Specification and Verification of Declarative Open Interaction Models

A Logic-Based Approach
Specification and Verification of Declarative Open Interaction Models

A Logic-Based Approach
Dedicated to Valentina and Giulia, and to Evita and Micol.
Foreword

Many emerging settings, such as business process management systems, clinical guidelines, services composition/choreographies and multi-agent systems, are characterized by a distribution of activities and resources and by exhibiting complex dynamics. This is due to their intrinsic openness, to the autonomy of their stakeholders, and to the unpredictability of the environment in which they are situated. To understand such dynamics and to develop methods and tools apt to accommodate modeling and reasoning about them is a very challenging goal.

All these systems center around the interaction of the involved parties. Interaction, in turn, must be disciplined so as to ensure that the interacting entities comply with external regulations, norms, business rules, and internal policies. Such forces have the effect of constraining the courses of interaction, and can be thus grouped under the umbrella term of business constraint.

Business constraints constitute a form of declarative knowledge: they restrict the set of compliant courses of interaction, without explicitly stating how the interacting entities must behave. Unfortunately, traditional modeling approaches have a closed and procedural nature, and thus require writing down explicitly all the compliant behaviors, producing “spaghetti-like” models which tend to sacrifice flexibility and readability.

In this respect, a shift toward open and declarative modeling abstractions is needed. However, the adoption of declarative open interaction models poses two challenging questions: how can we specify them, and what is their semantics? How is it possible to support their design, execution, verification and analysis?

In this book, Montali answers both of these fundamental questions. A broad survey of the state of the art is given, where Montali puts forward convincing evidence that closed, procedural approaches must be complemented with open and declarative ones. He then presents a great deal of background material on various languages and techniques belonging to different, usually unrelated, domains such as business process management, logic programming, knowledge representation and reasoning and multi-agent systems.
The main theme of the book is the integration and extensions of all these contributions within a computational logic-based framework called CLIMB. The result is a unified, synergic, and comprehensive framework, where non-IT experts can graphically specify interaction models, automatically obtaining a corresponding formal representation and a set of fully automated sound and complete verification facilities.

CLIMB exploits both the declarative advantages of computational logic and its computational power. On the one hand, computational logic defines a declarative and meaningful semantics to open and declarative interaction models. On the other hand, it provides a plethora of effective reasoning techniques that support interaction models prior to, during, and after their execution.

The book contains one of the most clever corpora of ideas centered around the application of computational logic-based techniques for the specification and verification of interaction models. It constitutes a solid and motivating ground for future developments. The technical excellence of the book and the significance of its multifarious contributions will be a valuable asset for researchers of diverse scientific communities, practitioners in emerging application domains, and future generations of doctoral students.

May 2010

Paola Mello
This book contains a revised and extended version of the dissertation the author wrote in the Artificial Intelligence Division of the Department of Electronics, Computer Science and Systems at the University of Bologna, Italy. The dissertation was submitted to the University of Bologna in conformity with the requirements for the degree of Doctor of Philosophy in April 2009. It was honored with the 2009 “Marco Cadoli” prize, awarded by the Italian Association for Logic Programming to the most outstanding theses focused on computational logic and discussed between 2007 and 2009.

Abstract

The advent of distributed and heterogeneous systems has laid the foundation for the birth of new architectural paradigms, in which many separated and autonomous entities collaborate and interact, with the aim of achieving complex strategic goals, impossible to be accomplished on their own. A non-exhaustive list of systems targeted by such paradigms includes business process management, clinical guidelines and careflow protocols, service-oriented computing, and multi-agent systems.

It is largely recognized that engineering these systems requires novel modeling techniques. In particular, many authors are claiming that an open, declarative perspective is needed to complement the closed, procedural nature of the state-of-the-art specification languages, toward flexibility, usability, and verifiability. For example, the ConDec language has been recently proposed to target the declarative and open specification of business processes, overcoming the over-specification and over-constraining issues of classical procedural approaches, which tend to force unnecessary rigidity on the way the systems sub-parts coordinate. On the one hand, the success of such novel modeling languages strongly depends on their usability by non-IT savvy: they must provide an appealing, intuitive graphical front-end. On the other hand, they
must be apt to verification, in order to guarantee the trustworthiness and reliability of the developed model, as well as to ensure that the actual executions of the system effectively comply with it.

The claim of this book is that computational logic is a suitable supporting framework for declarative open interaction models. In particular, the CLIMB (Computational Logic for the verification and Modeling of Business constraints) framework is proposed to address the specification, verification, execution, monitoring, and analysis of interaction models.

After having introduced the main distinctive features of open declarative interaction models and motivated their suitability in different application domains, we propose to adopt an extended version of the ConDec language for their graphical specification. We then show how all the (extended) ConDec constructs can be automatically formalized by using a subset of the SCIFF language. SCIFF is a framework based on computational logic (Abductive Logic Programming in particular), which encompasses a rule-based language with a clear declarative semantics for specifying the interaction, and a family of proof procedures for concretely addressing reasoning and verification. We illustrate how such reasoning techniques can be successfully exploited to provide support and verification capabilities along the whole life cycle of the targeted systems. A number of challenging tasks are addressed, including static verification of properties, composition and interoperability of interaction models, run-time compliance verification, monitoring, and mining.

The investigation is carried out spanning from theoretical aspects, such as proofs of formal properties and comparison with temporal logics, to practical applications, experimental evaluations, case studies, and tools.

The book is organized in four parts.

**Part I: Specification.** Chapter 2 gives a very accessible and precise overview of the application domains. Chapter 3 provides a critical overview of the ConDec language and introduces the framework of propositional linear temporal logic. Chapter 4 presents the CLIMB rule-based language, discusses its declarative semantics, and proves some interesting formal properties. Chapter 5 draws a bridge between the CLIMB and the ConDec languages, by proposing a translation and by running a theoretical investigation about the expressiveness of the two formalisms. Chapter 6 introduces and studies some very significant extensions to ConDec. Chapter 7 concludes this part with a discussion of related work and a summary.

**Part II: Static Verification.** Chapter 8 introduces the problem of interaction model design-time verification. Chapter 9 presents two proof-procedures for reasoning from CLIMB specifications, and formally investigates their properties of termination, soundness, and completeness. Chapter 10 addresses the static verification of ConDec models. Chapter 11 conducts an experimental evaluation of the proposed methods, by comparing their performance with that of model checkers. Chapter 12 discusses related work and gives a summary of this second part.
Part III: Runtime and A Posteriori Verification. Chapter 13 introduces the problem of open declarative interaction model run-time verification, and discusses the application of the proof-procedures to the run-time verification of interacting entities with respect to ConDec models. Chapter 14 presents a reactive form of event calculus, axiomatized on top of SCIFF, which enables monitoring and enacting ConDec models. Chapter 15 presents innovative declarative process mining techniques based on CLIMB, and it describes two implemented tools and their application. Chapter 16 concludes with related work and summary.

Part IV: Conclusions and Future Work.

Acknowledgments

First and foremost, I would like to thank my supervisor, Prof. Paola Mello, for having constantly supported and encouraged my research activity. She guided me through the intriguing world of computational logic with her deep knowledge and open mind. Thanks to Federico Chesani, for having shared with me every working day as a colleague and a friend. His observations and feedback have been fundamental to finalizing my thoughts. Thanks to Paolo Torroni, for having always encouraged me to pursue my research and for having shared with me his strong research attitude. A special thanks goes to Marco Gavanelli, for his constant support on the SCIFF Framework.

I am grateful to Prof. Bob Kowalski, for having first inspired and then appreciated my work. I am greatly indebted to Prof. Wil van der Aalst and Maja Pesic. The insightful and illuminating discussions I had with them during my two visits at TU/e strongly contributed to shaping the ideas presented in this book. Without their work on ConDec and declarative business process management, this book would have never been possible.

Last but not least, I would like to thank all the people who contributed to my education and research. In alphabetical order: Marco Alberti, Matteo Baldoni, Cristina Baroglio, Alessio Bottrighi, Anna Ciampolini, Carlo Giannelli, Evelina Lamma, Michela Milano, Fabrizio Riguzzi, Davide Sottara, Sergio Storari, Paolo Terenziani, Alessandra Toninelli.

My research activity has been partially supported by the FIRB Project TOCAI.IT: Knowledge-Oriented Technologies for Enterprise Aggregation in Internet and by the PRIN 2005 Project Languages for the Specification and Verification of Agents Interaction Protocols.


May 2010

Marco Montali
Acronyms

ALP     Abductive Logic Programming
B2B     Business-To-Business
B2C     Business-To-Consumer
BDD     ordered Binary Decision Diagram
BMK     Basic Medical Knowledge
BP      Business Process
BPM     Business Process Management
BPMN    Business Process Modeling Notation
CEC     Cached Event Calculus
CEP     Complex Event Processing
CG      Clinical Guideline
CHR     Constraint Handling Rules
CLIMB   Computational Logic for the verification and Modeling of Business constraints
CLP     Constraint Logic Programming
EBS     Event-Based System
EC      Event Calculus
FOL     First Order Logic
ILP     Inductive Logic Programming
KB      Knowledge Base
LP      Logic Programming
LTL     propositional Linear Temporal Logic
MAS     Multi-Agent System
MTL     Metric Temporal Logic
NAF     Negation As Failure
QoS     Quality of Service
REC     Reactive Event Calculus
SCIFF   Social Constrained IFF Framework
sciff   SCIFF Proof Procedure
g-sciff g-SCIFF Proof Procedure
SOA     Service Oriented Architecture
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOC</td>
<td>Service Oriented Computing</td>
</tr>
<tr>
<td>TPTL</td>
<td>Timed Propositional Temporal Logic</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>WfMS</td>
<td>Workflow Management System</td>
</tr>
<tr>
<td>WS</td>
<td>Web Service</td>
</tr>
</tbody>
</table>
## Contents

1 **Introduction** .......................................................... 1  
  1.1 Main Contributions of the Book ................................. 3  
    1.1.1 Specification of Open Declarative Interaction Models... 3  
    1.1.2 Static Verification of Interaction Models ............... 4  
    1.1.3 Run-Time Verification, Monitoring and Enactment  
       Facilities .................................................. 4  
    1.1.4 A-Posteriori Verification and Declarative Process Mining 5  
  1.2 Organization of the Book .................................... 6  
    1.2.1 Part I: Specification .................................... 6  
    1.2.2 Part II: Static Verification ............................ 7  
    1.2.3 Part III: Run-Time and A-Posteriori Verification ...... 8  
    1.2.4 Part IV: Conclusion and Future Work ................... 8  
  1.3 Web Site ....................................................... 8  

### Part I Specification

2 **Declarative Open Interaction Models** .......................... 11  
  2.1 An Informal Characterization of Interaction Models ....... 11  
    2.1.1 Interaction Abstractions: Activity, Event, Execution  
       Trace ...................................................... 11  
    2.1.2 Characterization of Time ................................ 13  
    2.1.3 Compliance ............................................ 14  
    2.1.4 Flexibility ........................................... 16  
    2.1.5 Openness ............................................. 19  
  2.2 Business Process Management ...................................... 20  
    2.2.1 Modeling Business Processes ................................ 22  
    2.2.2 Limits of Procedural Business Process Modeling ...... 23  
  2.3 Service Oriented Computing ....................................... 25  
    2.3.1 Service Oriented Architecture ............................ 25  
    2.3.2 Orchestration and Choreography .......................... 26
3 The ConDec Language
3.1 ConDec in a Nutshell
3.2 ConDec Models
3.3 Constraints
  3.3.1 Existence Constraints
  3.3.2 Choice Constraints
  3.3.3 Relation Constraints
  3.3.4 Negation Constraints
  3.3.5 Branching Constraints
3.4 ConDec at Work
  3.4.1 The Order Management Choreography in Natural Language
  3.4.2 The Order Management Choreography as a Contract
  3.4.3 Identification of Activities
  3.4.4 Elicitation of Business Constraints
  3.4.5 Completing the ConDec Model
3.5 Usability of ConDec
3.6 Linear Temporal Logic
  3.6.1 LTL Models
  3.6.2 Syntax of Linear Temporal Logic
  3.6.3 Semantics of Linear Temporal Logic
3.7 Translating ConDec into Linear Temporal Logic
  3.7.1 The Translation Function
  3.7.2 LTL Entailment as a Compliance Evaluator
  3.7.3 Linear Temporal Logic and Finite ConDec Traces
4 The CLIMB Rule-Based Language
4.1 The CLIMB Language in a Nutshell
4.2 The CLIMB Language
  4.2.1 Event Occurrences and Execution Traces
  4.2.2 Constraint Logic Programming
  4.2.3 Expectations
  4.2.4 Integrity Constraints
  4.2.5 The Static Knowledge Base
4.2.6 SCIff-lite and Composite Events .................... 93
4.3 CLIMB Declarative Semantics ......................... 95
  4.3.1 Abduction ........................................ 96
  4.3.2 Abductive Logic Programming ...................... 98
  4.3.3 Formalizing Interaction Models and Their Executions . 99
  4.3.4 SCIff-lite Specifications ......................... 101
  4.3.5 CLIMB Abductive Explanations .................... 102
  4.3.6 On the Formal Definition of Compliance .......... 105
4.4 Formal Properties ..................................... 109
  4.4.1 Equivalence Modulo Compliance ................... 109
  4.4.2 Compositionality Modulo Compliance ............... 110
  4.4.3 Replaceability of Rules .......................... 113

5 Translating ConDec into CLIMB .......................... 115
  5.1 Translation of a ConDec Model ....................... 116
  5.2 Translation of Events ................................ 117
  5.3 Embedding a Qualitative Characterization of Time into a
    Quantitative Setting ................................. 117
    5.3.1 Temporal Contiguity ............................ 117
    5.3.2 Compact Execution Traces ....................... 119
  5.4 Translation of ConDec Constraints ................... 121
    5.4.1 Translation of Existence Constraints .......... 121
    5.4.2 Translation of Choice Constraints ............. 123
    5.4.3 Translation of Relation and Negation Constraints . 125
  5.5 Dealing with Branching ConDec Constraints ........... 128
  5.6 Translation of a ConDec Choreography ............... 130
  5.7 Equivalence of ConDec Constraints .................. 130
  5.8 Soundness of the Translation ....................... 132
    5.8.1 Trace Mapping .................................. 133
    5.8.2 Behavioral Equivalence ......................... 134
    5.8.3 Soundness ...................................... 134
    5.8.4 On the Expressiveness of SCIff ................. 137

6 Extending ConDec .................................. 139
  6.1 Metric Constraints .................................. 139
    6.1.1 Temporal Contiguity in a Quantitative Setting ... 140
    6.1.2 Quantitative Formalization of Chain Constraints .. 141
    6.1.3 Init Constraint ................................ 142
    6.1.4 Extending ConDec with Quantitative Temporal
      Constraints ........................................ 142
  6.2 Data Aware Aspects .................................. 145
    6.2.1 Examples of Data Aware Business Constraints ..... 145
    6.2.2 The MXML Meta Model ............................ 147
    6.2.3 The Life Cycle of Non Atomic Activities in ConDec . 148
  6.3 Modeling Data in ConDec ............................ 149
XVIII Contents

6.3.1 Representing Non Atomic Activities and Data .............. 149
6.3.2 Modeling Data Aware Conditions .......................... 150
6.3.3 Modeling the Submit&Review Process Fragment .......... 154

6.4 Formalizing Data Aware Aspects in CLIMB .................... 155
6.4.1 Formalizing Data and Data Aware Conditions .............. 155
6.4.2 Formalizing the Effect of Roles .......................... 156
6.4.3 Formalizing the Activity Life Cycle ....................... 159

7 Related Work and Summary .................................... 163
7.1 Related Work .............................................. 163
7.1.1 Business Process Management ......................... 163
7.1.2 Clinical Guidelines .................................... 166
7.1.3 Service Oriented Computing .............................. 169
7.1.4 Multi-Agent Systems .................................... 170
7.2 Summary of the Part ....................................... 172

Part II Static Verification

8 Static Verification of Declarative Open Interaction Models 177
8.1 Desiderata for Static Verification Technologies ............. 177
8.2 Verification of a Single Model vs. a Composition of Models . 179
8.3 Static Verification of Properties ............................ 180
8.3.1 Existential vs. Universal Properties ..................... 180
8.3.2 General vs. Particular Properties ....................... 181
8.3.3 On the Safety-Liveness Classification ..................... 182
8.3.4 A ConDec Example ..................................... 184
8.4 A-Priori Compliance Verification ............................. 186
8.5 Composing ConDec Models .................................. 187
8.5.1 Compatibility between Local Models ..................... 188
8.5.2 Augmenting ConDec with Roles and Participants ........ 190
8.5.3 From Openness to Semi-Openness ........................ 192
8.6 Conformance with a Choreography ........................... 196

9 Proof Procedures ............................................. 201
9.1 The SCIFF Proof Procedure .................................. 202
9.1.1 Fulfilled, Violated and Pending Expectations ........... 202
9.1.2 Data Structures and Proof Tree .......................... 203
9.1.3 Transitions ............................................. 205
9.2 Formal Properties of the SCIFF Proof Procedure ............ 214
9.2.1 Soundness ............................................. 214
9.2.2 Completeness ......................................... 215
9.2.3 Termination ............................................ 216
9.2.4 ConDec and Termination of the SCIFF Proof Procedure .... 218
9.3 The g-SCIFF Proof Procedure ................................ 218
9.3.1 Generation of Intensional Traces ........................................... 219
9.3.2 Data Structures Revisited .................................................. 220
9.3.3 Transitions Revisited ......................................................... 221
9.3.4 Linking the Proof Procedures ............................................... 222
9.4 Formal Properties of the g-SCIFF Proof Procedure .................... 223
  9.4.1 Soundness ................................................................. 223
  9.4.2 Completeness Modulo Trace Generation .............................. 223
  9.4.3 Termination ............................................................... 225
9.5 Implementation of the Proof Procedures .................................. 227

10 Static Verification of ConDec Models with g-SCIFF ............... 229
  10.1 Existential and Universal Entailment in CLIMB .................... 229
    10.1.1 Specification of Properties with ConDec ...................... 229
    10.1.2 Formalizing Existential and Universal Entailment .......... 231
  10.2 Verification of Existential Properties with g-SCIFF .......... 232
    10.2.1 Conflict Detection with g-SCIFF ............................... 233
    10.2.2 Existential Entailment with g-SCIFF ........................... 233
  10.3 Verification of Universal Properties with g-SCIFF ............ 234
    10.3.1 Complementation of Integrity Constraints ................... 234
    10.3.2 Reducing Universal to Existential Entailment .............. 236
  10.4 ConDec Loops and Termination of g-SCIFF ......................... 239
    10.4.1 Constraints Reformulation ....................................... 240
    10.4.2 Looping ConDec Models ........................................... 241
10.5 A Preprocessing Procedure ............................................... 246

11 Experimental Evaluation ..................................................... 251
  11.1 Verification Procedure with g-SCIFF .................................. 251
  11.2 Scalability of the g-SCIFF Proof Procedure ....................... 253
    11.2.1 The Branching Responses Benchmark ............................ 253
    11.2.2 The Alternate Responses Benchmark ............................. 255
    11.2.3 The Chain Responses Benchmark ................................ 258
  11.3 Static Verification of ConDec Models as Model Checking ....... 261
    11.3.1 Existential and Universal Entailment of ConDec
        Properties in LTL ..................................................... 261
    11.3.2 ConDec Properties Verification as Satisfiability and
        Validity Problems ................................................... 262
    11.3.3 Model Checking ..................................................... 264
    11.3.4 Reduction of Validity and Satisfiability to Model
        Checking .......................................................... 266
    11.3.5 Verification Procedure by Model Checking .................. 268
  11.4 Comparative Evaluation ................................................ 269
    11.4.1 The Order&Payment Benchmarks .................................. 269
    11.4.2 Discussion .......................................................... 270
12 Related Work and Summary ............................................ 277
  12.1 Related Work .................................................. 277
    12.1.1 Verification of Properties ............................... 277
    12.1.2 A-Priori Compliance Verification ...................... 281
    12.1.3 Model Composition ........................................ 283
    12.1.4 Interoperability and Choreography Conformance ...... 284
  12.2 Summary of the Part ........................................... 285

Part III Run-Time and A-Posteriori Verification

13 Run-Time Verification .............................................. 289
  13.1 The Run-Time Verification Task ............................... 290
  13.2 SCIFF-Based Run-Time Verification .......................... 291
    13.2.1 Reactive Behavior of the SCIFF Proof Procedure ...... 292
    13.2.2 Open Derivations ....................................... 292
    13.2.3 Semi-Open Reasoning ................................... 295
  13.3 The SOCS-SI Tool ............................................. 298
  13.4 Speculative Run-Time Verification .......................... 300
    13.4.1 The Need for Speculative Reasoning .................... 300
    13.4.2 Speculative Verification with the g-SCIFF Proof Procedure ........................................ 301
    13.4.3 Interleaving the Proof Procedures ...................... 303

14 Monitoring and Enactment with Reactive Event Calculus .... 305
  14.1 Monitoring Issues and SCIFF .................................. 305
  14.2 Event Calculus ................................................. 307
    14.2.1 The Event Calculus Ontology ............................ 308
    14.2.2 Domain-Dependent vs. Domain-Independent Axioms ... 309
    14.2.3 Reasoning with Event Calculus .......................... 310
  14.3 The Reactive Event Calculus .................................. 312
    14.3.1 Axiomatization of the Reactive Event Calculus ...... 312
    14.3.2 Monitoring an Event Calculus Specification with the SCIFF Proof Procedure .................................. 315
  14.4 REC Illustrated: A Personnel Monitoring Facility ....... 316
    14.4.1 Formalizing the Personnel Monitoring Facility in REC 316
    14.4.2 Monitoring a Concrete Instance ........................ 318
    14.4.3 The Irrevocability Issue ................................ 319
  14.5 Formal Properties of REC ...................................... 320
    14.5.1 Soundness, Completeness, Termination ................. 320
    14.5.2 Irrevocability of REC .................................... 321
  14.6 Monitoring Optional ConDec Constraints with REC ........ 324
    14.6.1 Representing ConDec Optional Constraints in the Event Calculus ........................................ 325
    14.6.2 Identification and Reification of Violations .......... 328
14.6.3 Compensating Violations ............................................. 331
14.6.4 Monitoring Example .................................................. 332
14.7 Enactment of ConDec Models ........................................ 335
  14.7.1 Showing Temporarily Violated Constraints ................. 336
  14.7.2 Blocking Non Executable Activities ......................... 336
  14.7.3 Termination of the Execution ................................. 339
14.8 jREC: Embedding REC Inside a JAVA-Based Tool .......... 340

15 Declarative Process Mining ........................................... 343
  15.1 Declarative Process Mining with ProM, SCIFF Checker and
      DecMiner .................................................................. 345
  15.2 The SCIFF Checker ProM Plug-in ................................. 346
      15.2.1 CLIMB Textual Business Constraints .................... 347
      15.2.2 A Methodology for Building Rules ....................... 348
      15.2.3 Specification of Conditions ............................... 349
      15.2.4 Trace Analysis with Logic Programming ............... 350
      15.2.5 Embedding SCIFF Checker in ProM ....................... 351
  15.3 Case Studies ............................................................. 353
      15.3.1 The Think3 Case Study .................................... 354
      15.3.2 Conformance Verification of a Cervical Cancer
          Screening Process ............................................ 357
      15.3.3 Quality Assessment in Large Wastewater Treatment
          Plans .................................................................. 358
  15.4 The DecMiner ProM Plug-in ...................................... 361
      15.4.1 Inductive Logic Programming for Declarative Process
          Discovery .......................................................... 361
      15.4.2 Embedding DecMiner into the ProM Framework ....... 362
  15.5 The Checking–Discovery Cycle .................................. 364

16 Related Work and Summary ............................................. 367
  16.1 Related Work ............................................................. 367
      16.1.1 Run-Time Verification and Monitoring .................. 367
      16.1.2 Enactment ....................................................... 372
      16.1.3 Process Mining ............................................... 373
  16.2 Summary of the Part .................................................. 375

Part IV Conclusion and Future Work

17 Conclusion and Future Work ........................................... 379
  17.1 Conclusion .............................................................. 379
  17.2 Future Work ............................................................. 380
      17.2.1 Termination of Static Verification and ConDec
          Extensibility ..................................................... 380
      17.2.2 Reactive Event Calculus and Operational Support .... 381
XXII Contents

17.2.3 Integration of Regulative and Constitutive Rules ........ 381
17.2.4 Development of an Editor and Enactment Prototype ... 382
17.2.5 Service Contracting and Discovery in the Semantic Web 383
17.2.6 Integration of Declarative and Procedural Models ....... 383

References ..................................................... 385

Index .......................................................... 403