## Contents

### Part I Introduction to Internetware

1 **Internetware: A Shift of Software Paradigm**  
   1.1 Introduction ................................. 3  
   1.2 The Internetware Paradigm. ................. 4  
      1.2.1 The Need of a Paradigm Shift. ........ 4  
      1.2.2 Methodological Reflections ............ 5  
      1.2.3 Architecting Principles for Internetware 7  
      1.2.4 Our Roadmap and Explorations ............ 8  
   1.3 Open Coordination Model ...................... 8  
      1.3.1 Coordination Model for Internetware .... 8  
      1.3.2 A Software Architecture Based Approach 10  
   1.4 Environment-Driven Model .................... 11  
      1.4.1 Structure and Dynamics ................. 11  
      1.4.2 Framework and Techniques .............. 13  
   1.5 Dependability Assurance Framework with Trust-Management 14  
      1.5.1 A Trust-Based Approach ............... 14  
      1.5.2 A Dependability Assurance Framework with Trust Management 15  
   1.6 Conclusion ..................................... 15  
References ......................................... 16

2 **Technical Framework for Internetware: An Architecture Centric Approach** 19  
   2.1 Introduction .................................. 19  
   2.2 Software Model of Internetware .............. 21  
      2.2.1 Basic Component Model ................. 22  
      2.2.2 Open Collaboration Model ............... 22  
      2.2.3 Context Driven Model ................. 23  
      2.2.4 Intelligent Trustworthy Model ......... 23  
   2.3 Middleware for Internetware .................. 23  
      2.3.1 The Implementation Model of Basic Internetware Entities 24  
      2.3.2 The Implementation Model of On-Demand Collaborations 24  
      2.3.3 The Autonomic Management of Middleware 25  
      2.3.4 Componentization of Middleware ........ 25
2.4 Engineering Approach for Internetware
2.4.1 Software Architecture in the Whole Lifecycle
2.4.2 Development of Internetware Basic Entities and Their Collaborations
2.4.3 Domain Modeling for Internetware
2.5 Conclusion
References

Part II  Internetware Software Model

3 On Environment-Driven Software Model for Internetware
3.1 Introduction
3.2 An Approach to the Environment-Driven Model
3.2.1 Overall Interaction Patterns of Environment-Driven Applications
3.2.2 A Structuring Model for Environment-Driven Applications
3.2.3 Model and System Characteristics
3.3 Environment Modeling and Enabling Techniques
3.3.1 Context Processing Framework
3.3.2 Ontology-Based Modeling of Dynamic Context
3.3.3 Logic Distance-Based Context Retrieval
3.3.4 Blurring Degree-Based Privacy Protection
3.3.5 Negotiation-Based Context Information Selection
3.4 Goal-Driven Adaptation and Rearon Ontologies
3.4.1 Rearon Ontologies
3.4.2 Overall Interaction Patterns of Environment-Driven Applications
3.5 Prototypes and Experiments
3.5.1 Artemis-ARC
3.5.2 Artemis-MAC
3.5.3 Artemis-FollowMeLite
3.5.4 Artemis-FollowMeAgent
3.5.5 Environment-Driven Applications
3.6 Related Work
3.7 Conclusion
References

4 On Self-adaptation Model for Internetware
4.1 Introduction
4.2 Approach Overview
4.3 Key Techniques
4.3.1 Basic SA Model
4.3.2 SA Reflection
4.3.3 SA Dynamism
4.3.4 SA Reasoning
6.5 Manipulation of Recovered Software Architecture at Runtime ........................................ 131
6.5.1 Manipulation via Reflection ......................... 131
6.5.2 Programming Model ................................. 132
6.5.3 Graphical Tool ........................................ 132
6.6 Performance Evaluation ................................. 134
6.7 Related Work ............................................. 135
6.8 Conclusion and Future Work ............................. 136
References .................................................... 137

7 Supporting Runtime Software Architecture:
A Bidirectional-Transformation-Based Approach ........ 139
7.1 Introduction .............................................. 140
7.2 Runtime Software Architecture ......................... 141
7.2.1 An Illustrative Example ............................. 141
7.2.2 A Formal Description of Runtime Software Architecture ........................................ 142
7.3 Maintaining Causal Connections by Architecture-System Synchronization ......................... 144
7.3.1 The Four Properties .................................. 144
7.3.2 The Challenges ....................................... 144
7.4 Architecture-System Synchronization Based on Bi-transformation .................................... 145
7.4.1 Enabling Techniques ................................. 145
7.4.2 The Synchronization Algorithm ..................... 146
7.4.3 Assumptions ......................................... 149
7.4.4 Discussion About the Algorithm and the Properties ........................................ 149
7.5 Generating Synchronizers for Legacy Systems .......... 150
7.5.1 Implementing the Generic Synchronization Engine ........................................ 151
7.5.2 Generating Specific XMI Parsers and System Adapters ........................................ 151
7.6 Case Studies .............................................. 152
7.6.1 C2-JOnAS ............................................ 152
7.6.2 Client/Server-JOnAS ................................. 154
7.6.3 Other Case Studies .................................. 155
7.6.4 Summary and Discussion ........................... 155
7.7 Related Work ............................................. 158
7.8 Conclusion ................................................ 159
References .................................................... 159

8 Low-Disruptive Dynamic Updating of Internetware Applications ........................................ 163
8.1 Introduction .............................................. 163
8.2 Dynamic Updating: Eager Versus Lazy .................. 165
8.3 Javelus: Overview ....................................... 166
8.4 Dynamic Patch Generation .............................. 167
8.4.1 Identifying Changed Classes ......................... 167
8.4.2 Default Transformation and Custom Transformers ........................................ 168
9.8 Related Work ................................................................. 205
  9.8.1 Engineering Self-adaptive System ...................... 205
  9.8.2 Runtime Monitoring of Web Services ............. 206
  9.8.3 Parametric Properties Monitoring 
for OO Systems ..................................................... 207
9.9 Conclusion ................................................................. 207
References ................................................................. 208

10 Runtime Detection of the Concurrency Property 
in Asynchronous Pervasive Computing Environments ........ 211
10.1 Introduction ................................................................. 212
10.2 Property Detection for Pervasive Context ............. 213
  10.2.1 Asynchronous Message Passing 
and Logical Time ..................................................... 213
  10.2.2 Consistent Global State (CGS) 
and the Lattice Structure Among CGSs .......... 213
  10.2.3 Specification of Contextual Properties .......... 214
10.3 Concurrent Activity Detection in Asynchronous 
Pervasive Computing Environments ..................... 215
  10.3.1 \textit{Def} (\phi) and Concurrent Contextual 
Activities ............................................................... 215
  10.3.2 Design of the CADA Algorithm .................... 218
  10.3.3 Discussions ......................................................... 219
10.4 Performance Analysis ................................................. 220
  10.4.1 The Causes of Faults ........................................ 221
  10.4.2 Concurrency Among Sensed Activities .......... 221
  10.4.3 Sensed and Physical Activities .................... 223
  10.4.4 Discussions ......................................................... 224
10.5 Experimental Evaluation ........................................... 224
  10.5.1 Implementation .................................................... 224
  10.5.2 Experiment Setup .............................................. 225
  10.5.3 Effects of Tuning the Update Interval .......... 225
  10.5.4 Effects of Tuning the Message Delay .......... 226
  10.5.5 Effects of Tuning the Duration 
of Contextual Activities ........................................ 227
  10.5.6 Effects of Tuning the Number 
of Non-checker Processes .................................. 227
  10.5.7 Lessons Learned ............................................... 227
10.6 Related Work ............................................................... 228
10.7 Conclusion ................................................................. 228
References ................................................................. 229

Part IV Internetware Engineering Approach

11 A Software Architecture Centric Engineering Approach 
for Internetware ............................................................... 233
  11.1 Introduction ............................................................. 233
  11.2 Overview of ABC Methodology ......................... 236
  11.3 Feature Oriented Requirement Modeling 
for Internetware ..................................................... 239
### 13.5 Three Verification Criteria
- 13.5.3 Three Verification Criteria

### 13.6 Design of High-Level Software Architecture
- 13.6.1 An Overview
- 13.6.2 Resource Container Identification
- 13.6.3 Component Seed Creation
- 13.6.4 Component Construction
- 13.6.5 Interaction Analysis

### 13.7 Related Work

### 13.8 Conclusions and Future Work

### Appendix 1

### Appendix 2

### Appendix 3

### References

---

### 14 rcos: A Refinement Calculus of Internetware Systems

### 14.1 Introduction

### 14.2 Semantic Basis
- 14.2.1 Programs as Designs
- 14.2.2 Refinement of Designs

### 14.3 Syntax of rcos
- 14.3.1 Commands
- 14.3.2 Expressions

### 14.4 Semantics
- 14.4.1 Structure, Value and Object
- 14.4.2 Static Semantics
- 14.4.3 Dynamic Variables
- 14.4.4 Dynamic States
- 14.4.5 Evaluation of Expressions
- 14.4.6 Semantics of Commands
- 14.4.7 Semantics of a Program

### 14.5 Object-Oriented Refinement
- 14.5.1 Object System Refinement
- 14.5.2 Structure Refinement

### 14.6 Refinement Rules

### 14.7 Conclusions
- 14.7.1 Related Work
- 14.7.2 Support UML-like Software Development
- 14.7.3 Limitation and Future Work

### References

---

### Part V Experiments and Practices of Internetware

### 15 Refactoring Android Java Code for On-Demand Computation Offloading

### 15.1 Introduction

### 15.2 Design Pattern for Computation Offloading
- 15.2.1 The Source Structure and the Target Structure
- 15.2.2 Refactoring Steps Overview
- 15.2.3 An Illustrative Example
15.3 Implementation of DPartner
   15.3.1 Detect Movable Classes
   15.3.2 Generate Proxies
   15.3.3 Transform App Classes
   15.3.4 Cluster App Classes
   15.3.5 Determine the Computations to Be Offloaded
   15.3.6 Offload Computations at Runtime
15.4 Evaluation
   15.4.1 Experiment Setup
   15.4.2 Performance of Refactoring
   15.4.3 Comparison of App Performance
   15.4.4 Comparison of App Power Consumption
   15.4.5 The Effect of On-Demand Offloading
   15.4.6 Experiments on 3G Network
15.5 Related Work
15.6 Conclusion and Future Work
References

16 Towards Architecture-Based Management of Platforms in Cloud
   16.1 Introduction
   16.2 Motivating Example
   16.3 An Architecture-Based Model for Platform Management in Cloud
      16.3.1 Approach Overview
      16.3.2 The Architecture-Based Meta-Model
      16.3.3 Runtime Changes
   16.4 Implementations of the Runtime Architecture-Based Model
   16.5 Evaluation
      16.5.1 Anti-pattern Detection
      16.5.2 VM States Checking
   16.6 Related Work
   16.7 Conclusion and Future Work
References

17 Golden Age: On Multi-source Software Update Propagation in Pervasive Networking Environments
   17.1 Introduction
   17.2 System Model
   17.3 Update Propagation Strategies
      17.3.1 Random Spread
      17.3.2 Youngest Age
      17.3.3 Golden Age
   17.4 Performance Analysis
      17.4.1 Notations and Assumptions
      17.4.2 Number of Updated Nodes in the System
      17.4.3 Principles for Choosing the Golden Age
Internetware
A New Software Paradigm for Internet Computing
Mei, H.; Lü, J.
2016, XXVIII, 442 p. 191 illus., 117 illus. in color., Hardcover
ISBN: 978-981-10-2545-7