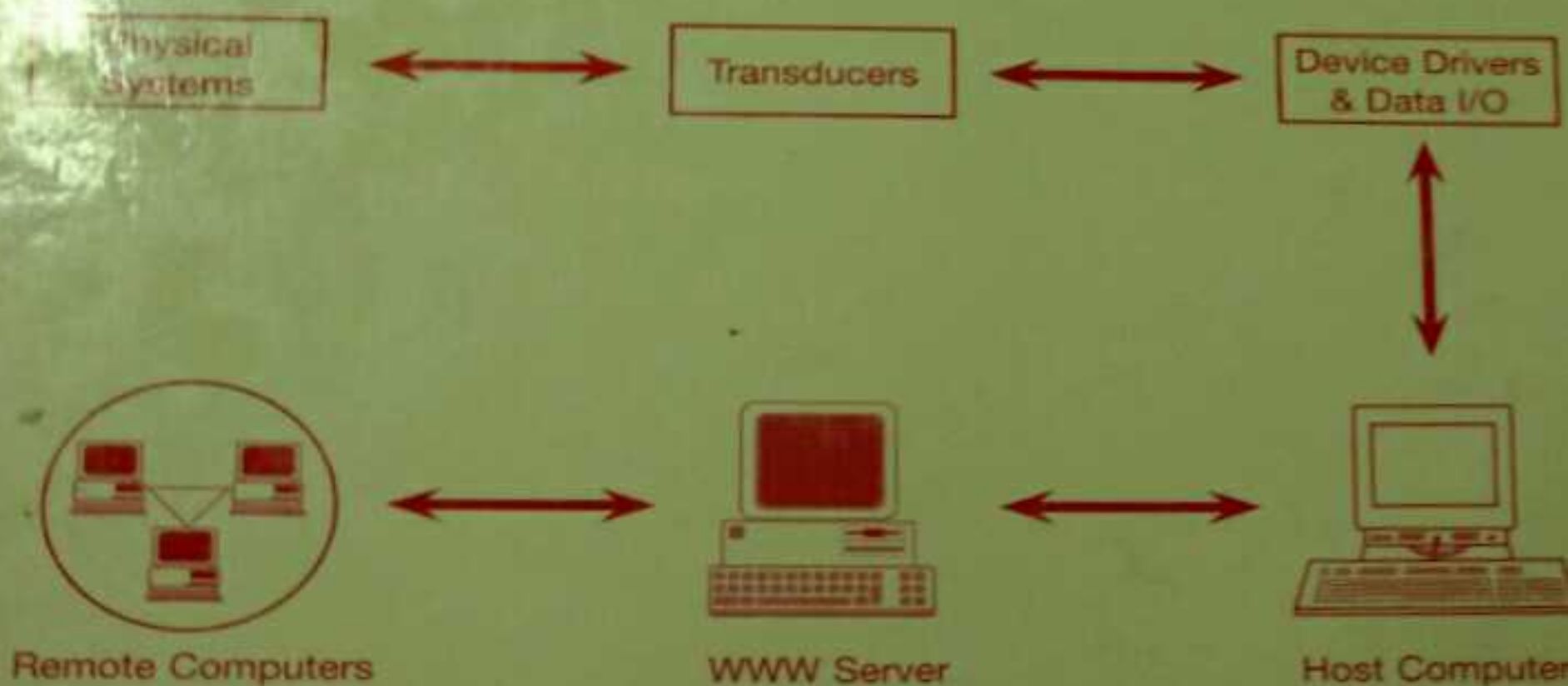


# Modern Industrial Automation Software Design

Principles and Real-World Examples

Jingfeng Wang and Kay Chen Tan




# *Modern Industrial Automation Software Design*


*Principles and Real-World Applications*

**Lingfeng Wang**

**Kay Chen Tan**

147  
ST. MARY'S LIBRARY  
Department of Computer Science  
UNIVERSITY OF KARACHI  
2-7-09

 **IEEE**  
IEEE PRESS

 **WILEY-  
INTERSCIENCE**

A JOHN WILEY & SONS, INC., PUBLICATION

# Contents

<i>Preface</i>	<i>ix</i>
<i>Acknowledgments</i>	<i>xiii</i>
<i>Acronyms</i>	<i>xv</i>

## *Part I Design Principles of Modern Industrial Automation Systems*

<i>1 Introduction</i>	<i>1</i>
1.1 <i>Developmental Trends</i>	<i>2</i>
1.2 <i>Classifications and Existing Products</i>	<i>3</i>
1.3 <i>Functionality of Industrial Automation Systems</i>	<i>5</i>
1.4 <i>About the Book</i>	<i>7</i>
<i>2 Virtual Instrumentation</i>	<i>9</i>
2.1 <i>Introduction</i>	<i>9</i>
2.2 <i>Characteristics of VXI Instruments</i>	<i>13</i>
2.3 <i>VXI Plug&amp;Play (VPP) Specification</i>	<i>14</i>
2.4 <i>Virtual Instrument Software Architecture (VISA)</i>	<i>16</i>



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

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	<i>IMC System Structure and Software Design</i>	108
9.3.1	<i>Overall structure of IMC systems</i>	108
9.3.2	<i>Configuration-based IMC software</i>	111
9.3.3	<i>Reconfigurable IMC software design</i>	112
9.3.4	<i>Development tool selection</i>	113
9.3.5	<i>Object-oriented methodology</i>	115
9.3.6	<i>Windows programming</i>	118
9.3.7	<i>Database technologies</i>	118
9.3.8	<i>Relational database model</i>	119
9.3.9	<i>Database management system (DBMS)</i>	119
9.3.10	<i>Database application</i>	120
9.3.11	<i>Delphi database functionality</i>	122
9.4	<i>RSFIMC Architecture</i>	122
9.4.1	<i>Data acquisition module</i>	124
9.4.2	<i>Data processing module</i>	124
9.4.3	<i>Data browsing module</i>	125
9.5	<i>RSFIMC Functions</i>	126
9.5.1	<i>User configuration</i>	126
9.5.2	<i>Running status indications</i>	133
9.5.3	<i>Alarm management</i>	134
9.5.4	<i>Data exchange</i>	135
9.5.5	<i>Visual database query</i>	140
9.5.6	<i>Remote communication</i>	142
9.6	<i>Summary</i>	144
10	<i>Flexible Measurement Point Management</i>	151
10.1	<i>Introduction</i>	152
10.2	<i>System Architecture</i>	153
10.2.1	<i>Overall architecture</i>	154
10.2.2	<i>Interfaces with other modules</i>	157
10.3	<i>Development Platform and Environment</i>	157
10.4	<i>Measurement Point Management</i>	158
10.4.1	<i>MP configuration</i>	158
10.4.2	<i>Task configuration</i>	159
10.4.3	<i>Dynamic configuration of MPs and tasks</i>	160
10.4.4	<i>System running</i>	161
10.5	<i>An Illustrative Example on a Serial Port Driver</i>	167
10.5.1	<i>Serial port hardware driver</i>	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

12.5.2	Data acquisition	222
12.5.3	User configuration	222
12.5.4	Running status indication and real-time/historical data analysis	223
12.5.5	Alarm management and post-fault diagnosis	224
12.5.6	Remote test	227
12.5.7	Other system functions	228
12.6	Implementation and Field Experience	229
12.6.1	On-site implementation and field experience	229
12.6.2	System benefits	230
12.7	Summary	232
13	An Internet-Based Online Real-Time Condition Monitoring System	239
13.1	Introduction	239
13.2	Problem Description	241
13.2.1	Field data acquisition devices	241
13.2.2	Field data acquisition workstation	242
13.2.3	System servers	243
13.2.4	Remote browsers	243
13.3	Requirements Capture and Elicitation	244
13.3.1	Data acquisition workstation software	245
13.3.2	Analysis (diagnosis) and management workstation software	245
13.4	Analysis	246
13.4.1	Data-flow model	246
13.4.2	Entity-relationship model	249
13.4.3	Event-response model	250
13.5	Transition to Design	251
13.5.1	Choice of development strategies	252
13.5.2	Choice of development environment and programming tool	254
13.6	Overall Design	259
13.6.1	Database design	260
13.6.2	Overall design of DAQ workstation software	263
13.6.3	Overall design of the A&M workstation software	279



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