	269
8.1.1	Systematic process of software descriptive measure development.....	270
8.1.2	Descriptive measures of C++ class and object modularity.....	274
8.1.3	Case study – eMulePlus software system.....	277
8.1.4	Summary – research contributions and considerations.....	278
8.1.4.1	Contributions.....	278
8.1.4.2	Considerations.....	279
8.1.5	Further work.....	280
1.	APPENDIX 1 - ENTITY-RELATIONSHIP MODEL SET DEFINITIONS.....	283
1.1.	ENTITIES.....	283
1.2.	BASIC RELATIONSHIPS.....	285
1.3.	DERIVED RELATIONSHIPS.....	289
2.	APPENDIX 2 - BASIC MODEL TO MEASUREMENT MODEL TRANSFORMATIONS.....	294
2.1.	TRANSFORMATION 1 - AA AND IAA.....	294
2.2.	TRANSFORMATION 2 - AM AND IAM.....	296
2.3.	TRANSFORMATION 3 - AO AND IAO.....	296
2.4.	TRANSFORMATION 4 - IMO.....	297
2.5.	TRANSFORMATION 5 - MICREADA.....	297
2.6.	TRANSFORMATION 6 - MICWRITEA.....	299
2.7.	TRANSFORMATION 7 - MICINVM.....	299
2.8.	TRANSFORMATION 8 - MIOCREADA.....	300
2.9.	TRANSFORMATION 9 - MIOCWRITEA.....	300
2.10.	TRANSFORMATION 10 - MIOCINVM.....	301
2.11.	TRANSFORMATION 11 - CEF.....	302
2.12.	TRANSFORMATION 12 - IDA.....	302
2.13.	TRANSFORMATION 13 - CIF.....	303
2.14.	TRANSFORMATION 14 - FIF.....	303
3.	APPENDIX 3 – EMULEPLUS C++ CLASS MODULARITY TO LINES OF CODE CORRELATION DATA.....	304

4.	APPENDIX 4 - EMULEPLUS SOFTWARE SYSTEM CONTENT VALIDATION	308
4.1.	EMULEPLUS CLASS MODULARITY	308
4.2.	EMULEPLUS OBJECT MODULARITY	312
4.3.	EMULEPLUS CONTENT VALIDATION MEASURED VALUES.....	317
5.	APPENDIX 5 – EMULEPLUS MODULARITY DATA VALUES	322
6.	APPENDIX 6 - CPARTFILE INTERFACE DEPENDENCE DATA	334
6.1.	CLASS CPARTFILE	334
6.2.	OBJECT CPARTFILE	344
	BIBLIOGRAPHY	356

List of Figures

FIGURE 1-1 RELATIONSHIPS BETWEEN ENTITIES, CHARACTERISTICS, FEATURES AND DESCRIPTIVE MEASURES	4
FIGURE 1-2 RELATIONSHIPS BETWEEN A SOURCE CODE ENTITY AND DESCRIPTIVE MEASURES THAT QUANTIFY LINES OF CODE TO DESCRIBE SOURCE CODE SIZE	5
FIGURE 1-3 SYSTEMATIC PROCESS OF SOFTWARE DESCRIPTIVE MEASURE DEVELOPMENT	7
FIGURE 1-4 PATHS OF DESCRIPTIVE MEASURE VALIDATION	12
FIGURE 1-5 STEPS OF DESCRIPTIVE MEASURE CONTENT VALIDATION. ADAPTED FROM SPROULL (1995, P.79).	15
FIGURE 1-6 STEPS OF DESCRIPTIVE MEASURE CONTENT VALIDATION. ADAPTED FROM SPROULL (1995, P.79).	17
FIGURE 2-1 CONTRIBUTION OF SOFTWARE DESCRIPTIVE MEASURE DEVELOPMENT PROCESS TO SOFTWARE QUALITY METRICS FRAMEWORK (IEEE COMPUTER SOCIETY 1998, P. 4).	24
FIGURE 2-2 CONTRIBUTION OF SOFTWARE DESCRIPTIVE MEASURE DEVELOPMENT PROCESS TO TAME GQM PARADIGM (BASILI 1988).....	25
FIGURE 3-1 CONCEPTUAL DEFINITION STAGE OF THE MEASURE DEVELOPMENT PROCESS.....	37
FIGURE 3-2 PROCESS OF CONCEPTUAL DEFINITION OF CHARACTERISTIC TO BE DESCRIBED BY MEASURES..	42
FIGURE 3-3 REPRESENTATION OF AN OBJECT ORIENTED MODULE WITH A HIGH LEVEL OF INFORMATION HIDING	48
FIGURE 3-4 REPRESENTATION OF OBJECT ORIENTED MODULE EXTERNAL RELATIONSHIPS	51
FIGURE 3-5 REPRESENTATION OF OBJECT ORIENTED MODULE CONNECTION OBSCURITY	53
FIGURE 3-6 REPRESENTATION OF OBJECT ORIENTED MODULE DEPENDENCY.....	55
FIGURE 3-7 MODULARITY SUB-CHARACTERISATION BASED ON MEYER'S (1997) RULES OF MODULARITY ..	57
FIGURE 3-8 INTERFACE DEPENDENCE SUB-CHARACTERISATION AND CONCEPTUAL DEFINITION.....	60
FIGURE 3-9 EXTERNAL RELATIONSHIPS SUB-CHARACTERISATION AND CONCEPTUAL DEFINITION	62
FIGURE 3-10 CONNECTION OBSCURITY SUB-CHARACTERISATION AND CONCEPTUAL DEFINITION	64
FIGURE 3-11 DEPENDENCY SUB-CHARACTERISATION AND CONCEPTUAL DEFINITION	67
FIGURE 4-1 ENTITY MODELLING STAGE OF THE MEASURE DEVELOPMENT PROCESS	71
FIGURE 4-2 NATURAL LANGUAGE AND MATHEMATICAL ENTITY MODELS EXPRESS THE THEORETICAL BASIS OF DESCRIPTIVE MEASURES.....	72
FIGURE 4-3 PROCESS OF NATURAL LANGUAGE MODELLING OF A SOFTWARE ENTITY.....	75
FIGURE 4-4 C++ CLASS INTERFACE AND HIDDEN ELEMENTS	83
FIGURE 4-5 C++ CLASS MATHEMATICAL MODEL.....	83
FIGURE 4-6 C++ CLASS INTERFACE DEPENDENCE MATHEMATICAL MODEL.....	86
FIGURE 4-7 C++ CLASS EXTERNAL RELATIONSHIPS MATHEMATICAL MODEL	88
FIGURE 4-8 C++ CLASS CONNECTION OBSCURITY MATHEMATICAL MODEL.....	91
FIGURE 4-9 C++ CLASS DEPENDENCY MATHEMATICAL MODEL.....	94
FIGURE 4-10 C++ OBJECT-CLASS INTERFACE AND HIDDEN ELEMENTS	99
FIGURE 4-11 C++ OBJECT-CLASS MATHEMATICAL MODEL.....	100
FIGURE 4-12 SOLUTION ADOPTED TO MULTIPLE ATTRIBUTES AND METHODS INHERITED FROM THE SAME CLASS.....	101
FIGURE 4-13 C++ OBJECT INTERFACE DEPENDENCE MATHEMATICAL MODEL	105
FIGURE 4-14 C++ OBJECT EXTERNAL RELATIONSHIPS MATHEMATICAL MODEL	108
FIGURE 4-15 C++ OBJECT CONNECTION OBSCURITY MATHEMATICAL MODEL.....	114
FIGURE 4-16 C++ OBJECT DEPENDENCY MATHEMATICAL MODEL	118
FIGURE 5-1 OPERATIONAL DEFINITION STAGE OF THE MEASURE DEVELOPMENT PROCESS	124
FIGURE 5-2 CHARACTERISTIC TO MEASURE RELATIONSHIP (CHARMER) DIAGRAM	127
FIGURE 5-3 C++ CLASS INTERFACE DEPENDENCE CHARACTERISTIC TO MEASURE RELATIONSHIP (CHARMER) DIAGRAM.....	138
FIGURE 5-4 C++ CLASS EXTERNAL RELATIONSHIPS CHARMER DIAGRAM.....	141
FIGURE 5-5 C++ CLASS CONNECTION OBSCURITY CHARMER DIAGRAM.....	144
FIGURE 5-6 C++ CLASS DEPENDENCY CHARMER DIAGRAM	147
FIGURE 5-7 C++ OBJECT INTERFACE DEPENDENCE CHARMER DIAGRAM	153
FIGURE 5-8 C++ OBJECT EXTERNAL RELATIONSHIPS CHARMER DIAGRAM.....	156
FIGURE 5-9 C++ OBJECT CONNECTION OBSCURITY CHARMER DIAGRAM - PART 1.....	161
FIGURE 5-10 C++ OBJECT CONNECTION OBSCURITY CHARMER DIAGRAM - PART 2.....	162
FIGURE 5-11 C++ OBJECT DEPENDENCY CHARMER DIAGRAM	166

FIGURE 5-12 EXAMPLE OF CONTENT VALIDATION OF OBJECT DEPENDENCY MEASURES USING CHARMER DIAGRAM	171
FIGURE 6-1 MEASUREMENT INSTRUMENT IMPLEMENTATION STAGE OF THE MEASURE DEVELOPMENT PROCESS.....	175
FIGURE 6-2 ELEMENTS OF A SOFTWARE BASED MEASUREMENT INSTRUMENT.....	177
FIGURE 6-3 MEASUREMENT INSTRUMENT IMPLEMENTING C++ CLASS AND OBJECT MODULARITY MEASURES	182
FIGURE 6-4 BASIC SOFTWARE MATHEMATICAL MODEL.....	186
FIGURE 6-5 IMPLEMENTATION OF BASIC SOFTWARE MODEL.....	187
FIGURE 6-6 UPDATE OF FIGURE 5-6 C++ CLASS EXTERNAL RELATIONSHIPS CHARMER DIAGRAM, REFLECTING MEASUREMENT INSTRUMENT IMPLEMENTATION	195
FIGURE 6-7 UPDATE OF FIGURE 5-10 C++ OBJECT EXTERNAL RELATIONSHIPS CHARMER DIAGRAM, REFLECTING MEASUREMENT INSTRUMENT IMPLEMENTATION	197
FIGURE 6-8 UPDATE OF FIGURE 5-11 C++ OBJECT CONNECTION OBSCURITY CHARMER DIAGRAM, REFLECTING MEASUREMENT INSTRUMENT IMPLEMENTATION	198
FIGURE 6-9 UPDATE OF FIGURE 5-12 C++ OBJECT CONNECTION OBSCURITY CHARMER DIAGRAM, REFLECTING MEASUREMENT INSTRUMENT IMPLEMENTATION	199
FIGURE 6-10 UPDATE OF FIGURE 5-13 C++ OBJECT DEPENDENCY CHARMER DIAGRAM, REFLECTING MEASUREMENT INSTRUMENT IMPLEMENTATION	200
FIGURE 7-1 CHARMER DIAGRAM SHOWING CONTENT VALIDATION FOR eMULEPLUS MEASURES DESCRIBING OBJECT CONNECTION OBSCURITY.....	220
FIGURE 7-2 CHARMER DIAGRAM SHOWING CONTENT VALIDATION FOR eMULEPLUS MEASURES DESCRIBING OBJECT DEPENDENCY.....	222
FIGURE 7-3 EFFECT OF MEASURE CIS5 ON eMULEPLUS UNWEIGHTED CLASS MODULARITY AGGREGATE VALUES	227
FIGURE 7-4 EFFECT OF MEASURE CIS5 ON eMULEPLUS WEIGHTED CLASS MODULARITY AGGREGATE VALUES	227
FIGURE 7-5 EFFECT OF MEASURE CIS5 ON eMULEPLUS CLASS MODULARITY DISTANCE VALUES	228
FIGURE 7-6 COMPARISON OF WEIGHTED AND UNWEIGHTED MODULARITY AGGREGATE AND MODULARITY DISTANCE INDICATORS OF eMULEPLUS CLASS MODULARITY	229
FIGURE 7-7 eMULEPLUS CLASS LINES OF CODE AND UNWEIGHTED MODULARITY AGGREGATE OUTLIERS	231
FIGURE 7-8 eMULEPLUS CLASS LINES OF CODE AND WEIGHTED MODULARITY AGGREGATE OUTLIERS.....	232
FIGURE 7-9 eMULEPLUS CLASS LINES OF CODE AND MODULARITY EUCLIDEAN DISTANCE OUTLIERS	232
FIGURE 7-10 FREQUENCY HISTOGRAM OF eMULEPLUS CLASS MODULARITY NORMALISED AGGREGATE VALUES	240
FIGURE 7-11 FREQUENCY HISTOGRAM OF eMULEPLUS OBJECT MODULARITY AGGREGATES	243
FIGURE 7-12 REPRESENTATION OF AN OBJECT ORIENTED MODULE WITH LOW INTERFACE DEPENDENCE...	249
FIGURE 7-13 GRAPHICAL REPRESENTATION OF CPARTFILE CLASS INTERFACE DEPENDENCE.....	251
FIGURE 7-14 CLASS CPARTFILE INTERFACE METHODS WITH MORE THAN 32 LINES OF CODE	254
FIGURE 7-15 CLASS CPARTFILE INTERFACE METHOD INVOCATIONS AND ATTRIBUTE ACCESSES.....	255
FIGURE 7-16 CLASS CPARTFILE INTERFACE METHODS THAT POTENTIALLY INCREASE INTERFACE DEPENDENCE.....	256
FIGURE 7-17 OBJECT CPARTFILE INHERITS ELEMENTS FROM THREE ANCESTOR CLASSES.....	258
FIGURE 7-18 GRAPHICAL REPRESENTATION OF CPARTFILE OBJECT INTERFACE DEPENDENCE	259
FIGURE 7-19 OBJECT CPARTFILE INTERFACE METHODS WITH MORE THAN 28 LINES OF CODE	262
FIGURE 7-20 OBJECT CPARTFILE INTERFACE METHOD INVOCATIONS AND ATTRIBUTE ACCESSES.....	263
FIGURE 7-21 OBJECT CPARTFILE INTERFACE METHODS THAT POTENTIALLY INCREASE INTERFACE DEPENDENCE.....	264
FIGURE APPENDIX 4-1 eMULEPLUS CLASS INTERFACE DEPENDENCE CHARMER DIAGRAM CONTENT VALIDATION	308
FIGURE APPENDIX 4-2 eMULEPLUS CLASS EXTERNAL RELATIONSHIPS CHARMER DIAGRAM CONTENT VALIDATION	309
FIGURE APPENDIX 4-3 eMULEPLUS CLASS CONNECTION OBSCURITY CHARMER DIAGRAM CONTENT VALIDATION	310
FIGURE APPENDIX 4-4 eMULEPLUS CLASS DEPENDENCY CHARMER DIAGRAM CONTENT VALIDATION ..	311
FIGURE APPENDIX 4-5 eMULEPLUS OBJECT INTERFACE DEPENDENCE CHARMER DIAGRAM CONTENT VALIDATION	312
FIGURE APPENDIX 4-6 eMULEPLUS OBJECT EXTERNAL RELATIONSHIPS CHARMER DIAGRAM CONTENT VALIDATION	313

FIGURE APPENDIX 4-7 EMULEPLUS OBJECT CONNECTION OBSCURITY CHARMER DIAGRAM CONTENT VALIDATION - PART 1	314
FIGURE APPENDIX 4-8 EMULEPLUS OBJECT CONNECTION OBSCURITY CHARMER DIAGRAM CONTENT VALIDATION - PART 2	315
FIGURE APPENDIX 4-9 EMULEPLUS OBJECT DEPENDENCY CHARMER DIAGRAM CONTENT VALIDATION	316

List of Tables

TABLE 4-1 CLASS MODULARITY SUB-CHARACTERISTICS SHARING COMMON NATURAL LANGUAGE MODEL FEATURES	95
TABLE 4-2 RULES FOR THE ASSIGNMENT OF C++ OBJECT-CLASS ELEMENT LEVELS OF PROTECTION	98
TABLE 4-3 SHORTCOMINGS OF C++ OBJECT MODULARITY MATHEMATICAL MODELS	120
TABLE 4-4 OBJECT MODULARITY SUB-CHARACTERISTICS SHARING COMMON NATURAL LANGUAGE MODEL FEATURES	122
TABLE 5-1 MEASURES OF C++ CLASS INTERFACE ELEMENT INTERDEPENDENCE.....	134
TABLE 5-2 MEASURES OF C++ CLASS INTERFACE ELEMENT INTERDEPENDENCE (CONT.).....	135
TABLE 5-3 MEASURES OF C++ CLASS INTERFACE SIZE	136
TABLE 5-4 MEASURES OF C++ CLASS DATA EXPOSURE.....	137
TABLE 5-5 MEASURES OF CLASS EXTERNAL RELATIONSHIPS	139
TABLE 5-6 MEASURES OF CLASS EXTERNAL RELATIONSHIPS (CONT.).....	140
TABLE 5-7 MEASURES OF CLASS UNSTATED RELATIONSHIP.....	142
TABLE 5-8 MEASURES OF CLASS DISTANT CONNECTION.....	142
TABLE 5-9 MEASURES OF CLASS UNEXPECTED RELATIONSHIP	143
TABLE 5-10 MEASURES OF CLASS CONNECTION VIA NON-STANDARD INTERFACE	143
TABLE 5-11 MEASURES OF CLASS SERVICE INVOCATION	145
TABLE 5-12 MEASURES OF CLASS EXTERNAL VARIABLE READING.....	145
TABLE 5-13 MEASURES OF CLASS EXTERNAL FUNCTION WRITING	146
TABLE 5-14 MEASURES QUANTIFYING FEATURES COMMON TO SEVERAL CLASS MODULARITY SUB-CHARACTERISTICS.....	148
TABLE 5-15 MEASURES OF OBJECT INTERFACE ELEMENT INTERDEPENDENCE.....	149
TABLE 5-16 MEASURES OF OBJECT INTERFACE ELEMENT INTERDEPENDENCE (CONT.)	150
TABLE 5-17 MEASURES OF OBJECT INTERFACE SIZE	151
TABLE 5-18 MEASURES OF OBJECT DATA EXPOSURE.....	152
TABLE 5-19 MEASURES OF OBJECT EXTERNAL RELATIONSHIPS	154
TABLE 5-20 MEASURES OF OBJECT EXTERNAL RELATIONSHIPS (CONT.).....	155
TABLE 5-21 MEASURES OF OBJECT VARIABLE CONNECTION	157
TABLE 5-22 MEASURES OF OBJECT UNSTATED RELATIONSHIP.....	157
TABLE 5-23 MEASURES OF OBJECT DISTANT CONNECTION.....	158
TABLE 5-24 MEASURES OF OBJECT UNEXPECTED RELATIONSHIP	158
TABLE 5-25 MEASURES OF OBJECT UNEXPECTED RELATIONSHIP (CONT.)	159
TABLE 5-26 MEASURES OF OBJECT CONNECTION VIA NON-STANDARD INTERFACE	159
TABLE 5-27 MEASURES OF OBJECT CONNECTION VIA NON-STANDARD INTERFACE (CONT.).....	160
TABLE 5-28 MEASURES OF OBJECT SERVICE INVOCATION	163
TABLE 5-29 MEASURES OF OBJECT INTERFACE PROVISION.....	163
TABLE 5-30 MEASURES OF OBJECT EXTERNAL VARIABLE READING.....	164
TABLE 5-31 MEASURES OF OBJECT EXTERNAL FUNCTION WRITING	165
TABLE 5-32 MEASURES QUANTIFYING FEATURES COMMON TO SEVERAL OBJECT MODULARITY SUB-CHARACTERISTICS.....	167
TABLE 6-1 C++ SOFTWARE MODULARITY MEASUREMENT MODEL SETS DERIVED FROM C++ BASIC SOFTWARE MODEL SOURCE SETS	190
TABLE 6-2 IMPLEMENTATION OF C++ MATHEMATICAL ENTITY MODELS AS C++ SOFTWARE MODULARITY MEASUREMENT MODEL	191
TABLE 6-3 MEASURES OF C++ OBJECT MODULARITY OMITTED FROM MEASUREMENT INSTRUMENT IMPLEMENTATION	194
TABLE 6-4 CHARACTERISTIC TO MEASURE RELATIONSHIP (CHARMER) DIAGRAMS DESCRIBING IMPLEMENTED MEASURES	202
TABLE 7-1 RESULTS OF eMULEPLUS SYSTEM CONTENT VALIDATION	223
TABLE 7-2 eMULEPLUS CLASS LINES OF CODE, MODULARITY AGGREGATE AND MODULARITY DISTANCE DISTRIBUTION STATISTICS	230
TABLE 7-3 STRENGTH OF RELATIONSHIPS BETWEEN eMULEPLUS CLASS LINES OF CODE, UNWEIGHTED AND WEIGHTED MODULARITY AGGREGATES AND MODULARITY DISTANCE	233
TABLE 7-4 eMULEPLUS SYSTEM CLASS MODULARITY AGGREGATE STATISTICS	239
TABLE 7-5 eMULEPLUS MODULARITY AGGREGATES OF CLASSES WITH RELATIVELY LOW MODULARITY ..	241
TABLE 7-6 eMULEPLUS SYSTEM OBJECT MODULARITY AGGREGATE STATISTICS	242

TABLE 7-7 eMULEPLUS NORMALISED MODULARITY AGGREGATES OF OBJECT-CLASSES WITH POTENTIALLY LOW MODULARITY	245
TABLE 7-8 SUMMARY OF CLASS CPartFile INTERFACE SIZE MEASUREMENT DATA.....	253
TABLE 7-9 SUMMARY OF OBJECT CPartFile INTERFACE SIZE MEASUREMENT DATA.....	261
TABLE APPENDIX 3-1 eMULEPLUS SOFTWARE SYSTEM CLASS MODULARITY MEASURE TO CLASS LINES OF CODE CORRELATION.....	304
TABLE APPENDIX 4-1 eMULEPLUS SOFTWARE SYSTEM CONTENT VALIDATION MEASURED VALUES	317
TABLE APPENDIX 5-1 eMULEPLUS SOFTWARE SYSTEM CLASS LINES OF CODE, UNWEIGHTED AND WEIGHTED MODULARITY AGGREGATE AND MODULARITY EUCLIDEAN DISTANCE VALUES	322
TABLE APPENDIX 5-2 eMULEPLUS SOFTWARE SYSTEM CLASS AND OBJECT MODULARITY AGGREGATE VALUES	328

Abstract

Software measurement has been of interest to software engineers for almost as long as software has been developed. While the evolution of systematic processes of software development has seen a trend away from reliance on the expertise of individual software developers alone to ensure software quality, systematic processes of software measure development have not evolved to a similar extent. The problem with defining software measures according to an informal process is that the quality of measures can be highly dependent on the expertise of the individual measure developers. If a systematic process of software measure development were defined, that promoted transparency and objectivity in measure development, then this systematic process could support the development of high quality measures by less expert users.

In this thesis, a systematic process of software descriptive measure development is described and demonstrated. The approach taken to defining this systematic process is to investigate the various processes by which currently available software descriptive measures have been developed. These processes are then amalgamated with an established systematic method of measure development used in the field of social science. Applying the stages of measure development thus identified to the task of developing measures to describe C++ class and object modularity tests the feasibility of this measure development process. Insights gained through this testing provide feedback to further refine the process. In this way, a systematic process of descriptive software measure development is defined alongside the definition of a set of measures that provide a detailed description of the complex software characteristic of modularity. The products of each stage of this measure development process assist a user to validate the measures with respect to an intended application, and to analyse and interpret the measurement data obtained by applying the measures to a software system. This is demonstrated in a case study that also provides an indirect indication of the quality of the process by which the measures were developed.

The major contribution of this work is the systematic process of descriptive software measure development, as it has a wide application and can be used to develop measures to describe many software characteristics of interest. A second important contribution is made by the set of measures of C++ class and object modularity developed to demonstrate this systematic descriptive measure development process.