Creating Agile Business Systems with Reusable Knowledge

Agility and innovation are necessary to achieve global excellence and customer value in twenty-first century business; yet most approaches to business process engineering in existence today sacrifice these in favor of operational efficiency and economics. Moreover, the IT systems used to automate and encapsulate business processes are inflexible and unable to respond to the constantly changing business environment. Mitra and Gupta provide insight to close this gap – they show how innovation can be systematized with normalized patterns of information, how business processes and information systems may be tightly aligned, and how these processes and systems can be designed to automatically adapt to change by re-configuring shared patterns of knowledge. The authors present a modular approach to building business systems that parallels that of object oriented software. They describe how business knowledge can be identified, encapsulated, and shared, as well as how reusable process modules can be developed to offer the systems flexibility. The book provides practical templates required for accelerating integration, analysis, and design. Mitra and Gupta lay the foundation of a new paradigm in which computers manipulate meanings, not blind symbols.

This book will appeal to consultants, analysts, and managers in IT firms looking to develop new, more flexible products for their clients. It will also be of interest to researchers and graduate students in business, management, and IT. By deeply integrating business knowledge and processes with IT systems design, this book is a valuable addition to both communities.

Amit Mitra is Managing Consultant at Headstrong LLC, in addition to President and Principal Consultant at Sprybiz LLC. He is an alumnus of KPMG and former Chief Methodologist of the American International Group.

Amar Gupta holds a number of positions at the University of Arizona, Tucson. He is Professor of Entrepreneurship and MIS; Thomas R. Brown Chair in Management and Technology; and Senior Director for Research and Business Development at the Eller College of Management; and he is Professor of Computer Science in the College of Science. In addition, he serves as Visiting Professor, Engineering Systems Division, College of Engineering at the Massachusetts Institute of Technology (MIT).
Creating Agile Business Systems with Reusable Knowledge

A. Mitra and A. Gupta
I dedicate this book to my father, Ajoy Mitra, my mother, Sevati Mitra, and my teachers, who helped shape my thoughts, to my wife Snigdha and my children Tanya and Trishna, who helped shape my life, and in turn shape this book.

Amit Mitra

I dedicate this book to my mother, my sister (Beena), my wife (Poonam), my children (Amrit and Amita), and to my teachers in India (especially at IIT, Kanpur and IIT, Delhi), the UK (at Birkbeck College and the Imperial College of Science and Technology), and the US (primarily at MIT).

Amar Gupta
Contents

List of boxes \hspace{1cm} page ix
List of figures \hspace{1cm} xi
List of tables \hspace{1cm} xiv
Foreword \hspace{1cm} xv
Preface \hspace{1cm} xvii
Acknowledgements \hspace{1cm} xix

Introduction \hspace{1cm} 1
1 What is this book about and who should read it? \hspace{1cm} 1
2 What will the information be used for? \hspace{1cm} 2
3 Technology’s broken promise \hspace{1cm} 2
4 Component reuse – the genesis \hspace{1cm} 4
5 Scope of this book \hspace{1cm} 6
6 Foundation of knowledge reuse: three pillars \hspace{1cm} 7
7 How this book is organized \hspace{1cm} 11

1 On the nature of reality and the nature of business \hspace{1cm} 14
1 The nature of knowledge \hspace{1cm} 16
2 Modeling the real world \hspace{1cm} 20
3 Metaworld of information \hspace{1cm} 21
4 Basic metaobject inventory \hspace{1cm} 39
5 Metaobjects and the natural repository of knowledge \hspace{1cm} 39
6 The architecture of knowledge and the scope of the metamodel in this book \hspace{1cm} 43

2 The object at the root of it all \hspace{1cm} 51
1 Object class versus object instance \hspace{1cm} 52
2 The state of an object \hspace{1cm} 55
3 Inheriting behavior – subtypes, supertypes, and partitioning of objects \hspace{1cm} 79
4 The problem of perspective \hspace{1cm} 94
5 Repositories of meaning \hspace{1cm} 103
<table>
<thead>
<tr>
<th>3</th>
<th>The nature of attributes</th>
<th>112</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 The structure of attributes and states</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>2 Attribute constraints</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>3 Naming and expressing attributes</td>
<td>158</td>
</tr>
<tr>
<td>4</td>
<td>Domains and their expression</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>1 The meaning and architecture of format</td>
<td>172</td>
</tr>
<tr>
<td></td>
<td>2 The meaning of units of measure</td>
<td>234</td>
</tr>
<tr>
<td></td>
<td>3 Domains of meaning and the metamodel of object</td>
<td>252</td>
</tr>
<tr>
<td></td>
<td>4 Storing abstract meaning</td>
<td>324</td>
</tr>
<tr>
<td></td>
<td>Appendix: Key shared components of knowledge</td>
<td>327</td>
</tr>
</tbody>
</table>

*Bibliography* 354

*Index* 383
Boxes

1 Example of the process engineering value chain  page 5
2 Example of a model for baking a cookie  21
3 Objects, domains, and formats  27
4 Mr. Domain’s calculations  31
5 Conversion between UOMs  35
6 The architecture of knowledge  44
7 The architecture of knowledge reuse can help make information systems flexible and scalable  47
8 How rules shift between business process automation and technology layers  48
9 Business definition of an object  51
10 Properties of objects  60
11 States, attributes, state variables, and type indicators: much ado about nothing  60
12 The state of a system is the collective state of the objects it involves  66
13 Set membership  67
14 Object instances must exist for a finite period of time once they are created (often forever after they reach some terminal state)  68
15 Mathematical definition of a business object  71
16 Domains and measures of distance  72
17 State spaces with qualitative attributes  75
18 Chaotic behavior  78
19 Set operations  81
20 Inheritance, state space, and polymorphism  82
21 Exclusion partitions, variation inheritance, and polymorphism  86
22 UML syntax for partitions and subtypes  89
23 Subtyping criteria  93
24 Perspective is an object  96
25 An example of the problem of perspective  99
26 Synonyms and homonyms  101
27 The OAV model and the structure of attributes  114
28 Components of inclusion and exclusion sets  128
29 The structure of bounds and ranges  133
30 Set versus list  134
31 Dormant ranges and containers for value sets  135

© Cambridge University Press  www.cambridge.org
List of boxes

32 Cardinality, cardinality ratio, and object counts 141
33 Relationships between attributes, meanings, and expressions 146
34 Identifying domains 160
35 The many faces of meaning 164
36 When one object represents another 173
37 Location in space 187
38 Metamodels of format, format conversion, encryption, and formatting constraint 234
39 Full formats of values 243
40 Measure conversion 248
41 Conflicting subtypes, state spaces, perspectives, and polymorphisms of metaobjects 251
42 Domain analysis and primary physical domains 264
43 The principle of subtyping by adding information 269
44 Domain analysis for components of knowledge 281
45 Partial order, fuzzy meaning, and the scaling of derived domains 284
46 “Softness” of information 293
47 Domains, relationships, and the Cartesian product 296
48 Domain rules ready reckoner 314
49 The information content of domains – new learning and changing perspectives – an example in color 322
## Figures

1. Information is a key resource, but investment in information systems is fraught with risk  
   *page 3*

2. Reusing business knowledge: the three pillars  
   *page 8*

3. Analysis paralysis: only a few critical rules, reused often, connect the business of the enterprise but they are lost in a tangled web of minutiae  
   *page 10*

4. More and more, past experience is under pressure from the demands of scale, scope, and agility that businesses are placing on systems  
   *page 15*

5. Knowledge is the meaning of business practices, rules, goals, and guidelines and how they fit into an integrated whole  
   *page 17*

6. Processes are a special kind of relationship: they carry information on “before and after” effects between objects  
   *page 23*

7. How is information naturally manifest in the real world?  
   *page 24*

8. (Partial) metamodel of domain  
   *page 34*

9. Basic metabject inventory: *kinds of rules each metaobject normalizes*  
   *page 40*

10. A rule: organization ships product assembled from objects  
    *page 41*

11. Adding components to assemble configurations of rules  
    *page 42*

12. The black box perspective of behavior  
    *page 44*

13. Process decomposition  
    *page 45*

14. Node–branch representation  
    *page 45*

15. The architecture of knowledge  
    *page 46*

16. Business process automation is only one of several mechanisms that implement abstract business rules in the physical world  
    *page 48*

17. The architecture of reusable knowledge components  
    *page 49*

18. The scope of the metamodel of knowledge in this book is focused on pure business rules  
    *page 49*

19. Object classes are collections of object instances with common properties  
    *page 53*

20. Object classes are collections of object instances with common properties (state chart perspective)  
    *page 56*

21. Disallowed effects, default, and “maybe” states  
    *page 57*

22. The state of an object changes in response to discrete events  
    *page 70*

23. Example of two-dimensional state space  
    *page 73*

24. Example of a three-dimensional state space  
    *page 74*

25. Example of trajectory through state space  
    *page 78*
List of figures

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Example of object partitions</td>
<td>84</td>
</tr>
<tr>
<td>27</td>
<td>Constraints across partitions</td>
<td>88</td>
</tr>
<tr>
<td>28</td>
<td>Perspective is a point of view</td>
<td>95</td>
</tr>
<tr>
<td>29</td>
<td>Name is an object class linked to perspective</td>
<td>103</td>
</tr>
<tr>
<td>30</td>
<td>Primary names, perspectives, and aliases</td>
<td>103</td>
</tr>
<tr>
<td>31</td>
<td>Fundamental metamodel of object</td>
<td>105</td>
</tr>
<tr>
<td>32</td>
<td>Expanded inventory of metaobjects</td>
<td>106</td>
</tr>
<tr>
<td>33</td>
<td>Views are aggregate objects</td>
<td>109</td>
</tr>
<tr>
<td>34</td>
<td>An attribute is the overlap between an object and a domain</td>
<td>113</td>
</tr>
<tr>
<td>35</td>
<td>The structure of attributes and metamodel of state</td>
<td>117</td>
</tr>
<tr>
<td>36</td>
<td>Attributes can represent an object’s participation in relationships</td>
<td>119</td>
</tr>
<tr>
<td>37</td>
<td>Attributes and values inherit properties of domains</td>
<td>120</td>
</tr>
<tr>
<td>38</td>
<td>Attributes values may be constrained</td>
<td>124</td>
</tr>
<tr>
<td>39</td>
<td>Inclusion and exclusion sets are mutually exclusive partitions</td>
<td>125</td>
</tr>
<tr>
<td>40</td>
<td>Subtypes of ordinal domains may contain both discrete values and ranges</td>
<td>130</td>
</tr>
<tr>
<td>41</td>
<td>Metamodel of value set</td>
<td>133</td>
</tr>
<tr>
<td>42</td>
<td>Metamodel of value constraint</td>
<td>139</td>
</tr>
<tr>
<td>43</td>
<td>The state of the check is constrained jointly by two other attributes</td>
<td>143</td>
</tr>
<tr>
<td>44</td>
<td>A quantitative relationship relates check amount, monthly rental, and energy charges</td>
<td>143</td>
</tr>
<tr>
<td>45</td>
<td>A rule constraint is an aggregate object</td>
<td>145</td>
</tr>
<tr>
<td>46</td>
<td>Intermediate rules for mapping object instances to object population</td>
<td>148</td>
</tr>
<tr>
<td>47</td>
<td>Constrain is a recursive relationship</td>
<td>155</td>
</tr>
<tr>
<td>48</td>
<td>Metamodel of “rule constrain”</td>
<td>156</td>
</tr>
<tr>
<td>49</td>
<td>Integrated metamodel of value constraint</td>
<td>157</td>
</tr>
<tr>
<td>50</td>
<td>Patterns are symbols that consist of symbols</td>
<td>177</td>
</tr>
<tr>
<td>51</td>
<td>Examples of finite delimited and undelimited patterns</td>
<td>185</td>
</tr>
<tr>
<td>52</td>
<td>Metamodel of measure of similarity</td>
<td>201</td>
</tr>
<tr>
<td>53</td>
<td>Kinds of patterns</td>
<td>207</td>
</tr>
<tr>
<td>54</td>
<td>The metamodel of pattern is the source of universal properties, subtypes and partitions of pattern</td>
<td>211</td>
</tr>
<tr>
<td>55</td>
<td>Perspectives of object – two-dimensional cross sections of a three-dimensional array</td>
<td>217</td>
</tr>
<tr>
<td>56</td>
<td>Candu Compoot’s three-dimensional array</td>
<td>218</td>
</tr>
<tr>
<td>57</td>
<td>Integrated metamodel of pattern</td>
<td>222</td>
</tr>
<tr>
<td>58</td>
<td>Structure of language</td>
<td>225</td>
</tr>
<tr>
<td>59</td>
<td>Format conversion is a recursive aggregate relationship</td>
<td>230</td>
</tr>
<tr>
<td>60</td>
<td>The metamodel of format maps values to symbols</td>
<td>231</td>
</tr>
<tr>
<td>61</td>
<td>The metamodel of format conversion maps symbols to symbols</td>
<td>231</td>
</tr>
<tr>
<td>62</td>
<td>Metamodel of representation</td>
<td>233</td>
</tr>
<tr>
<td>63</td>
<td>Two maps working in tandem map quantitative values to physical symbols</td>
<td>236</td>
</tr>
<tr>
<td>64</td>
<td>Metamodel of unit of measure</td>
<td>244</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>65</td>
<td>Structure of unit of measure conversion rules</td>
<td>247</td>
</tr>
<tr>
<td>66</td>
<td>Conflicting subtypes?</td>
<td>249</td>
</tr>
<tr>
<td>67</td>
<td>Information hierarchy of domains</td>
<td>259</td>
</tr>
<tr>
<td>68</td>
<td>The metamodel of domain</td>
<td>312</td>
</tr>
<tr>
<td>69</td>
<td>Shifting perspectives of color</td>
<td>319</td>
</tr>
</tbody>
</table>
## Tables

<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Proximity metrics in spaces of different kinds</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>Valid measures of proximity in mixed spaces</td>
<td>203</td>
</tr>
<tr>
<td>3</td>
<td>A two-dimensional slice of Candu’s nine-dimensional array</td>
<td>218</td>
</tr>
</tbody>
</table>
Foreword

The health and well-being, and today the very survival, of an enterprise depend on its ability to respond and adapt in timely, innovative, and effective manners. The relatively static behaviors of the past have been changed unalterably by the explosion of telecommunication and information technologies/capabilities as typified through the emergence of the World-Wide Web (W³). Enterprises are learning to adapt to the challenges of the new global business and national security environment by exploiting the same capabilities that are driving the dynamic environment, telecommunications and IT. In essence, information and the knowledge derived therefrom have emerged as key assets of the enterprise in responding and adapting to the demands of the global environment.

The experiences over the past decade for a wide variety of enterprises, including both governmental and commercial entities, are reflected by more failures than successes in embracing successful strategies and solutions for creating, engineering, and evolving the knowledge system that serves the enterprise most effectively. Certainly, problems have arisen through failures of leadership and management, who have been unable to break the static behaviors and narrow organizational views that served them well in the past. On the other hand, engineers, given responsibility for exploiting information technologies to meet the information needs of leadership and management for knowledgeable decision making, have found the challenges of dealing with the complex event-driven environments and the complex array of enterprise stakeholders and systems vastly more difficult than the systems engineering problems of the past.

From both successful and failed efforts, there is an ever-growing body of knowledge about some of the keys to successfully reengineering the global enterprise as a flexible and adaptive entity. The concepts that are seeing increasing attention include enterprise architecting, service-oriented architectures, business process modeling, enterprise and e-business patterns, enterprise systems engineering, and agile development methods. All of these approaches and methods contribute to one or more of the fundamental advantages that are driving the developments. These advantages span a number of enterprise dimensions and can be summarized in the following way:

- **Strategic focus**: provides a basis for understanding the contributions of complex, large, distributed information systems in achieving enterprise goals and missions.
- **Broadened communications**: enhances communication across the enterprise community from leaders/managers/users to engineers/developers/testers.
• **Performance and QoS**: improves performance and quality of service (QoS) for decision support and knowledge-based decision making.
• **Timely and flexible response**: enables flexibility for timely and effective response to new and unexpected situations.
• **Integration and interoperable operations**: enables mechanisms for assuring integrated and interoperable applications, both among legacy and new systems.
• **Commercial technology evolution**: facilitates the introduction and effective use of rapidly changing commercial information systems and technology.
• **Cost-effective migration**: establishes a foundation for value-based thinking, analysis of alternatives, and investment planning to establish cost-effective system evolution.
• **Organizational efficiencies**: allows organizational efficiencies due to reduced staffing requirements, easier system evolution process, etc.

But from reviewing and assessing a myriad of enterprise developments, it is apparent that a much deeper understanding is needed to increase the probability of success for enterprises working to meet the challenges of the global environment. In this book, complemented by their earlier book, *Agile Systems with Reusable Patterns of Business Knowledge*, the authors provide a very comprehensive and integrated perspective on the range of topics mentioned above. Starting from basic principles, the book presents an approach to enterprise reengineering that merits careful attention and thoughtful application. As they say in the Preface, the book provides a description of a “hidden and elegant theoretical framework: a framework that is a direct bridge between business process engineering and systems engineering.” The approach that is presented is ambitious and provocative, and commands thoughtful consideration from developers and researchers in this field of ever-increasing importance.

**Harold W. Sorenson**

*Professor of Engineering Systems*

*Faculty Director*

*Graduate Program in Architecture-based Engineering of Enterprise Systems*

*Jacobs School of Engineering*

*Rady School of Management*

*University of California, San Diego*

*La Jolla, CA 92037*

[Former Chief Scientist, US Air Force; Former Senior Vice President and General Manager for Air Force Systems at MITRE; Former Chief Engineer for AF Electronic Systems Center]
Preface

Why this book? – Because it is a book begging to be written. The real world is chaotic and never stands still. Businesses constantly strive to re-invent themselves under continual, and often intense, pressures of competitive, regulatory, and technological change. The pivotal issue in business computing lies in incorporating new learning in automated information systems; adding to what is already known and adapting automatically as perspectives and priorities continually change. This is the challenge for which we have sought the answers presented in this book.

Where did it all start? It was 1992. One of the authors was the Chief Methodologist for the American International Group, an unusual global corporation that believed in turning on a dime. The firm needed a systems development discipline to facilitate nimble and innovative business practices. Thus a truly exciting and wonderful journey began – a journey we want to share with you. In this book, you will find readily usable patterns and models you can leverage to establish business requirements, object models, and knowledge bases to support the agile and exacting business needs of the twenty-first century. You will also find the exciting and simple beauty of a framework that is the direct bridge between business process engineering and systems engineering. Yes, it is a proven framework that works for every industry and business application we have tested it on – from telecommunications to insurance, from financial services to manufacturing.

In the chapters that follow, we will share with you not only how this framework works in practice, but also how it actually anticipates key requirements even before users articulate them, such as those that flowed from strategic shifts in the regulatory bedrock of the US telecommunications industry. The theoretical foundation of the approach is not only deep, but also elegant in its simplicity.

Where will it eventually lead? In the short term, it can make your business more agile. It can provide reusable models, processes, and business knowledge components to compress your time to market new or improved products, services, and processes. It can also show you how you can compress systems development and integration times. However, it is the vision at the end of this journey that is the most fascinating of all. The concepts in this book can provide the foundation of disciplines that can make business systems truly maintenance free – systems based on software that can automatically adapt to change and chaos. These systems can be supported by automated intelligent agents¹ that will, some day, maintain

¹ The “Intelligent agents” section of the Bibliography at the end of this book lists papers that describe agent technology and the-state-of-the-art.
software and respond to environmental change at the speed of thought, a vision we will share with you at the end of this book.

As practitioners, managers, and teachers in the field of information systems, we often talk of change control. Change plays havoc with our plans and products. However, the wealth in the knowledge economy will flow from global excellence, thriving on change and innovation. The only justification for technology will, and must be, change facilitation, not change control. Are we ready?

“Wouldst thou,” – so the helmsman answered
“Learn the secret of the sea?
Only those who brave its dangers
Comprehend its Mystery!”

(Henry Wadsworth Longfellow,
The Galley of Count Arnaldos)
Acknowledgements

We deeply appreciate the time that Dr. E. C. Subbarao, our former teacher, spared from his busy schedule to help shape this book. We thank Srividhya Subramanian, Yan An Surendra Sarnikar, Shivram Mani, and Prithi Avanavadi for going through this entire manuscript carefully. We are grateful to Jayant Pal and Raghunandan Dhat of Setu Inc. for their enthusiasm in instantiating the concepts in software, and to Pat Fayad, for keeping the faith, and her unflagging encouragement as this book was being written. Finally, one of the authors (Amar Gupta) would like to thank the Engineering Systems Division of the Massachusetts Institute of Technology (MIT), especially Professor Daniel Hastings and Professor Joel Moses, for providing a Visiting Professorship and other support in 2005 and 2006 when this book was being finalized.