Integration of Operational Tasks in Chemical Plants

This thesis is presented for the degree of
Doctor of Philosophy

by

Magid Nikraz

BCM, The University of Western Australia, 2003
MEng, Curtin University of Technology, 2004

School of Electrical, Energy and Process Engineering
Murdoch University, 2007
Declaration

I declare that this thesis is my own account of my research and contains as its main content work which has not previously been submitted for a degree at any tertiary education institution.

..........................................

Magid Nikraz
Abstract

The overall, coordinated management of different operational tasks in a chemical plant can improve operational efficiency. These operational tasks can be hierarchically categorised, from the lowest to highest level, as: data acquisition; regulatory control; monitoring; data reconciliation; fault detection and diagnosis; supervisory control; scheduling; and planning. Although each of these tasks is responsible for a particular function, they are dependent on each other, which is why an approach wherein all the different tasks can be integrated into a single unified framework is desirable. While integration has important benefits such as a significant reduction in operator workload and improved decision making, its realisation presents considerable challenges. Few previous works have addressed this topic and even fewer have investigated recent computing paradigms which may greatly assist in the development of a unifying framework.

Multi-agent systems were introduced and investigated in this study as a possible means for achieving integration of operational tasks in chemical plants. Multi-agent systems are the subject of a sub-field of computing research known as agent-based computing. Agent-based computing represents a relatively recent and powerful high-level computing paradigm.

Initially, a number of software applications were developed for the purposes of this study to assist realisation of the operational tasks. To simplify the process of system development and provide guidance for those unfamiliar with multi-agent systems wishing to adopt the proposed technique, an extensive methodology was devised. The operational tasks were then integrated using the proposed methodology to form an integrated multi-agent system, with the pilot plant at Murdoch University being used as a test base for the solution. The results were positive and demonstrated that the proposed agent-based solution was able to effectively account for the pilot plant setting. It was concluded that, in addition to presently available integration techniques and base technologies, the agent-based approach to integration of operational tasks in chemical plants presents a viable alternative solution.
# Contents

**Declaration**

**Abstract** iii

**Contents** v

**List of Figures** ix

**List of Tables** xiii

**Acknowledgements** xv

**Publications** xvii

## 1. Introduction
1.1 Background 1
1.2 Scope of the Study 2
1.3 Structure of the Thesis 3

## 2. Literature Review
2.1 Introduction 5
2.2 Operational Tasks in Chemical Plants 5
2.3 Definition of Integration 9
2.4 Integration of Operational Tasks in Chemical Plants: Previous Studies 10
2.5 Software Architectures and Technologies Enabling Integration 18
   2.5.1 Base Technologies 19
   2.5.2 Integration Styles 25
   2.5.3 Current Trends in Computing 30
2.6 Conclusions 31

## 3. Multi-Agent Systems
3.1 Introduction 35
3.2 Multi-Agent Systems 35
   3.2.1 What is an Agent? 36
3.2.2 What is a Multi-Agent System? 37
3.2.3 Multi-Agent Systems in Perspective 37
3.2.4 The Importance of Multi-Agent Systems 40
3.3 Multi-Agent System Applications 41
  3.3.1 Manufacturing 41
  3.3.2 Process Control 43
  3.3.3 Application Integration 45
  3.3.4 Multi-Agent System Application Guidelines 48
  3.3.5 Multi-Agent System Development Issues 50
3.4 Multi-Agent System Development Toolkits 51
  3.4.1 An Overview of JADE 54
3.5 Multi-Agent System Development Methodologies 58
3.6 Conclusions 60

4. Research Design
  4.1 Introduction 63
  4.2 An Agent-Based Solution 63
  4.3 The Murdoch University Pilot Plant 66
  4.4 Research Procedure 74
  4.5 Conclusions 74

5. Development of the Operational Tasks
  5.1 Introduction 77
  5.2 Data Acquisition 77
  5.3 Regulatory Control 77
  5.4 Monitoring 78
  5.5 Data Reconciliation 80
  5.6 Fault Detection and Diagnosis 93
  5.7 Supervisory Control 107
  5.8 Scheduling and Planning 108
  5.9 Conclusions 108

6. A Methodology for the Development of Multi-Agent Systems using JADE
  6.1 Introduction 109
## List of Figures

2.1 Classic pyramid model of process plant operations (Reklaitis & Koppel 1996) 6
2.2 A framework for integrated process supervision (Rengaswamy 1995) 13
2.3 CHEM integration architecture (Matania 2005) 15
2.4 Coordinated Knowledge Management Method (Power 2004) 17
2.5 File transfer integration style (Hohpe & Woolf 2004) 27
2.6 Shared database integration style (Hohpe & Woolf 2004) 28
2.7 Remote procedure call integration style (Hohpe & Woolf 2004) 28
2.8 Messaging integration style (Hohpe & Woolf 2004) 29
2.9 A basic service-oriented architecture (Hundigam 2004) 30

3.1 Characterisation of a multi-agent system (Jennings 2000) 37
3.2 Paradigm shifts over time (Dignum et al. 2002) 39
3.3 Conventional versus cooperative decision making (Mařík & McFarlane 2005) 42
3.4 Structure of an ARCHON community (Cockburn & Jennings 1996) 44
3.5 Three approaches to agentification (Genesereth & Ketchpel 1994) 46
3.6 Manual vs. automated coordination (Genesereth 1997) 47
3.7 Levels of communication between software (Finin & Labrou 1999) 48
3.8 Containers and platforms in JADE (Caire 2003) 55
3.9 Internal architecture of a generic JADE agent (Bellifimine 2001) 56

4.1 Ground level view of pilot plant 70
4.2 The pilot plant from above 70
4.3 SCAN 3000 overview of the pilot plant 71
4.4 Control room from the outside 71
4.5 Inside the control room 72
4.6 Overview of the precipitation (heating) section 72
<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7</td>
<td>Flowsheet of separation and heating sections</td>
<td>73</td>
</tr>
<tr>
<td>5.1</td>
<td>Steady-state detection</td>
<td>78</td>
</tr>
<tr>
<td>5.2</td>
<td>Exponential filtering of data</td>
<td>79</td>
</tr>
<tr>
<td>5.3</td>
<td>System setup</td>
<td>81</td>
</tr>
<tr>
<td>5.4</td>
<td>A general system with its associated interactions (Stephanopoulos 1984)</td>
<td>82</td>
</tr>
<tr>
<td>5.5</td>
<td>Detailed view of separation and heating sections</td>
<td>95</td>
</tr>
<tr>
<td>5.6</td>
<td>Procedure for data analysis and set preparation</td>
<td>97</td>
</tr>
<tr>
<td>5.7</td>
<td>The normaliser plug-in</td>
<td>98</td>
</tr>
<tr>
<td>5.8</td>
<td>Linear PCA neural network using the Sanger synapse</td>
<td>99</td>
</tr>
<tr>
<td>5.9</td>
<td>Control panel setting of PCA neural network</td>
<td>100</td>
</tr>
<tr>
<td>5.10</td>
<td>Overview of fault detection</td>
<td>100</td>
</tr>
<tr>
<td>5.11</td>
<td>SOM using the Kohonen synapse</td>
<td>103</td>
</tr>
<tr>
<td>5.12</td>
<td>Control panel setting for the SOM neural network</td>
<td>103</td>
</tr>
<tr>
<td>5.13</td>
<td>Feed-forward MLP training overview</td>
<td>104</td>
</tr>
<tr>
<td>5.14</td>
<td>Control panel setting for MLP neural network</td>
<td>104</td>
</tr>
<tr>
<td>5.15</td>
<td>NN.B1 setting</td>
<td>105</td>
</tr>
<tr>
<td>5.16</td>
<td>Delay layer settings</td>
<td>105</td>
</tr>
<tr>
<td>5.17</td>
<td>Overview of fault diagnosis</td>
<td>105</td>
</tr>
<tr>
<td>5.18</td>
<td>The Export NeuralNet function</td>
<td>106</td>
</tr>
<tr>
<td>6.1</td>
<td>Overview of the methodology</td>
<td>110</td>
</tr>
<tr>
<td>6.2</td>
<td>Use case diagram for cinema organiser case study</td>
<td>121</td>
</tr>
<tr>
<td>6.3</td>
<td>Agent diagram for cinema organiser case study after Step 2</td>
<td>121</td>
</tr>
<tr>
<td>6.4</td>
<td>Agent diagram for cinema organiser case study after Step 4</td>
<td>122</td>
</tr>
<tr>
<td>6.5</td>
<td>Agent diagram for cinema organiser case study after Step 5</td>
<td>122</td>
</tr>
<tr>
<td>6.6</td>
<td>Agent deployment diagram for cinema organiser case study after Step 6</td>
<td>123</td>
</tr>
<tr>
<td>6.7</td>
<td>Summary of the analysis phase</td>
<td>124</td>
</tr>
<tr>
<td>6.8</td>
<td>The dynamic template pattern</td>
<td>143</td>
</tr>
<tr>
<td>6.9</td>
<td>Service registrations and searches in the cinema organiser case study</td>
<td>144</td>
</tr>
<tr>
<td>6.10</td>
<td>The listener adding behaviour pattern</td>
<td>144</td>
</tr>
<tr>
<td>6.11</td>
<td>Performing SQL queries on a ‘transduced’ database</td>
<td>145</td>
</tr>
<tr>
<td>6.12</td>
<td>State diagram for ‘let the cinema organiser user select friends to invite’ responsibility</td>
<td>145</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

6.13 The cinema organiser ontology 146
6.14 An alternative ontology to describe the cinema organiser domain 146
6.15 Summary of the design phase 147

7.1 Use case diagram for pilot plant case study 155
7.2 Agent diagram for pilot plant case study after Step 2 155
7.3 Agent diagram for pilot plant case study after Step 4 156
7.4 Agent deployment diagram for pilot plant case study after Step 6 158
7.5 Agent startup (common to all agents except operator agent) 170
7.6 State diagram for fault detection agent 171
7.7 Agent shutdown (common to all agents except operator agent) 171
7.8 State diagram for fault diagnosis agent 173
7.9 State diagram for monitoring agent – steady-state detection 175
7.10 State diagram for monitoring agent – exponential filtering 175
7.11 State diagram for monitoring agent – sensor limit checking 176
7.12 State diagram for monitoring agent – data creation 176
7.13 State diagram for data acquisition agent 178
7.14 State diagram for supervisory control agent 179
7.15 State diagram for data reconciliation agent 181
7.16 Operator agent’s GUI after initialisation 191
7.17 Data reconciliation agent’s GUI initialisation 191
7.18 Operator agent notified that all agents initialised 192
7.19 The GUI provided by JADE giving an overview of containers after all agents have been initialised 192
7.20 Operator alerted that agents have been started and are running 193
7.21 Agents are running normally 193
7.22 Data reconciliation agent’s GUI display during normal operation 194
7.23 Operator agent alerted that a fault has occurred 194
7.24 Operator alerted that sensors outside their limits 195
7.25 Operator alerted that plant no longer at steady-state 195
7.26 Operator alerted that fault has been diagnosed and asked whether fault has been rectified 196
7.27 Display after fault rectified in plant 196
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.28</td>
<td>Plot of sensor history as system is returning to steady-state after fault rectified in the plant</td>
</tr>
<tr>
<td>7.29</td>
<td>Operator alerted that gross error detected and asked if error rectified</td>
</tr>
<tr>
<td>7.30</td>
<td>Operator alerted that random error detected and that reconciled data will be used for any subsequent calculations</td>
</tr>
<tr>
<td>7.31</td>
<td>Operator alerted that a set-point change is desirable and asked whether they would like to view the optimised set-points</td>
</tr>
<tr>
<td>7.32</td>
<td>The optimised set-points are presented to the operator</td>
</tr>
<tr>
<td>7.33</td>
<td>Operator agent’s GUI after agents have been stopped</td>
</tr>
<tr>
<td>7.34</td>
<td>Data reconciliation agent’s GUI after it has been stopped</td>
</tr>
<tr>
<td>7.35</td>
<td>Operator agent’s GUI after agents have been killed</td>
</tr>
<tr>
<td>7.36</td>
<td>Operator agent’s GUI before operator agent is killed</td>
</tr>
<tr>
<td>7.37</td>
<td>Data acquisition agent’s command-prompt output</td>
</tr>
<tr>
<td>7.38</td>
<td>Data reconciliation agent’s command-prompt output</td>
</tr>
<tr>
<td>7.39</td>
<td>Fault diagnosis agent’s command-prompt output</td>
</tr>
<tr>
<td>7.40</td>
<td>Monitoring agent’s command-prompt output</td>
</tr>
<tr>
<td>7.41</td>
<td>Supervisory control agent’s command-prompt output</td>
</tr>
<tr>
<td>7.42</td>
<td>Operator agent’s command-prompt output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Faults 1-13</td>
</tr>
<tr>
<td>A.2</td>
<td>Dimensionally reduced normal data</td>
</tr>
<tr>
<td>A.3</td>
<td>Ellipse encompassing dimensionally reduced normal data</td>
</tr>
<tr>
<td>A.4</td>
<td>Normalisation of NN.A</td>
</tr>
<tr>
<td>A.5</td>
<td>Flowchart for fault detection and diagnosis</td>
</tr>
</tbody>
</table>
## List of Tables

2.1 Middleware summary (adapted from Braunschweig (2005)) 23

4.1 How an agent agent-based solution addresses the requirements of an integrated chemical plant 65
4.2 Process set-points 73
4.3 Control tuning parameters 73

5.1 Measured and unmeasured variables 85
5.2 Units of measured variables 87
5.3 Constants 90
5.4 Simulated faults based on Figure 5.5 95

6.1 Responsibility table for cinema organiser case study after Step 3 121
6.2 Responsibility table for cinema organiser case study after Step 4 122
6.3 Responsibility table for cinema organiser case study after Step 5 123
6.4 Interaction table for cinema organiser agent after Step 2 of design 143
6.5 Interaction table for cinema organiser agent after Step 4 of design 143

7.1 Responsibility table for pilot plant case study after Step 3 156
7.2 Responsibility table for pilot plant case study after Step 4 157
7.3 Responsibilities requiring an interaction (underlined) 165
7.4 Fault detection agent interaction table 166
7.5 Fault diagnosis agent interaction table 167
7.6 Monitoring agent interaction table 167
7.7 Data acquisition agent interaction table 168
7.8 Supervisory control agent interaction table 169
7.9 Data reconciliation agent interaction table 169
7.10 Operator agent interaction table 170
<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.11</td>
<td>Fault detection agent behaviours</td>
<td>172</td>
</tr>
<tr>
<td>7.12</td>
<td>Fault detection agent interaction table updated</td>
<td>172</td>
</tr>
<tr>
<td>7.13</td>
<td>Fault diagnosis agent behaviours</td>
<td>174</td>
</tr>
<tr>
<td>7.14</td>
<td>Fault diagnosis agent interaction table updated</td>
<td>174</td>
</tr>
<tr>
<td>7.15</td>
<td>Monitoring agent behaviours</td>
<td>176</td>
</tr>
<tr>
<td>7.16</td>
<td>Monitoring agent interaction table updated</td>
<td>177</td>
</tr>
<tr>
<td>7.17</td>
<td>Data acquisition agent behaviours</td>
<td>178</td>
</tr>
<tr>
<td>7.18</td>
<td>Data acquisition agent interaction table updated</td>
<td>178</td>
</tr>
<tr>
<td>7.19</td>
<td>Supervisory control agent behaviours</td>
<td>179</td>
</tr>
<tr>
<td>7.20</td>
<td>Supervisory control agent interaction table updated</td>
<td>180</td>
</tr>
<tr>
<td>7.21</td>
<td>Data reconciliation agent behaviours</td>
<td>181</td>
</tr>
<tr>
<td>7.22</td>
<td>Data reconciliation agent interaction table updated</td>
<td>182</td>
</tr>
<tr>
<td>7.23</td>
<td>Operator agent behaviours</td>
<td>183</td>
</tr>
<tr>
<td>7.24</td>
<td>Operator agent interaction table updated</td>
<td>183</td>
</tr>
<tr>
<td>7.25</td>
<td>Comparison of major works with this study</td>
<td>211</td>
</tr>
</tbody>
</table>
Acknowledgments

Firstly, I would like to express my sincerest gratitude to my supervisor, Associate Professor Parisa A. Bahri, for her support and guidance throughout the course of this research. I was honoured and fortunate enough to be one of her PhD students.

The productive environment at Murdoch University has fostered the success of this research. I am delighted to thank the staff and students at the School of Electrical, Energy and Process Engineering. I would particularly like to thank Hugh McConville for his assistance with the pilot plant, Daniel McGill for proofreading this thesis and Jill Brown for her help with administrative matters.

A very special thank-you goes to Giovanni Caire of Telecom Italia Lab (TILAB) in Turin, Italy. His help with aspects of the JADE software and the multi-agent system development methodology has been absolutely fundamental to the success of this research. I am also grateful to TILAB for allowing me to visit their premises in December 2004 to work on the development of the methodology.

I should also thank Professor Peter L. Lee who was formerly Executive Dean of Engineering, Science and Computing at Curtin University of Technology and is now with the University of South Australia. His advice pertaining to the organisation of this thesis is appreciated.

Thanks to Dr Yvonne Power for her useful suggestions, especially in the initial stages of the research.

Most importantly, I wish to thank my wife Mahsa who has been a constant source of encouragement and unconditional support. I am also grateful to our families, relatives and friends for their caring attitude during this period.

Finally, I would like to thank the Australian Research Council (ARC), Murdoch University and the Parker Centre for funding this research.

I apologise to those whose contributions, inadvertently, have not been acknowledged.
Publications

Several parts of the work and ideas presented in this thesis have been published by the author during the course of the research. These publications are listed below.


Theoretical Physics & Mathematics (IPM) and Computer Society of Iran (CSI), pp. 707-714.


