Creative Software Development: An Empirical Modelling Framework

by

Paul Edward Ness

Thesis

Submitted to the The University of Warwick

in partial fulfilment of the requirements

for admission to the degree of **Doctor of Philosophy**

Department of Computer Science

University of Warwick

October 1997

Contents

List of Examples viii

List of Tables xi

List of Figures xii

Acknowledgments xiii

Declarations xiv

Abstract xv

Chapter 1 Introduction 1

1.1 Preliminaries 1

1.2 Aims and motivations 2

1.3 Methodological challenges 4

1.4 Sources of illustrative material 5

1.5 Thesis outline 7

1.6 Abbreviations 9

Chapter 2 Background to EM, PD and SD 11

2.1 Empirical Modelling 11

2.1.1 General concepts and principles 12

2.1.2 Agentless systems 17

2.1.3 Single-agent systems and modelling 19

2.1.4 Computer as artefact 21

2.1.5 Multi-agent systems and modelling 23
2.1.6 Agent concept 25
2.1.7 Conceptualization 26
2.1.8 Situating EM 28
2.2 Product design 29
2.3 Software development 32

Chapter 3 Characterization of EM, PD and SD **38**

3.1 Background 38
3.2 Artefacts 39
3.3 Subjects 45
3.4 Actions 47
3.5 Constraints 53
3.6 Environments 56
3.7 Knowledge 59
3.8 Summary and conclusion 62

Chapter 4 Artefacts of EM, PD and SD **65**

4.1 Definition 65
4.2 Need for artefacts and how they help cognition 67
4.3 Characterization of artefacts 68
4.3.1 Novelty and familiarity 71
4.3.2 Ambiguity and unambiguity 72
4.3.3 Implicit meaningfulness and explicit meaning 73
4.3.4 Emergence and completeness 74
4.3.5 Incongruity and congruity 75
4.3.6 Divergence and convergence 76
4.4 Further characterizations of artefacts 77
4.4.1 Construals 77
4.4.2 Design drawings 79
4.5 Summary 82
Appendix: Illustrative examples for Section 4.3 83

Chapter 5 Actions of EM, PD and SD 97

5.1 Definition 97

5.2 Generative actions 99

5.2.1 Retrieval 100

5.2.2 Association 101

5.2.3 Synthesis 102

5.2.4 Transformation 103

5.2.5 Analogical transfer 103

5.2.6 Categorical reduction 104

5.3 Exploratory actions 105

5.3.1 Creative exploration and SD 106

5.3.2 Attribute finding 107

5.3.3 Conceptual interpretation 108

5.3.4 Functional inference 108

5.3.5 Contextual shifting 109

5.3.6 Hypothesis testing 110

5.3.7 Searching for limitations 111

5.4 Further characterizations of actions 112

5.4.1 Observation and experimentation 112

5.4.2 Methods and methodology 114

5.5 Summary 115

Appendix: Illustrative examples for Sections 5.2 and 5.3 117

Chapter 6 SD as Systems Development 137

6.1 Generalizing computers, programs and programming 137

6.1.1 Computer as artefact 138

6.1.2 Program as configuration 141

6.1.3 Programming as configuring 145

6.2 Addressing the essential difficulties of software 149

6.2.1 Complexity 149

6.2.2 Conformity 151

6.2.3 Changeability 152

6.2.4 Invisibility 154
6.3 Attacks on the essential difficulties of software 155
6.3.1 Buy versus build 155
6.3.2 Requirements refinement and rapid prototyping 156
6.3.3 Incremental development - grow don't build software 158
6.3.4 Great designers 159
6.4 Software and SD in the future 160
6.4.1 Networked computing and concurrency 161
6.4.2 Software agents 162
6.4.3 Object standards and technology 162
6.4.4 Product-oriented development 164
6.5 Requirements engineering in the future 165
6.5.1 Emerging importance of context 165
6.5.2 End of requirements as contract 168
6.5.3 Supporting market-driven inventors 169
6.5.4 Coping with incompleteness 170
6.5.5 Integrating design artefacts 171
6.5.6 Making requirements methods and tools more accessible . . . 171
6.6 Conclusion 172
6.7 Limitations of EM for developing software 175
6.7.1 Quality 176
6.7.2 Management 177
6.7.3 Methodology 179
6.7.4 Scale 180

Chapter 7 Conclusions and further work **182**

7.1 Conclusions 182
7.2 Further work 184
7.2.1 Software engineering 184
7.2.2 EM in context 188
7.2.3 Distribution of EM tools 189

Bibliography **191**

Appendix A Empirical Modelling of a Sailboat **206**

A.1 Common-sense knowledge 206

A.2 Agent-oriented modelling 207

A.3 Observation-oriented modelling 207

A.4 Definitive representation of the sailboat 209

A.5 Exploring the sailboat simulation 213

A.6 Extending the sailboat model 214

Appendix B Experiences Using the Shlaer-Mellor Method **217**

B.1 Overview of the Shlaer-Mellor method 218

B.2 Domain analysis 219

B.3 Using the Teamwork tool in analysis and design 219

B.4 Automatic code generation 220

B.5 Testing 221

B.6 Conclusion 222

Appendix C SUL, MUL and Hydrolift Artefacts **223**

C.1 SUL artefacts 223

C.1.1 SUL LSD specification 223

C.1.2 SUL visualization/animation 224

C.1.3 SUL DoNaLD script 224

C.1.4 SUL ADM script 227

C.1.5 SUL sketch 228

C.1.6 SUL statement of requirements 228

C.2 MUL artefacts 229

C.2.1 MUL LSD specification 229

C.2.2 MUL visualization/animation 230

C.2.3 MUL DoNaLD redefinitions 230

C.2.4 MUL ADM redefinitions 230

C.2.5 MUL sketch 231

C.2.6 MUL statement of requirements and models 231

C.3 Hydrolift artefacts 232
C.3.1 Hydrolift LSD specification 232
C.3.2 Hydrolift visualization/animation 233
C.3.3 Hydrolift DoNaLD redefinitions 233
C.3.4 Hydrolift ADM redefinitions 235
C.3.5 Hydrolift sketch 236
C.3.6 Statement of requirements and models 236

Appendix D Reviews **238**

Creative Cognition: Theory, Research, and Applications 238
Total Design: Integrated Methods for Successful Product Engineering 242
Creating Innovative Products Using Total Design: The Living Legacy
of Stuart Pugh 244

List of Examples

2.1 Observation and agent-oriented analysis in OXO 14
2.2 Definitive representation of state in OXO 16
2.3 0-agent view in OXO 19
2.4 1-agent view in OXO 20
2.5 Computer as artefact in OXO 22
2.6 n-agent view in OXO 24
2.7 Classification of agents in OXO 26
2.8 Conceptualization in OXO 27
2.9 Structure models in SD 35
2.10 Behaviour models in SD 35
2.11 Process models in SD 36

3.1 LSD specifications in EM 41
3.2 Visualizations in EM 41
3.3 Animations in EM 42
3.4 Sketches in PD 43
3.5 Structure, behaviour and process models in SD 44
3.6 Generative and exploratory actions in EM 49
3.7 Generative and exploratory actions in PD 50
3.8 Transformational actions in SD 52
3.9 Statement of requirements as constraint in SD 55
3.10 LSD specification as constraint in EM 56
3.11 Traditional environments in design 59

4.1 Familiarity in stating requirements 83
4.2 Familiarity in EM 84
4.3 Novelty in EM 85
4.4 Ambiguity in stating requirements 86
4.5 Unambiguity in SD 87
4.6 Unambiguity in EM 88
4.7 Ambiguity in EM 89
4.8 Explicit meaning in SD 90
4.9 Implicit meaningfulness in EM 91
4.10 Emergence and completeness in EM 92
4.11 Completeness in SD 93
4.12 Incongruity and congruity in PD 94
4.13 Incongruity and congruity in EM 95
4.14 Divergence and convergence in EM 96

5.1 Retrieval in EM 117
5.2 Retrieval in PD 118
5.3 Retrieval in SD 119
5.4 Association in EM 120
5.5 Association in PD 121
5.6 Association in SD 122
5.7 Synthesis in EM 123
5.8 Synthesis in SD 124
5.9 Transformation in SD 125
5.10 Transformation in EM 126
5.11 Transfer in EM 127
5.12 Transfer in PD 128
5.13 Transfer in SD 128
5.14 Reduction in EM 129
5.15 Reduction in SD 130
5.16 Lack of incentive for exploration in SD 131
5.17 Attribute finding in EM 132

5.18 Conceptual interpretation in EM 133
5.19 Interpretation in PD 134
5.20 Functional inference in EM 135
5.21 Contextual shifting in EM 136

7.1 Organization of observations in EM 187
7.2 Sequencing of observations in EM 188

A.1 Representing the modeller in the SBS 208
A.2 Representing the sail in the SBS 210
A.3 Representing the rig and hull in the SBS 211
A.4 Exploration in the SBS 214
A.5 Emergence in the SBS 215
A.6 Openness in the SBS 216

List of Tables

3.1 Elements of the product design specification 54

4.1 Creative and analytical properties 69

5.1 Generative and exploratory actions 99

6.1 Contrasting complexity in software and systems. 150
6.2 Contrasting conformity in software and systems. 152
6.3 Contrasting changeability in software and systems. 153
6.4 Contrasting invisibility in software and systems. 154
6.5 EM themes associated with buy versus build. 155
6.6 EM themes associated with requirements and prototyping. 157
6.7 EM themes associated with incremental development. 158
6.8 EM themes associated with great designers. 160
6.9 Patterns of large software systems: failure and success. 176

List of Figures

2.1 Tools and notations in EM 17

2.2 tkeden being used to model OXO game 22

2.3 Activity model for total design 30

3.1 Baltimore and Ohio Railroad drafting room 1899 59

3.2 Comparison of EM, PD and SD 63

5.1 Geneplore Model 98

5.2 Lift system components 134

D.1 Geneplore Model 240

D.2 Activity model for total design 243

Acknowledgments

I would like to thank my friend and supervisor, Meurig Beynon, for his support and guidance throughout the research and writing of this thesis and for his infectious enthusiasm for my ideas.

I would also like to thank my friends, both inside and outside the department, who made the good times great and the bad times bearable over the last few years, Anna Philippou especially.

Special thanks goes to my mother, father and brother for the help they have given me in so many different ways. I am indebted to my brother Steve who took on the unpleasant task of introducing me to computing.

Finally, thanks goes to the EPSRC and IBM who funded the research through a CASE award and to Hugh Darwen and Steve Russ for making it happen.

* * * * * * * *

Further thanks goes to Mike Holcombe for his constructive comments during the viva and to Meurig Beynon and Steve Russ for guiding me through the revisions.

Declarations

This thesis is presented in accordance with the regulations for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The work described in this thesis has been undertaken by myself except where otherwise stated.

The various aspects concerning the Empirical Modelling of the sailboat have been published in [NBY94]. The view of Empirical modelling expressed in this thesis has been represented in [BNR95]. The view of Empirical Modelling as a new paradigm for computing has been presented in [BN95].

Abstract

The commercial success of software development depends on innovation [Nar93a]. However, conventional approaches inhibit the development of innovative products that embody novel concepts. This thesis argues that this limitation of conventional software development is largely due to its use of analytical artefacts, and that other activities, notably Empirical Modelling and product design, avoid the same limitation by using creative artefacts. Analytical artefacts promote the methodical representation of familiar subjects whereas creative artefacts promote the exploratory representation of novel subjects. The subjects, constraints, environments and knowledge associated with a design activity are determined by the nature of its artefacts.

The importance of artefacts was discovered by examining the representation of different kinds of lift system in respect of Empirical Modelling, product design and software development. The artefacts were examined by identifying creative properties, as characterized in the theory of creative cognition [FWS92], together with their analytical counterparts. The processes of construction were examined by identifying generative and exploratory actions. It was found that, in software development, the artefacts were analytical and the processes transformational, whereas, in Empirical Modelling and product design, the artefacts were both creative and analytical, and the processes exploratory.

A creative approach to software development using both creative and analytical artefacts is proposed for the development of innovative products. This new approach would require a radical departure from the established ideas and principles of software development. The existing paradigm would be replaced by a framework based on Empirical Modelling. Empirical Modelling can be thought of as a situated approach to modelling that uses the computer in exploratory ways to construct artefacts. The likelihood of the new paradigm being adopted is assessed by considering how it addresses the topical issues in software development.