

Steven Kelly

Towards a Comprehensive
MetaCASE and CAME
Environment

Conceptual, Architectural, Functional
and Usability Advances in MetaEdit+



UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 1997

ABSTRACT

Kelly, Steven

Towards a Comprehensive MetaCASE and CAME Environment:

Conceptual, Architectural, Functional and Usability Advances in MetaEdit+
Jyväskylä: University of Jyväskylä, 1997. 267 p.

(Jyväskylä Studies in Computer Science, Economics and Statistics

ISSN 0357-9921; 41)

ISBN 951-39-0118-1

Finnish summary

Diss.

Computer Aided Software Engineering environments have shown some promise in meeting the challenge of the software crisis. CASE tools have however generally supported only one fixed method each, whereas organisations modify methods as they use them. This lack of flexibility in CASE tools is being addressed through configurable *metaCASE* tools, which allow modification of the methods supported by the CASE tool.

MetaCASE tools are still comparatively in their infancy, and have yet to be widely taken into use. To help them mature and become more usable, the process, concepts, and tools for configuring them to support methods should be improved, and they should be able to support a wider range of interlinked, evolving methods. In addition, the support offered to the normal CASE user must be improved to meet that now expected of normal CASE tools.

The thesis extrapolates from existing metaCASE tools, trends in methods and investigations of CASE tool use to present a set of three extensions needed for the next generation of metaCASE: *multiple tools* to represent and manipulate the same data to *multiple simultaneous users*, working as part of projects which employ *multiple methods*. Improvements are also needed in the basic underlying data models used in CASE, in particular concerning relationships and complex objects.

The research follows an evolutionary constructive research paradigm. A new data model, GOPRR, is designed to answer the needs of metaCASE. The proposed extensions of metaCASE are implemented and tested in the building of the GOPRR-based MetaEdit+ metaCASE environment and its repository.

Keywords: CASE environments, metamodelling, method engineering, computer aided method engineering (CAME), metaCASE, repository, matrix representations

ACM Computing Review Categories:

- D.2.1 Software Engineering: Requirement/Specifications:
Languages, Methodologies, Tools
- D.2.2 Software Engineering: Tools and Techniques:
Computer-aided software engineering (CASE)
- D.2.10 Software Engineering: Design:
Methodologies, Representation
- E.2 Data Storage Representations:
Linked representations
- H.2.1 Database Management: Logical Design:
Data models
- H.2.4 Database Management: Systems:
Concurrency
- H.5.2 Information Interfaces and Presentation: User Interfaces:
Evaluation/Methodology
- I.6.5 Simulation and Modelling: Model Development:
Modeling methodologies

Author's Address:

Steven Kelly
University of Jyväskylä
Department of Computer Science and Information Systems
P.O. Box 35
FIN-40351 Jyväskylä
Finland
Email: kelly@cs.jyu.fi
Fax: +358 14 603011

ACKNOWLEDGEMENTS

The early funding for this thesis was provided through the MetaPHOR project (funded by the Academy of Finland and the University of Jyväskylä). More recently, I have been funded as a member of the University of Jyväskylä's COMAS graduate school.

My research advisor, Professor Kalle Lyytinen, has given invaluable advice and direction on the overall structure and form of the research, in addition to useful comments and suggestions on early versions of the papers. The research forms a part of the MetaPHOR project, and as such has benefited from the experience of other members of the project. In particular, I am indebted to Kari Smolander, who was central in the formalisation of OPRR and the development of MetaEdit, and to Matti Rossi and Juha-Pekka Tolvanen for many discussions on how to make a practical metamodelling and CASE environment. In addition Pentti Marttiin, Veli-Pekka Tahvanainen, Harri Oinas-Kukkonen, Liu Hui, Minna Koskinen and Janne Kaipala have all contributed in creating the favourable conditions for research within the project.

I would like to extend my thanks for the useful and insightful comments of all the unknown reviewers of the papers, and the external reviewers of the thesis, Dr. John Venable and Ass. Prof. Benkt Wangler.

Behind every thesis seems to be one person in particular who has been influential in directing the student towards research. In my case this was Ivan Lowe, whom I met during my work with Wycliffe Bible Translators. As a linguist with a doctorate in physics he encouraged this budding linguist to get a doctorate: hopefully computers are as appropriate as physics! My sincere thanks for the initial encouragement and subsequent prayer and moral support.

Surrounding the research environment have been a number of people whose contribution to the success of research is often underestimated. I would thus especially like to thank Elina Isännäinen, who helped me and so many other foreign students; the department amanuensis, Mirja Tervo; its secretary, Kaarina Suonia; and laboratory engineer, Kari Kivelä, for their tireless work in keeping the department and this research project up and running.

An important factor in this research has been the chance to see the results being taken into real world use and receive feedback on them, through the commercialisation of MetaEdit+ by MetaCase Consulting. Thanks are due to all the board and staff of MetaCase Consulting for their assistance and intelligent management of this marriage of research and commerce, and in particular to Janne Luoma for his astonishing patience in the face of the inevitable bugs and missed deadlines. Perhaps his earlier valuable work on the implementation of the repository, along with Marko Somppi, has given him a rare insight into the real-world systems development process!

Finally, my particular thanks to my parents for their constant yet undemanding encouragement throughout my academic career; to my wife, Riitta, who has provided support, encouragement and a listening ear throughout; to my son, Joel, for the joy and love he shares and for saving me from working too hard; and to God, who makes it all worthwhile.

CONTENTS

CHAPTER 1 Introduction to the Research	11
1 Introduction.....	12
2 Background	14
2.1 Terminology.....	14
2.2 Definition of the ISD process.....	19
2.3 Processes and products	22
2.4 Concepts and representations	24
3 Current situation and related research	25
3.1 Current tool support.....	25
3.2 Problems.....	38
4 Research problem.....	49
4.1 Research environment and limitations	49
4.2 Input from CASE	50
4.3 Input from metaCASE.....	51
4.4 Research problem definition	52
5 Research methodology	53
5.1 Choice and description of methodology	53
5.2 Application of the methodology in this research	56
6 Summary of the thesis	59
6.1 What's in a Relationship?	
On Distinguishing Property Holding and Object Binding.....	59
6.2 MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment	60
6.3 MetaEdit+: CASE Functionality to Support Production, Coordination and Organizational Control and Innovation.....	61
6.4 Application of Repository Technology and Concepts to a MetaCASE Environment.....	62
6.5 A Matrix Editor for a MetaCASE Environment.....	64
6.6 Evaluating Method Engineer Performance: An Error Classification and Preliminary Empirical Study	65
6.7 About the joint articles	66
7 Conclusion	66
7.1 Contribution of the thesis	66
7.2 Directions for further research	68
References.....	70

CHAPTER 2	What's in a Relationship?	
	On Distinguishing Property Holding and Object Binding.....	81
1	Introduction.....	82
2	Background	83
2.1	Theoretical background of relationships.....	83
2.2	MetaCASE	85
2.3	MetaEdit+	85
3	Requirements and starting point.....	87
3.1	MetaCASE: modelling methods, not the real world	87
3.2	Problems with combining binding and property-possession	89
4	Bindings as a solution to relationship problems.....	90
4.1	Beyond the binary relationship: more roles and objects.....	91
4.2	Relationship properties and bindings can exist separately	93
4.3	Roles and relationships can exist separately.....	94
4.4	Implementation efficiency	95
4.5	Graphs and bindings.....	97
4.6	Bindings allow polymorphism of metatypes.....	98
5	Conclusions	100
	References.....	101
	Appendix 1	105
	Postscript	110
CHAPTER 3	MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment.....	113
1	Introduction.....	114
2	Related research	116
2.1	Lack of method integration mechanisms	116
2.2	Insufficient multi-user support	116
2.3	Insufficient support for multiple representation paradigms.....	117
2.4	Lack of method modifiability and evolution.....	117
2.5	Lack of information retrieval and computational facilities	118
2.6	Summary	118
3	The MetaEdit+ environment.....	119
3.1	General architecture	120
3.2	Tool architecture	122
4	Conceptual data model.....	123
4.1	The OPRR model	123
4.2	Extensions in the GOPRR model.....	124
4.3	Example.....	126
5	Method management tools.....	127
5.1	Motivation and purpose of the method management tools	127
5.2	Design principles of method management tool family.....	128
5.3	An example of a method specification	130
6	Discussion and conclusions	130
	Acknowledgments.	132
	References.....	133
	Postscript	137

CHAPTER 4	MetaEdit+: CASE Functionality to Support Production, Coordination and Organizational Control and Innovation.....	138
1	Introduction.....	139
2	Background and related research.....	141
3	The MetaEdit+ environment.....	143
4	MetaEdit+ components and architecture.....	144
4.1	The MetaEngine and its data architecture.....	147
4.2	Tool sets.....	148
5	Model editing tools.....	150
5.1	Diagram Editor	150
5.2	Matrix Editor.....	151
5.3	Table Editor.....	152
6	Model retrieval tools.....	153
6.1	Repository Browsers	153
6.2	Graphical Query Tool	154
6.3	Report Editor.....	155
7	Model linking and annotation tools	156
7.1	Linking Ability: Hypertext editor.....	157
7.2	Debate Browser.....	158
8	Discussion and conclusions	159
	Acknowledgements	161
	References	161
CHAPTER 5	Application of Repository Technology and Concepts to a MetaCASE Environment.....	165
1	Introduction.....	166
2	Background	167
2.1	Related research.....	167
2.2	Requirements for multi-user (meta-)CASE	169
2.3	MetaEdit+.....	170
2.4	ArtBase	173
3	The Repository: A modern vision	174
3.1	Repository system architecture	175
3.2	Repository engine.....	176
3.3	Generic repository tools.....	178
3.4	Tools using the repository	180
4	Locks in ArtBase and MetaEdit+	183
4.1	Concepts	183
4.2	Automatic locking strategies	185
5	Evaluation.....	193
6	Conclusions	194
	Acknowledgements	195
	References	195

CHAPTER 6 A Matrix Editor for a MetaCASE Environment	199
Preface.....	200
1 Introduction.....	201
2 Background	204
2.1 Overview of MetaEdit+	204
2.2 Matrix methods.....	206
3 Functionality.....	209
3.1 Groups	209
3.2 Representational graph type conversions	210
3.3 Axis algorithms.....	212
4 User interface.....	214
4.1 Matrix Editor tool window	214
4.2 Axes	216
4.3 Elements	217
5 Conclusions	218
Acknowledgements	220
References.....	220
Postscript	222
CHAPTER 7 Evaluating Method Engineer Performance: An Error Classification and Preliminary Empirical Study	223
1 Introduction.....	225
2 Background and terminology	226
3 Classification of facets in metamodeling	228
4 Research hypothesis and scoring.....	229
5 Experiment set-up and tools used	231
6 Preliminary analysis of user metamodels.....	232
7 Conclusions	235
References.....	236
Appendix 1	238
APPENDIX 1 GOPRR Description.....	239
1 Background and motivation	241
2 GOPRR concepts	244
3 A model of GOPRR.....	251
4 GOPRR and object orientation.....	258
5 Example metamodels	262
References.....	264
Yhteenveto (Finnish summary).....	267