Modern Industrial Automation
Software Design
Principles and Real-World Applications

Lingfeng Wang
Kay Chen Tan

IEEE PRESS
WILEY-INTERSCIENCE
A JOHN WILEY & SONS, INC., PUBLICATION
Contents

Preface
Acknowledgments
Acronyms

Part I  Design Principles of Modern Industrial Automation Systems

1  Introduction
   1.1  Developmental Trends  2
   1.2  Classifications and Existing Products  3
   1.3  Functionality of Industrial Automation Systems  5
   1.4  About the Book  7

2  Virtual Instrumentation
   2.1  Introduction  9
   2.2  Characteristics of VXI Instruments  13
   2.3  VXI Plug&Play (VPP) Specification  14
   2.4  Virtual Instrument Software Architecture (VISA)  16
2.4.1 VISA model structure 17
2.4.2 VISA characteristics 18
2.5 Programming platforms 19
  2.5.1 Textual programming 20
  2.5.2 Visual programming 20
  2.5.3 Graphical programming 21
2.6 Liquefied Petroleum Gas Network (PLPGN) Monitoring 23
  2.6.1 Overall structure design 24
2.7 Hardware and Software Design 26
  2.7.1 Development requirements 26
  2.7.2 Development environment 27
  2.7.3 Configurations of system hardware and software 27
2.8 Summary 29

3 Component-Based Measurement Systems 31
  3.1 Introduction 31
  3.2 Component Technology 32
  3.3 Component-Based Industrial Automation Software 35
  3.4 Writing Component 36
  3.5 Case Study 1 36
  3.6 Case Study 2 38
    3.6.1 Definition of base class of instruments 39
    3.6.2 UI base class of VIs 40
  3.7 Summary 41

4 Object-Oriented Software Engineering 43
  4.1 Software Development Models 44
  4.2 Object Orientation 48
    4.2.1 OOA/OOD 48
    4.2.2 Advantages 51

5 Graphical User Interface Design 53

6 Database Management 59
  6.1 Database Systems 60
  6.2 Relational Database 61
CONTENTS vii

6.3 Structured Query Language (SQL) 64
6.4 Open Database Connectivity (ODBC) 66

7 Software Testing 69
7.1 Software and Industrial Automation 69
7.2 Software Testing Strategies 71
  7.2.1 Black-box testing 72
  7.2.2 White-box testing 73
7.3 Software Testing Processes and Steps 73
  7.3.1 Unit testing 75
  7.3.2 Integration testing 76
  7.3.3 Verification testing 78
  7.3.4 System testing 78
  7.3.5 Validation 79
7.4 Software Performance Testing 79
  7.4.1 Availability testing 80
  7.4.2 Reliability testing 81
  7.4.3 Survivability testing 81
  7.4.4 Flexibility testing 81
  7.4.5 Stress testing 82
  7.4.6 Security testing 82
  7.4.7 Usability testing 82
  7.4.8 Maintainability testing 83
7.5 Software Maintenance 84
7.6 Summary 85

Part II Real-World Applications

8 Overview 91

9 An Object-Oriented Reconfigurable Software 93
  9.1 Introduction 94
    9.1.1 Evolution of reconfigurable software 94
  9.2 Design Requirements, Development Environments, and Methodologies 105
    9.2.1 Design requirements 105
    9.2.2 Development environments 106
    9.2.3 Development methodologies 107
9.3 IMC System Structure and Software Design 108
  9.3.1 Overall structure of IMC systems 108
  9.3.2 Configuration-based IMC software 111
  9.3.3 Reconfigurable IMC software design 112
  9.3.4 Development tool selection 113
  9.3.5 Object-oriented methodology 115
  9.3.6 Windows programming 118
  9.3.7 Database technologies 118
  9.3.8 Relational database model 119
  9.3.9 Database management system (DBMS) 119
  9.3.10 Database application 120
  9.3.11 Delphi database functionality 122
  9.4 RSFIMC Architecture 122
    9.4.1 Data acquisition module 124
    9.4.2 Data processing module 124
    9.4.3 Data browsing module 125
  9.5 RSFIMC Functions 126
    9.5.1 User configuration 126
    9.5.2 Running status indications 133
    9.5.3 Alarm management 134
    9.5.4 Data exchange 135
    9.5.5 Visual database query 140
    9.5.6 Remote communication 142
  9.6 Summary 144

10 Flexible Measurement Point Management 151
  10.1 Introduction 152
  10.2 System Architecture 153
    10.2.1 Overall architecture 154
    10.2.2 Interfaces with other modules 157
  10.3 Development Platform and Environment 157
  10.4 Measurement Point Management 158
    10.4.1 MP configuration 158
    10.4.2 Task configuration 159
    10.4.3 Dynamic configuration of MPs and tasks 160
    10.4.4 System running 161
  10.5 An Illustrative Example on a Serial Port Driver 167
    10.5.1 Serial port hardware driver 168
10.5.2 Serial port system driver 170
10.5.3 DIT maintenance for serial port system driver 171
10.5.4 Hardware simulation terminal 172
10.6 Summary 172

11 A Blending System Using Multithreaded Programming 179
11.1 Introduction 179
11.2 Overall Blending System Configuration 181
  11.2.1 Hardware configuration 181
  11.2.2 Software configuration 183
  11.2.3 Multithread-based communication 183
11.3 The Overall Software Design 185
  11.3.1 Design requirements 186
  11.3.2 Software structure 188
  11.3.3 VxD 189
  11.3.4 Front-end software 189
  11.3.5 Device management module 190
  11.3.6 User management 190
  11.3.7 Database management 190
11.4 Field Experience and Summary 190
  11.4.1 Field experience 191
  11.4.2 Summary 191

12 A Flexible Automatic Test System for Rotating Turbine Machinery 197
12.1 Introduction 198
12.2 Design Goals of FATSFTM 199
12.3 Design Strategies of FATSFTM 201
  12.3.1 Hardware design strategy 201
  12.3.2 Software design strategy 202
12.4 Test Software Development Process 206
  12.4.1 Requirements capture 207
  12.4.2 Analysis 207
  12.4.3 Design 212
  12.4.4 Programming 219
  12.4.5 Testing 220
12.5 Function of FATSFTM 221
  12.5.1 Initialization and self-examination 221
CONTENTS

13.6.4 Design of Web server CGI application 282
13.7 Detailed System Design and Implementation 282
  13.7.1 Implementation of DAQ module 282
  13.7.2 Implementation of data management module 285
  13.7.3 Communication module 287
  13.7.4 Multitasking coordination 291
  13.7.5 Implementation of Web server 293
13.8 Field Experience 295
13.9 Summary 298

14 Epilog 303
  14.1 Middleware 303
  14.2 Unified Modeling Language (UML) 304
  14.3 Agent-based software development 305
  14.4 Agile methodologies 308
  14.5 Summary 309

Index 310