A FORMALIZATION AND EXTENSION OF THE PURDUE ENTERPRISE
REFERENCE ARCHITECTURE AND THE PURDUE METHODOLOGY

REPORT NUMBER 158

Purdue Laboratory for Applied Industrial Control

Prepared by

Hong Li
Theodore J. Williams

December 1994

Purdue Laboratory for Applied Industrial Control
School of Engineering
Purdue University
West Lafayette, Indiana 47907
This Report is Published as Part of
Engineering Experiment Station Bulletin 143 Series

School of Engineering
Purdue University
West Lafayette, Indiana 47907
FOREWORD

This report is effectively a reprinting of the Dissertation for the degree of Doctor of Philosophy in Industrial Engineering at Purdue University in December 1994. Dr. Li's doctoral program was carried out through the School of Industrial Engineering and his work was done in the Purdue Laboratory for Applied Industrial Control under the direction of the undersigned. His program featured an extensive review of the Purdue Enterprise Reference Architecture published by the Instrument Society of America in October 1992. His project developed several important extensions to the earlier work. These involved particularly:

1. A theoretical basis for the Architecture developed from the Design Axioms of Professor Nam Suh.
2. Development of a classification scheme for the interfaces involved in the Architecture and an extensive discussion of their meaning and use.
3. Illustration of the recursive nature of the architecture and its meaning and an extensive demonstration of its universal applicability.
4. A hierarchical scheme for specifying the various types of formality which can be applied to the architecture and its applications.

These we believe will be very important to PERA's future acceptance and use.

Theodore J. Williams,
Director,
Purdue Laboratory for
Applied Industrial Control
# TABLE OF CONTENTS

<p>| LIST OF TABLES | .......................................................... | v |
| LIST OF FIGURES | .......................................................... | |
| 1. INTRODUCTION AND PURPOSE | .................................................. | 1 |
| 1.1 Introduction | ............................................. | 1 |
| 1.2 Problems and Approaches | ......................................... | 5 |
| 1.2.1 Formalism Involved in Information Presentation between Program Participants | | 5 |
| 1.2.2 Formalism Involved in Development of an Enterprise Reference Architecture and Methodology | | 7 |
| 1.2.3 Formalism Involved in Information Transformation through Interfaces | | 8 |
| 1.4.4 Basic Definitions Concerning Interfaces | | 9 |
| 2. BACKGROUND: WHAT ARE ENTERPRISE REFERENCE ARCHITECTURES AND WHY DO WE NEED THEM | .................................................. | 13 |
| 2.1 Needs of Enterprise Integration | .......................................... | 13 |
| 2.2 Definitions of Enterprise Integration and Enterprise Reference Architecture | ................................................. | 16 |
| 2.3 Functions of Enterprise Reference Architectures | ............................................... | 18 |
| 2.4 Types of Reference Architectures | ........................................ | 22 |
| 2.5 Needs for Enterprise Integration Methodology | .......................................................... | 25 |
| 2.5.1 Problems beyond the Enterprise Reference Architecture | .......................................... | 25 |
| 2.5.2 Components of an Enterprise Integration Methodology | ........................................ | 26 |
| 2.5.3 Formalism Required by Methodology | .................................................. | 29 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.4 A Short Review of the Available Type Two Reference Architectures and their Associated Methodologies</td>
<td>30</td>
</tr>
<tr>
<td>3. A DESCRIPTION OF THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE AND ITS ASSOCIATED METHODOLOGY</td>
<td>33</td>
</tr>
<tr>
<td>3.1 The Purdue Enterprise Reference Architecture</td>
<td>33</td>
</tr>
<tr>
<td>3.1.1 A Life Cycle Presentation on the Purdue Architecture</td>
<td>34</td>
</tr>
<tr>
<td>3.1.2 The Choice of Human Tasks</td>
<td>45</td>
</tr>
<tr>
<td>3.1.3 Development of the Modular Task and Function Representation Method</td>
<td>49</td>
</tr>
<tr>
<td>3.1.4 The Task Modules and Function Representation Method</td>
<td>68</td>
</tr>
<tr>
<td>3.2 The Purdue Methodology</td>
<td>74</td>
</tr>
<tr>
<td>3.3 Tools, Etc., Available for Exploiting the Architecture</td>
<td>80</td>
</tr>
<tr>
<td>4. CIMOSA, GRAI-GIM AND THEIR FORMALISMS</td>
<td>93</td>
</tr>
<tr>
<td>4.1 Introduction to CIMOSA</td>
<td>93</td>
</tr>
<tr>
<td>4.2 CIMOSA Approach and its Modeling</td>
<td>96</td>
</tr>
<tr>
<td>4.2.1 CIMOSA Modeling Framework and its Life Cycle</td>
<td>97</td>
</tr>
<tr>
<td>4.2.2 CIMOSA Modeling Constructs and Modeling Methods</td>
<td>100</td>
</tr>
<tr>
<td>4.2.3 CIMOSA Implementation Modeling Structure and IIS</td>
<td>117</td>
</tr>
<tr>
<td>4.3 Introduction to GRAI-GIM</td>
<td>126</td>
</tr>
<tr>
<td>4.4 GRAI-GIM and its Formalisms</td>
<td>130</td>
</tr>
<tr>
<td>4.4.1 GRAI Reference Model and Structured Approach</td>
<td>130</td>
</tr>
<tr>
<td>4.4.2 GRAI-GIM Modeling Formalisms and Modeling Framework</td>
<td>136</td>
</tr>
<tr>
<td>5. APPROACHES TO FORMALIZATION OF PERA AND THE PURDUE METHODOLOGY THROUGH INTERFACE CONCEPTS</td>
<td>143</td>
</tr>
<tr>
<td>5.1 Formalism Depends on Concerns for Enterprise Integration</td>
<td>143</td>
</tr>
<tr>
<td>5.2 Formalization of Purdue Architecture for Information Presentation between Program Participants</td>
<td>145</td>
</tr>
<tr>
<td>5.3 Semantic Basis for Further Development of</td>
<td></td>
</tr>
</tbody>
</table>
Enterprise Reference Architectures  .........  148
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>The Importance of Interfaces in Enterprise Integration and their Treatment</td>
<td>150</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Introduction</td>
<td>150</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Interface Standards in Enterprise Integration</td>
<td>150</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Categories of Interfaces</td>
<td>156</td>
</tr>
<tr>
<td>5.4.4</td>
<td>A Discussion of Internal Interfaces</td>
<td>156</td>
</tr>
<tr>
<td>5.4.5</td>
<td>The Use of the Architecture Diagrams for the Representation of Interfaces in Integrated Enterprises</td>
<td>160</td>
</tr>
<tr>
<td>5.4.6</td>
<td>Some Further Definitions of Internal Interfaces</td>
<td>171</td>
</tr>
<tr>
<td>5.5</td>
<td>A Discussion of Program Interfaces</td>
<td>172</td>
</tr>
<tr>
<td>5.6</td>
<td>Additional Architectural Considerations</td>
<td>180</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Representation of Entity Interactions</td>
<td>180</td>
</tr>
<tr>
<td>5.7</td>
<td>Promotion of a Minimum Formalism</td>
<td>187</td>
</tr>
<tr>
<td>6.</td>
<td>THE USE OF THE AXIOMS OF ENGINEERING DESIGN AS A BASIS FOR DESCRIBING A GENERALIZED ENTERPRISE DEVELOPMENT PROGRAM’S CHARACTERIZATION AND EXECUTION</td>
<td>191</td>
</tr>
<tr>
<td>6.1</td>
<td>The Axioms of Design</td>
<td>191</td>
</tr>
<tr>
<td>6.2</td>
<td>Applicability of Suh’s Axioms</td>
<td>196</td>
</tr>
<tr>
<td>6.3</td>
<td>Functional Requirements and Design Parameters of the Purdue Methodology</td>
<td>201</td>
</tr>
<tr>
<td>6.4</td>
<td>An Analysis: What is PERA and the Purdue Methodology</td>
<td>202</td>
</tr>
<tr>
<td>7.</td>
<td>EXTENSION OF PERA</td>
<td>237</td>
</tr>
<tr>
<td>7.1</td>
<td>Comparison of Models of System Lifecycle</td>
<td>237</td>
</tr>
<tr>
<td>7.2</td>
<td>An Enterprise Genealogy</td>
<td>244</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Types of Enterprise Entities Related to Enterprise Lifecycles</td>
<td>244</td>
</tr>
<tr>
<td>7.2.2</td>
<td>A Genealogical Tree of PERA Enterprise Development</td>
<td>248</td>
</tr>
<tr>
<td>7.2.3</td>
<td>The Purdue Guide as an Information Product and Tool Serving for Enterprise Development</td>
<td>255</td>
</tr>
<tr>
<td>7.3</td>
<td>Discussions of the Genealogical Tree of PERA</td>
<td>259</td>
</tr>
<tr>
<td>7.3.1</td>
<td>PERA’s Contribution to System Thinking</td>
<td>259</td>
</tr>
</tbody>
</table>
and Systems Engineering ............... 259
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.3.2</td>
<td>The Architectural Formalism of PERA</td>
<td>264</td>
</tr>
<tr>
<td>7.4</td>
<td>Program Iteration Shown by the PERA Lifecycle Model</td>
<td>273</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Implications of Program Interfaces</td>
<td>275</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Graphic Model of Program Iteration by PERA</td>
<td>276</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Program Management and Program Stability</td>
<td>278</td>
</tr>
<tr>
<td>8.</td>
<td>COMPARISONS BETWEEN TYPE TWO ENTERPRISE REFERENCE ARCHITECTURES</td>
<td>283</td>
</tr>
<tr>
<td>8.1</td>
<td>Comparison Done by the IFAC/IFIP Task Force on Architectures for Integrating Manufacturing Activities and Enterprises</td>
<td>283</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Evaluation Methods developed by the Task Force</td>
<td>284</td>
</tr>
<tr>
<td>8.1.2</td>
<td>The Co-Representation Mapping Method</td>
<td>286</td>
</tr>
<tr>
<td>8.1.3</td>
<td>The Matrix Form Mapping Method</td>
<td>290</td>
</tr>
<tr>
<td>8.2</td>
<td>A Discussion of the Task Force's Evaluation</td>
<td>292</td>
</tr>
<tr>
<td>8.3</td>
<td>Multiple Streams of Development in Type Two Reference Architectures</td>
<td>303</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Independency in PERA's Multistream Development</td>
<td>304</td>
</tr>
<tr>
<td>8.3.2</td>
<td>Dependency in CIMOSA and GRAI-GIM within their Four View Development</td>
<td>307</td>
</tr>
<tr>
<td>8.3.3</td>
<td>Phase Dependencies and the Four View Approach</td>
<td>308</td>
</tr>
<tr>
<td>8.3.4</td>
<td>An Explanation for the Different Development Patterns</td>
<td>311</td>
</tr>
<tr>
<td>8.3.5</td>
<td>&quot;Divide et impera (Divide and rule)&quot;</td>
<td>316</td>
</tr>
<tr>
<td>9.</td>
<td>CONCLUSIONS AND RECOMMENDATIONS</td>
<td>319</td>
</tr>
<tr>
<td>9.1</td>
<td>Conclusions</td>
<td>319</td>
</tr>
<tr>
<td>9.2</td>
<td>Recommendations for Further Research</td>
<td>322</td>
</tr>
<tr>
<td></td>
<td>LIST OF REFERENCES</td>
<td>325</td>
</tr>
</tbody>
</table>
APPENDICES

Appendix A: A Comparative Discussion of The Book of Change and the Purdue Enterprise Reference Architecture ............... 339
Appendix B: Glossary of the Purdue Reference Architecture and its Associated Methodology ....................... 343
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Benefits of the Use of an Architecture</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>Basic Classes of Tasks of the Information Architecture of the Enterprise</td>
<td>59</td>
</tr>
<tr>
<td>3.2</td>
<td>Expansion Principles of the Purdue Architecture</td>
<td>69</td>
</tr>
<tr>
<td>3.3</td>
<td>Basic Principles of the Purdue Methodology for Enterprise Integration</td>
<td>78</td>
</tr>
<tr>
<td>3.4</td>
<td>Areas of Interest on the Architecture Framework for Discussing Development and Implementation Aids for Programs and Enterprise Studies</td>
<td>84</td>
</tr>
<tr>
<td>5.1</td>
<td>A Suggested Set of Programming and Communications Interface Standards</td>
<td>153</td>
</tr>
<tr>
<td>5.2</td>
<td>Definitions of Program Interfaces Involved in an Enterprise Integration Program</td>
<td>184</td>
</tr>
<tr>
<td>6.1</td>
<td>FR-(1) and DP-(1)</td>
<td>203</td>
</tr>
<tr>
<td>6.2</td>
<td>FR-(1-1) and DP-(1-1)</td>
<td>204</td>
</tr>
<tr>
<td>6.3</td>
<td>FR-(1-1-1) and DP-(1-1-1)</td>
<td>205</td>
</tr>
<tr>
<td>6.4</td>
<td>FR-(1-2) and DP-(1-2)</td>
<td>206</td>
</tr>
<tr>
<td>6.5</td>
<td>FR-(1-2-1) and DP-(1-2-1)</td>
<td>207</td>
</tr>
<tr>
<td>6.6</td>
<td>FR-(1-2-2) and DP-(1-2-2)</td>
<td>209</td>
</tr>
<tr>
<td>6.7</td>
<td>FR-(1-3) and DP-(1-3)</td>
<td>210</td>
</tr>
<tr>
<td>6.8</td>
<td>FR-(1-4) and DP-(1-4)</td>
<td>212</td>
</tr>
<tr>
<td>6.9</td>
<td>FR-(1-4-1) and DP-(1-4-1)</td>
<td>213</td>
</tr>
<tr>
<td>6.10</td>
<td>FR-(1-4-2) and DP-(1-4-2)</td>
<td>214</td>
</tr>
</tbody>
</table>
Table

6.11 FR-(1-5) and DP-(1-5) ........................................ 215
6.12 FR-(1-6) and DP-(1-6) ........................................ 216
6.13 FR-(2) and DP-(2) ............................................... 217
6.14 FR-(2-1) and DP-(2-1) ........................................... 218
6.15 FR-(2-1-1) and DP-(2-1-1) ...................................... 219
6.16 FR-(2-1-2) and DP-(2-1-2) ...................................... 220
6.17 FR-(2-2) and DP-(2-2) .......................................... 223
6.18 FR-(2-2-1) and DP-(2-2-1) ..................................... 224
6.19 FR-(2-3) and DP-(2-3) .......................................... 226
6.20 FR-(2-4) and DP-(2-4) .......................................... 228
6.21 FR-(3) and DP-(3) .................................................. 230
6.22 FR-(3-1) and DP-(3-1) .......................................... 231
6.23 FR-(4) and DP-(4) .................................................. 232
6.24 What is PERA and its Associated
  Purdue Methodology ............................................. 233
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 The &quot;Wind-Chime&quot; — A Presentation of the Phases (or Layers) of the Purdue Architecture in Terms of the Types of Tasks which Occur throughout the Life Cycle of the Architecture</td>
<td>35</td>
</tr>
<tr>
<td>3.2 Definition of the Components of the Concept and Definition Layers for the Manufacturing Case</td>
<td>37</td>
</tr>
<tr>
<td>3.3 Developing the Relationships of the Several Sub-Architectures of the Purdue Architecture for Manufacturing Systems</td>
<td>41</td>
</tr>
<tr>
<td>3.4 Development of an Enterprise Integration Program as Shown by the Purdue Architecture (Phases and Layers of the Program)</td>
<td>42</td>
</tr>
<tr>
<td>3.5 Introduction of the Automatability, Humanizability and Extent of Automation Lines to Define the Three Implementation Architectures</td>
<td>47</td>
</tr>
<tr>
<td>3.6 Definition of Human Task Types in the Several Areas Represented in the Human and Organizational Architecture</td>
<td>51</td>
</tr>
<tr>
<td>3.7 Definition of the Task Module</td>
<td>52</td>
</tr>
<tr>
<td>3.8 Functional Composition and Decomposition of the Task Modules of Purdue Architecture</td>
<td>55</td>
</tr>
<tr>
<td>3.9 The Purdue Scheduling and Control Hierarchy for an Industrial Plant (Discrete Manufacturing Assumed)</td>
<td>60</td>
</tr>
<tr>
<td>3.10 Mapping of the Purdue Reference Model onto the Purdue Reference Enterprise Architecture</td>
<td>61</td>
</tr>
<tr>
<td>3.11 General Task Module with Implementation</td>
<td></td>
</tr>
</tbody>
</table>
Considerations  ....................................  63
<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.12 Information Task Module with Implementation Considerations</td>
<td>64</td>
</tr>
<tr>
<td>3.13 Manufacturing Task Module with Implementation Considerations</td>
<td>65</td>
</tr>
<tr>
<td>3.14 Human and Organizational Task Module with Implementation Considerations</td>
<td>66</td>
</tr>
<tr>
<td>3.15 Expert System or other Rule-Based Module</td>
<td>67</td>
</tr>
<tr>
<td>3.16 Further Explanation of the Definition of the Generic Enterprise by the Purdue Architecture</td>
<td>71</td>
</tr>
<tr>
<td>3.17 Further Explanation of the Definition of the Generic Enterprise by the Purdue Architecture</td>
<td>72</td>
</tr>
<tr>
<td>3.18 The Modified Evolution Diagram for the Generic Enterprise</td>
<td>75</td>
</tr>
<tr>
<td>3.19 Definition of Human and Organizational Architecture for an Information Services Enterprise</td>
<td>77</td>
</tr>
<tr>
<td>3.20 The Purdue Guide to Master Planning and Implementation for Enterprise Integration Programs and Master Plan Information Flow</td>
<td>81</td>
</tr>
<tr>
<td>3.21 Abbreviated Sketch to Represent the Structure of the Purdue Enterprise Reference Architecture in Relation to Table 3.4</td>
<td>83</td>
</tr>
<tr>
<td>4.1 The CIMOSA-Cube: An Overview of the CIMOSA Architectural Framework</td>
<td>101</td>
</tr>
<tr>
<td>4.2 Overview of Basic Constructs and their Relationships in the CIMOSA Function View</td>
<td>103</td>
</tr>
<tr>
<td>4.3 Descriptive Parts of CIMOSA Enterprise Function</td>
<td>104</td>
</tr>
<tr>
<td>4.4 The CIMOSA Template for Specification of the Functional Part of the CIMOSA Enterprise Function</td>
<td>107</td>
</tr>
<tr>
<td>4.5 Sketch of an Example Manufacturing System</td>
<td>113</td>
</tr>
<tr>
<td>4.6 Processes and Activities for the Example Manufacturing System</td>
<td>114</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>4.7</td>
<td>115</td>
</tr>
<tr>
<td>4.8</td>
<td>116</td>
</tr>
<tr>
<td>4.9</td>
<td>119</td>
</tr>
<tr>
<td>4.10</td>
<td>123</td>
</tr>
<tr>
<td>4.11</td>
<td>124</td>
</tr>
<tr>
<td>4.12</td>
<td>127</td>
</tr>
<tr>
<td>4.13</td>
<td>131</td>
</tr>
<tr>
<td>4.14</td>
<td>133</td>
</tr>
<tr>
<td>4.15</td>
<td>134</td>
</tr>
<tr>
<td>4.16</td>
<td>137</td>
</tr>
<tr>
<td>4.17</td>
<td>138</td>
</tr>
<tr>
<td>4.18</td>
<td>139</td>
</tr>
<tr>
<td>4.19</td>
<td>141</td>
</tr>
<tr>
<td>4.20</td>
<td>142</td>
</tr>
<tr>
<td>5.1</td>
<td>147</td>
</tr>
<tr>
<td>5.2</td>
<td>149</td>
</tr>
<tr>
<td>5.3</td>
<td>151</td>
</tr>
<tr>
<td>5.4</td>
<td>155</td>
</tr>
<tr>
<td>5.5</td>
<td>161</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>5.6</td>
<td>162</td>
</tr>
<tr>
<td>5.7</td>
<td>163</td>
</tr>
<tr>
<td>5.8</td>
<td>164</td>
</tr>
<tr>
<td>5.9</td>
<td>165</td>
</tr>
<tr>
<td>5.10</td>
<td>166</td>
</tr>
<tr>
<td>5.11</td>
<td>167</td>
</tr>
<tr>
<td>5.12</td>
<td>173</td>
</tr>
<tr>
<td>5.13</td>
<td>174</td>
</tr>
<tr>
<td>5.14</td>
<td>175</td>
</tr>
<tr>
<td>5.15</td>
<td>176</td>
</tr>
<tr>
<td>5.16</td>
<td>177</td>
</tr>
<tr>
<td>5.17</td>
<td>178</td>
</tr>
<tr>
<td>5.18</td>
<td>181</td>
</tr>
<tr>
<td>5.19</td>
<td>182</td>
</tr>
<tr>
<td>5.20</td>
<td>183</td>
</tr>
<tr>
<td>5.21</td>
<td>186</td>
</tr>
<tr>
<td>5.22</td>
<td>189</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>6.1 Two Function Requirements of the Opener are Independent</td>
<td>195</td>
</tr>
<tr>
<td>6.2 Hierarchy of Functional Requirements and Design Parameters for Purdue Methodology for Enterprise Integration</td>
<td>199</td>
</tr>
<tr>
<td>6.3 Development of Enterprise Requirements</td>
<td>208</td>
</tr>
<tr>
<td>7.1 The Waterfall Model</td>
<td>239</td>
</tr>
<tr>
<td>7.2 The Spiral Model</td>
<td>240</td>
</tr>
<tr>
<td>7.3 Subdivision of the Enterprise into Separated Enterprise Business Entities</td>
<td>245</td>
</tr>
<tr>
<td>7.4 Relationships between the Strategic Management Enterprise Entity and the Engineering or Manufacturing Enterprise Entities</td>
<td>249</td>
</tr>
<tr>
<td>7.5 A Genealogical Tree of Enterprise Entities</td>
<td>250</td>
</tr>
<tr>
<td>7.6 Users' Input to a Product Development Process Carried Out by the Manufacturer of the Product</td>
<td>253</td>
</tr>
<tr>
<td>7.7 The Purdue Guide as a Product of # 2 Enterprise Entity</td>
<td>257</td>
</tr>
<tr>
<td>7.8 Development Duties of # 1 and # 2 Enterprise Entities</td>
<td>258</td>
</tr>
<tr>
<td>7.9 What is CIM?</td>
<td>261</td>
</tr>
<tr>
<td>7.10 Applicability of Prof. Suh's Axioms and PERA's Lifecycle</td>
<td>263</td>
</tr>
<tr>
<td>7.11 Traditional and Modern Techniques of Decision Making</td>
<td>267</td>
</tr>
<tr>
<td>7.12 How Management Mandates, Requirements, etc., Interface Functional and Detailed Design Practices and Results</td>
<td>269</td>
</tr>
<tr>
<td>7.13 Representation of Possible Program Iteration as Shown in PERA</td>
<td>277</td>
</tr>
<tr>
<td>7.14 An Unstable Rework Loop will Invoke More and More Changes</td>
<td>281</td>
</tr>
<tr>
<td>8.1 A Right View of CIMOSA Building Blocks</td>
<td>287</td>
</tr>
<tr>
<td>Figure</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>8.2 Development of the CIMOSA Basic Skeleton Diagram</td>
<td></td>
</tr>
<tr>
<td>8.3 CIMOSA Architecture Mapped Against CIMOSA Basic Skeleton Diagram</td>
<td></td>
</tr>
<tr>
<td>8.4 The Evaluation Matrix</td>
<td></td>
</tr>
<tr>
<td>8.5 Symbology Used for Completing the Matrix form Evaluation</td>
<td></td>
</tr>
<tr>
<td>8.6 The Matrix Evaluation Form for CIMOSA</td>
<td></td>
</tr>
<tr>
<td>8.7 The Matrix Evaluation Form for GRAI-GIM</td>
<td></td>
</tr>
<tr>
<td>8.8 The Matrix Evaluation Form for Purdue</td>
<td></td>
</tr>
<tr>
<td>8.9 A Possible Hierarchical Structure of the Architectural Formalism Proposed in this Study</td>
<td></td>
</tr>
<tr>
<td>8.10 How the Purdue Methodology Handles the Concepts of Generic, Partial and Particular Models as Espoused by CIMOSA</td>
<td></td>
</tr>
<tr>
<td>8.11 Multiple Stream Development of Program Products in PERA</td>
<td></td>
</tr>
<tr>
<td>8.12 Comparison of the Semantics for CIMOSA and GRAI-GIM</td>
<td></td>
</tr>
<tr>
<td>8.13 GRAI-GIM: Working Streams in Four Views to Three Technical Domains</td>
<td></td>
</tr>
<tr>
<td>8.14 Task Module as Basic Unit in PERA’s Multiple Stream Development</td>
<td></td>
</tr>
<tr>
<td>8.15 Modeling Sequence of the GIM Method</td>
<td></td>
</tr>
<tr>
<td>8.16 Complexity of the Four-View Development Pattern</td>
<td></td>
</tr>
</tbody>
</table>
1. INTRODUCTION AND PURPOSE

1.1 Introduction

Modern technological developments, especially in the information technology fields, have provided tremendous opportunities for corresponding developments in industries of all types. At the same time, these same advances have caused the complexity of all kinds of enterprise systems to grow to such a degree that the former common classical and/or conventional means of management and operation can no longer effectively support the emerging growth of highly sophisticated combinations of people, machines and technologies and their functions now present in such systems.

One potential method of countering this increasing complexity and the resulting difficulty of management and control of an enterprise is to enlist the use of the computer to aid the personnel involved in these tasks. Computers will collect the very large amounts of required data and integrate and reduce it to manageable proportions for the required human consideration and decision making. Where the necessary applications of the results of this data analysis require only routine decisions, the computer systems can even make these independently to further reduce the workload of associated personnel. Such combinations of personnel and computers have recently been called by the term, enterprise
integration. One branch of this latter field is entitled computer integrated manufacturing or CIM.

However, development of the necessary computer based systems to carry out the above tasks is itself a very large and difficult undertaking, and major efforts are underway to explore new and better ways of accomplishing such work. One method to reduce this difficulty is to develop a methodology to organize globally and simplify the tasks involved as much as possible. A type of such a design system now being extensively studied world-wide is that of an Enterprise Reference Architecture and its associated Methodology [1].

At present, there are three major architectures of the above type being evaluated. They are (1) the Computer Integrated Manufacturing Open System Architecture (CIMOSA) which was prepared by the European Computer Integrated Manufacturing Architecture Consortium, AMICE (a backward acronym) under ESPRIT (European Strategic Program in Information Technology) of the European Community (EC) [2–15]; (2) the GRAI Integrated Methodology (GRAI-GIM) developed by the GRAI Laboratory of the University of Bordeaux in France [16–18]; and (3) the Purdue Enterprise Reference Architecture and associated Methodology developed by the Purdue Laboratory for Applied Industrial Control (PLAIC) of Purdue University [1, 19, 20].

The Purdue Enterprise Reference Architecture and Methodology (PERA) [20] was first developed by PLAIC starting in December 1990 and first published in December 1991 [19]. Despite its recent appearance, it has already been shown to be the forerunner of a whole new type of reference architecture and the most complete of this type to date [1].
The above report [1] of the IFAC/IFIP Task Force on Architectures for Integrating Manufacturing Activities and Enterprises has characterized the Purdue Architecture as, "—an informally described means for leading a users application group through all of the phases of an enterprise integration program from initial concept through use to final plant obsolescence." [6, p. V-1-8].

This report also describes the Purdue Architecture as follows:

• "The most important contribution of the Purdue Architecture is its detailed and practical proposal as to how to go about integrating enterprises. Some aspects (like manufacturing control) are dealt with in great detail." [1, p. IV-2-110]

• "The model and description categories of the Purdue Architecture are defined in a finer detail than in the other architectures. Furthermore, there are detailed methods provided for producing all these descriptions provided one has selected a specific modeling method (or English) to capture these descriptions." [1, p. IV-2-110]

• "As an informal description it is the easiest to understand by non-computer science educated users. This is especially true because of its easy-to-grasp graphical presentation of its overall structure and layered phases for program development. Its associated methodology is also the most complete, particularly in its discussion of the planning phases of an integration program. The organization of the Purdue Architecture is also important in that its structure
approntimates the way many plant workers think of their factories and their operations." [1, p. V-1-8]

• "— It appears that the Purdue Architecture develops to great detail the architecture proper, while other approaches, like CIMOSA [5-15] and GRAI-GIM [16-18], provide modeling framework and modeling methods to fill in the elements of the Purdue Architecture." [1, p. IV-2-86]

On the other hand, this publication again comments on the lack of formality in the Purdue Architecture as follows:

"There is no commitment in the Purdue Architecture as to how (or whether) to develop formal models; at the moment it is up to the team creating the master plan as to what way of description suits best the given individual system's architecture." [1, p. IV-2-86]

As a generic guideline for enterprise integration of a very broad scope, it is very important for an enterprise reference architecture and its methodology to define each step of an enterprise development program by means which are precisely based on widely accepted instructive structures. Formalism as it will be used in this study is defined as follows:

Definition 1.1 (Formalism) Formalism is the development of engineering and related concepts, definitions, proposals, etc., in a way that can lead to future standardization by recognized standards writing bodies. This means that all definitions can be agreed to by all parties concerned, and all concepts, etc., are written and approved by all parties involved. All modeling formalisms used are from already standardized techniques if available and proper.
Formality demands the use of a rigorously defined Syntax (form) and Semantics (meaning) to allow capture of all details of tools, techniques and knowledge in the studied field in computer executable form to the maximum extent possible. The computer executability used will be compatible with current "open systems" in computer usage today.

It is the purpose of this study to undertake an investigation of the potential for supplying a more formal underpinning for the Purdue Enterprise Reference Architecture and its Methodology. The following section will present the details to be explored in this study.

1.2 Problems and Approaches

There are major requirements for formalism(s) which arise during development of an enterprise reference architecture and its associated methodology. All of them are related to the problems which may occur with information presentation or communication during the formation and execution of enterprise development programs using the architecture and methodology. The more completely an enterprise reference architecture can specify such a program and the associated components of the enterprise, the more requirements its inherent formalism must meet in order to present the knowledge required without ambiguity.

1.2.1 Formalism Involved in Information Presentation between Program Participants

The Purdue Enterprise Reference Architecture requires that the goals and aspirations which the higher management of the enterprise has concerning the expected outcome of the
proposed endeavor must be reflected in the requirements of the development program for the subject enterprise. That is to say, PERA presents a complete framework for the proposed program which describes every step of an enterprise development program starting with its very initial concepts in the eyes of the higher management.

When the business strategies, mission, vision and values of the planned enterprise as decided by the management are transformed into the functional requirements for the program, the exact meanings of the expression of the information, i.e., its semantics, must be totally available to the planners. The same conditions must also be met when information on the program development is presented to higher management for program review. All the information included in the program must be completely understood by both the customer of the program, i.e., the management, and the technical professionals carrying out the planning work, i.e., the integration working team(s).

Because the totality of the knowledge involved in any enterprise development program is too large to be grasped by any single person, a multidisciplined working team or even several such teams will be involved during the period of program development as pointed out by the discussion accompanying every major reference architecture. All communications between different disciplines and between different working teams must maintain the consistency of the information involved. Thus, a well defined semantics is also required. Because of the heavy involvement of technical information, an established syntax defined for both graphical and textual presentations will also be needed for clear understanding.
1.2.2 Formalism Involved in Development of an Enterprise Reference Architecture and Methodology

The importance of formalism should not be forgotten in the further development of the enterprise reference architectures and methodologies themselves. It must be noted that proposed requirements for reference architectures and their methodologies have been defined in several places in the open literature [1, 3, 7, 9, 16, 19, 21, 22]. Also, each major architecture may have its own features which are considered by the developer to be better than the counterparts of the others. Thus, a combination of the features of the currently available architectures may lead to a truly universally acceptable architecture and methodology.

However, in such a new area, the developers of reference architectures and methodologies are usually working in very different academic and industrial backgrounds. Thus, there is no current common terminology for use as a basis for communication between them. A major task therefore is to develop a dictionary or glossary accepted by all to make it possible for them to establish a commonly acceptable system.

One of the major issues in the development of reference architectures is the introduction of computerized modeling tools for system specification and design. This is considered to be a major strong point of CIMOSA [1]. It was their formal specification method which was accompanied by these formal computerized modeling tools which was the first push for the formalization of reference architectures. Although the importance of the computerized tools has been widely accepted, the relationships between the necessity for a total formalism for a reference architecture and the formalism
necessary for its computerized tools remain to be further investigated.

1.2.3 Formalism Involved in Information Transformation through Interfaces

The Purdue Enterprise Reference Architecture recognizes that in any enterprise development program, the human and organizational system is equally as important as the information system and manufacturing equipment components. Accordingly, three Implementation or Physical Sub-Architectures are introduced into its enterprise program. They are the Information System Architecture, the Human and Organizational Architecture and the Manufacturing Equipment Architecture (if the subject enterprise is a manufacturing enterprise). Such partitions require the use of physical interfaces between each of them and the others. Such interfaces also raise the necessity of rigid protocols for the information transfer across these interfaces.

The Purdue Methodology and Architecture recognizes such interfaces not only as interconnections between elements within each of the Implementation Architectures or between these Implementation Architectures themselves, but also those across the boundaries between this Architecture and the outside world or environment. These interfaces are the electronic and/or mechanical interconnections which permit two or more physical (or human/organizational or both) modules to carry out those functions of the two or more modules which are interconnected informationally.

Below are listed some of identified interfaces in the Purdue Architecture, where standardization is expected for the interfaces and their associated protocols:
• Engineer or Programmer − Computer
• Plant Worker or Operator − Computer
• Computer − Computer
• Non-Computer Machine − Computer
• Etc.

Human related interfaces in the Enterprise Integration are very crucial and have unique features of their own. Human interaction with the Information Systems Architecture always involves the transfer of data or information by means of an interface which allows the machine and human to interpret the results of the operations of the other in passing information flow across the interface between the involved entities. The term information "interpretation" is used here in contrast to the term information "execution", since the human entity on one side of the interface can not directly handle the electronic based information flow of the Information Systems components. Hence, the existence of the interface is absolutely necessary.

This may not be the case when the human functions as a machine component, such as when they are involved with the Manufacturing Equipment Architecture, because here the human can directly handle the material flow of the manufacturing process. Therefore, the definition of an interface would not be necessary between a human operator and a manufacturing machine.

1.2.4 Basic Definitions Concerning Interfaces

The problems listed above from Section 1.2.1 to Section 1.2.3 indicate that the definition of an interface which follows should cover standardization considerations arising
in the functional design or specification phase and in the detailed design phase of a project program in enterprise integration.

Definition 1.2 (Interface) An interface is a shared boundary between two entities.

In the Functional View of PERA, the interface is a special functional module which performs one or more of the following tasks between two entities, two other functional modules which want to exchange something between them, as follows:

- Transfer and transformation as necessary for information flow;
- Transfer and transformation as necessary for energy flow; and
- Transfer and transformation as necessary for material flow.

In the Implementation Architectures, the entities in Definition 1.2 may be considered to be the computer hardware, computer software, non-computer machines, or humans involved. The interface between them may be manufacturing or computer equipment, or simply electrical circuits, or a combination of all or part of these.

The specifications involved in the detailed design of an interface may include:

- Communication interface design

Communication interface design considers the specifications for the information transfer and any needed transformation involved between the two entities on either side of the interface. In the context of PERA, this interface may be located in the
Information System Architecture or in the Human and Organizational Architecture or form the boundary between them. These involve the specifications and designs for the semantics, syntax and/or protocols employed in the communication between the two entities. In the case of information being presented to humans where both the information provider and receiver are humans, the design would probably consider the terminology (language semantics), graphical semantics and syntax. To facilitate integration, standardized protocols and other communication standards must be followed in this communication interface design.

- Manufacturing equipment interface design

Manufacturing equipment design considers the specifications for the physical flow of material and energy across an interface between the two pieces of manufacturing equipment. The design of a heat exchanger and the design of in-process storage may be typical examples. In order to integrate the designs, as well as the future operations, of the physical equipment involved, current standards must be followed in the design process as much as possible.

Another important concept related to communications standardization is the concept of protocol. In terms of communications, a protocol can be defined as:

Definition 1.3 (Protocol) A protocol is the set of rules that define the physical, electronic signal and coding standards for physical interfaces in communications.
A protocol does for the physical interface what semantics and syntax do for human communications. It assures that the interpretation for the signal is the same on both sides of the interface.

It seems that if the concepts of interface and protocol were generalized, all the considerations related to formalization of PERA could be traced to the interface standardization and/or the protocol standardization. Establishment of the minimum formalism needed to enhance understandability of the architecture and its methodology, and therefore to facilitate their universal acceptance, may possibly be solved by standardization alone (see Chapter 5 for further extensive discussions of this subject of interfaces).
2. BACKGROUND: WHAT ARE ENTERPRISE REFERENCE ARCHITECTURES AND WHY DO WE NEED THEM

World-wide industry today is under intense competitive pressures. Major changes are being experienced with respect to resources, markets, and product strategies, and in both manufacturing and information technologies. As a result of international competition, only the most productive and cost-effective companies will survive. The enterprises involved are thus faced with the needs to optimize the way in which they function in order to achieve the best possible performance within the constraints imposed on them. This task is difficult in terms of understanding the nature of the problem as well as the most effective solution strategies, and in terms of forming and implementing plans that develop from this understanding. In order to carry out this task, enterprise integration is becoming a widely-accepted approach to attack the problem and to fulfill the desired accomplishments.

2.1 Needs of Enterprise Integration

In order to understand the needs for enterprise integration, one must understand the business needs which are challenging companies of all kinds in a highly competitive business environment. The "pushing force of newly developed and emerging technologies" on one hand, and the "pulling force
of new requirements and new problems” [23] on the other hand have left little room for those who do not accept the two-sided challenge.

Modern information technologies have made possible large and complex enterprises and their development programs. Advanced information systems are also connecting together, in one way or another, thousands of small businesses electronically. Access to commercial information on products, technologies and marketplace factors has never been so handy before. In order to stay in business and increase their profit margins, companies have to be able to catch the possibilities revealed by the development of advanced information technologies in order to achieve a high level performance such as a fast response to customers requirements for high quality products and services; as well as to take advantage of low cost, flexible and productive production systems. On the other hand, driving forces from inside the companies are also forcing a higher level of overall performance in order to reduce the company’s environmental impact and to meet the higher standards of job safety and job satisfaction of their employees.

The use of integrated manufacturing carried out with the aid of computers, often called computer integrated manufacturing (CIM), has been seen by many as the means by which much of the above could be accomplished. CIM is usually considered to be the process of integration of all of the elements involved in manufacturing. It is usually confined to purely technical integration by means of computerized information techniques. However, efforts in this direction have shown that such a narrow outlook is not really satis-
factory for companies in achieving the high level performance required above [19, 23–27].

In early integration attempts, for example, the comput- erization of offices, factory operations and engineering design tools alone only resulted in nonintegratable “islands of automation” rather than the originally desired integrated enterprise as a whole. This was caused by technically attacking the integration problem piecemeal and because the use of only one single communication and interface discipline was not enforced during the development. A seemingly good technical remedy for the integration approach might be the arrangement of one global or top-down integration project. However, if carried out as one single massive project, this is frequently beyond the resource capabilities of even the largest companies. Often there is no ready method available to break such a large project into suitable sub-units.

Besides, this narrow focus may not take other business critical success factors such as economic, social, human and organizational aspects into account. These factors are no less important than the purely computer-based technologies and should be coordinated with and fully integrated into the information integration process itself [16, 19, 23–29].

One of the problems encountered has been the difficulty to measure the success of a project. Even when successful technologically, the resulting systems often received little acceptance by their operational and administrative staffs and therefore failed. This lack of ownership was generally due to noninvolvement of these personnel in systems plan-
ining and development and from the individual's fear of adequacy and/or job security in relation to the new system.

Nowadays, more and more people are realizing that CIM should refer to a global approach in an integrated way to encompass all aspects, methods, means and techniques in order to establish an integrated enterprise, even though each person may still define their own practical approach differently.

2.2 Definitions of Enterprise Integration and Enterprise Reference Architecture

In order to differentiate the broad approach from the narrow one, the term Enterprise Integration, instead of CIM, will be used in this study. It is defined as follows.

Definition 2.1 (Enterprise Integration) Enterprise Integration refers to a global approach to coordinate in an integrated way all the components of an enterprise in order to improve its overall performance.

Modern society is based on a specialization of functions. Correspondingly, in the field of engineering, each individual or organization, ideally, works on what he or she does best and passes the results of that work on to the next person or organization in the process chain. One result of this specialization is that interindividual and interorganizational communication is very crucial to the success of any such process. Traditionally, product specifications and engineering drawings were the means and forms of communication in the process of product design, production and distribution, while structured models and methods such as SADT [30], SSAD [31], IDEF [32], etc., were developed to deal with the
process of physical system analysis, design and implementa-

tion.

Since the advent of CIM and Enterprise Integration, however, the traditional methods mentioned above have not been adequate to handle all the complexity in the field of enterprise integration. This is especially so when the overall project must include not only a description of what the system will be like when the integration is completed but also how it can be achieved. Therefore, a complete integration methodology must be adopted, and within that methodology, a framework, i.e., a Reference Architecture must be defined to specify the general context of the whole integration process as a unifying guideline to all the parties involved and to enhance the communication and coordination between them.

In the field of Enterprise Integration, there is a common agreement that an Enterprise Reference Architecture and its associated Methodology, as the architectural specification and integration methodology, are strategically necessary and crucial for the success of an Enterprise Program [16, 19, 26, 27, 29, 33, 34]. More and more people have realized that, because of the broad approach, Enterprise Integration "can not be bought" [16], and the strategic development decision, must be made "to address the way that organization, people (not to consider human factors in itself), and technology can be used to the satisfy the business needs" which is "long before any detailed design work is undertaken or even requirements specification developed" [26]. Each company must develop its own projects to meet every requirement needed for its business success by following an Enterprise Reference Architecture and its associated Methodology. In this study, the Reference Architecture of
Enterprise Integration, or the Enterprise Reference Architecture}, is defined as:

Definition 2.2 (Enterprise Reference Architecture) The term Enterprise Reference Architecture refers to a generic framework to guide the application of all the disciplines involved in every step of an enterprise integration program or any other enterprise development program throughout its whole life cycle.

As a generic framework, an Enterprise Reference Architecture models systematically the whole life history of an enterprise development program from its first concept in the eyes of the management and the strategic planners who initially develop it, through its subsequent stages of definition, functional design or specification, detailed design, physical implementation or construction, and finally operation through to obsolescence. All of the integration functions and activities involved in the life cycle of the enterprise development program can be mapped upon the architectural framework, and the development of a particular enterprise program can be described in step by step detail.

2.3 Functions of Enterprise Reference Architectures

Although many proposed Enterprise Reference Architectures have been developed [1, 5, 16, 19], an Enterprise Reference Architecture to be valuable should be able to fulfill the following requirements and functions needed for Enterprise Integration or any other system engineering type projects [1, 19]:

- The Enterprise Reference Architecture should explicitly show the evolution of an enterprise development
program or an enterprise undertaking from initial concept through operation inclusively as a complete system. In other words, an architecture or structure of the particular program should be presented to illustrate the life history or life cycle of the program developing the integrated enterprise. The effect of adaptation or change in the system should be modeled so as to reflect the changing enterprise environment, constraints, system implementations and operations in an evolutionary manner. That is, the Architecture should be able to be open to describe all aspects of the full system life cycle.

- The Enterprise Reference Architecture should be able to guide all the parties from each of the different disciplines involved in the course of an enterprise development program in a wide range of industries. The guidelines must be able to show every step of the development and all of the transformations included in the steps in the integration procedures to coordinate efforts between the multidisciplinary approaches used as a whole. Meanwhile, limitations imposed by a specific design on the generic requirements for the system integration should be visible. Related tools and means available for integration procedures should also be presented. In other words, the architecture should be open to the use of all pertinent disciplines.

- The Enterprise Reference Architecture should incorporate several different views where they are necessary to completely describe an enterprise. All the aspects of the program development, such as informa-
tion and control, manufacturing equipment, human and business organization, as well as those interfaces which exist between them and with foreign elements, should be modeled explicitly, functionally and structurally. All the related strategic considerations in integration such as economic questions, decision-making, business processes, etc., should also be reflected in a form suitable for translation into a processable implementation. The Architecture should be open to the description of all the aspects of the system development.

- The Architecture should present a method for the breakdown of all system functions to their inherent generic functions and tasks. These should be accomplished with a minimum number of types of basic building blocks or modules which can be applied to a wide range of manufacturing operations and organizations. Conversely, it should be possible to build up the overall functions from these ultimate basic building blocks. All existing relationships of those entities represented by the blocks for modeling should be shown. The Architecture should be open to all of the methods of achieving modularity.

- The presentation methods used by the Architecture should be descriptive rather than prescriptive to give it the widest possible scope. The Architecture itself should be independent of existing technologies in manufacturing, automation and computer sciences, and independent of any given predetermined realization in terms of system configuration or implementations. Therefore, the specification for the
architecture will be open-ended in its ability to be extended and in its ability to encompass and promote new concepts and technologies, once they become stable, without invalidating current realizations. That is, the Architecture should be open to further technology development.

- The Architecture should provide a common basis for the development of the needs for standards for various aspects of a system and should be able to indicate their relative values. Standardization issues, such as standard evaluation criteria and other modeling methods and tools, arising from integration projects will be identified and incorporated. Existing standards can be included into the system framework where applicable. Guidance for developing new standards will be provided. During system development, details of a particular enterprise or business entity or a proposal can be mapped against a similar existing accepted standard system to point out deficiencies in the existing or proposed system. In line with the standards, a heterogeneous, multi-vendor environment for hardware and software suppliers both for information systems and manufacturing systems should be provided. The Architecture is open to standardization.

The developers of the Purdue Methodology believe [19, 27] that practically, all business enterprises, sooner or later, will require changes (improvements) in their company’s business practices, information handing methods, and manufacturing or process technologies (some or all). Thus, an architecture which can describe enterprise conception,
development, implementation and operation, all at one time, gives the best initial approach to all such contemplated changes. It also explains better than any other tool the relationships of the elements of the Enterprise Integration system. It is thus the glue that holds all aspects of the project together. The overall benefits of the use of an architecture are given in Table 2.1.

It can be seen that an integration architecture, as a reference guideline for the application of enterprise integration, must consider not only the "what" of integration but also the "how" of integration as well. That is, how integration is accomplished in terms of development of the industrial or enterprise system, not just the overall design of a computer system to construct an integrated physical structure.

2.4 Types of Reference Architectures

An architecture can be defined as a description or model of the structure of either a physical or a conceptual entity. Therefore, there are two types of architectures which deal with the integration of manufacturing entities or enterprises [1]. They are,

- The structural arrangement or design of a physical system such as a computer control system, or a communication system of an enterprise;

- The structural arrangement or organization of the development and implementation of a project or a program such as a manufacturing or enterprise integration or other enterprise development program.
It can be seen that Definition 2.2 and all the functions of the Enterprise Reference Architecture as already discussed in this chapter should therefore belong to the second
### TABLE 2.1 BENEFITS OF THE USE OF AN ARCHITECTURE

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An architecture presents a vision of the total enhancement process for a business;</td>
</tr>
<tr>
<td>2</td>
<td>The architecture allows one to readily determine the feasibility of the proposed program of improvement particularly in comparison to other competitive programs;</td>
</tr>
<tr>
<td>3</td>
<td>Similarly, the architecture allows a determination of whether the enhancement program proposed is the best method of attack;</td>
</tr>
<tr>
<td>4</td>
<td>A structure is necessary to the enhancement process to assure:</td>
</tr>
<tr>
<td></td>
<td>- Completeness;</td>
</tr>
<tr>
<td></td>
<td>- Consideration of alternatives and prevent</td>
</tr>
<tr>
<td></td>
<td>- False steps in the process;</td>
</tr>
<tr>
<td></td>
<td>- Misallocation of resources;</td>
</tr>
<tr>
<td></td>
<td>- Etc.</td>
</tr>
<tr>
<td>5</td>
<td>An easy expansion from manufacturing to any enterprise is readily available, for example in the Purdue Architecture, by treating the right-hand side as customer product and service operations, rather than just manufacturing;</td>
</tr>
<tr>
<td>6</td>
<td>Use of the architecture gives assurance that the final result will be a completely integrated system provided the separate projects into which the overall complete program may have been divided comprise the parts of an architecture-guided Master Plan.</td>
</tr>
</tbody>
</table>
type, called Type Two hereafter, in order to fulfill the functions mentioned above. The first type, called Type One hereafter, is actually a subset of the Type Two, since the latter will comprise many physical systems (see Section 2.3).

There are only three major architectures identified to date in the open literature which fall into the category of Type Two above [1]. These are

• The Open System Architecture for Computer Integrated Manufacturing–CIMOSA as developed by the European CIM Architecture Consortium (AMICE [a backward acronym]) under ESPRIT Projects 668, 2422 and 5288 of the European Community. This work was initiated in 1984 (hereafter called CIMOSA) [2-15].

• GRAI-GIM. The GRAI Integrated Methodology as developed by the GRAI Laboratory of the University of Bordeaux in France. This work resulted from production management studies initiated at the GRAI Laboratory as early as 1974. It has taken its current form since about 1984 (hereafter called GRAI-GIM) [16-18].

• The Purdue Enterprise Reference Architecture and the related Purdue Methodology as developed at Purdue University as part of the work of the Industry-Purdue University Consortium for CIM. This latter work started formally in 1989 but bears on The Purdue Reference Model developed starting in 1986 and earlier work of the Purdue Laboratory for Applied Industrial Control (PLAIC) dating back to the mid-seventies (hereafter called Purdue) [1, 19, 20].
While the use of suitable Type Two architecture(s) is vitally important for the continuing growth of the field of Enterprise Integration and for its acceptance as the mode of optimization of the production and operation of manufacturing plants and other enterprises in the future, an Enterprise Reference Architecture will not accomplish the task alone. An accompanying methodology of Enterprise Integration is necessary.

2.5 Needs for Enterprise Integration Methodology

2.5.1 Problems beyond the Enterprise Reference Architecture

The Reference Architecture alone is an abstract framework or outline of the enterprise development steps. For practitioners working on application of Enterprise Integration, it alone cannot answer every application question arising from this field. If the functions of an Enterprise Reference Architecture and Methodology as listed in Section 2.3 are to be performed, the needs listed below must also be satisfied:

- Methods must be developed to make the use of the modeling tools, techniques, architectures, etc., noted earlier as transparent as possible to users from a vast spread of different backgrounds and training. It must not be required that the ultimate users need to develop a computer science background to understand and apply them, i.e., all the theoretical complexity should be hidden from the application practitioner;

- One of the major difficulties in developing and implementing an enterprise integration program is the
tremendous amount of detail required to specify, design and construct such a system. Means must be found to reduce greatly at least the "apparent degree" of the amount of detail required. Possible applicable methods involve such techniques as configuration and parameterization rather than redesign or reprogramming; massive reuse of generic modules from past systems; etc.

2.5.2 Components of an Enterprise Integration Methodology

Referring to the application methodology, Professor Guy Doumeingts has made a very important statement of the needs and components of the Methodology in introducing a description of the GRAI-GIM Methodology developed by his Laboratory (Laboratorie GRAI, Universite Bordeaux 1, Bordeaux, France) [16]. This statement (with slight editing to better fit the current case) is presented here as follows [1]:

"An Integrated Manufacturing System solution cannot be bought off the shelf; each firm must be involved in devising its own, which explains why methodologies must be made available so that CIM systems can be built.

'Designing a CIM system meets with a number of difficulties:

- The system is extremely complex, so some special techniques must be used to understand this complexity in order to efficiently act on it and to define the possible intervention area of every operator related to their skills;
• The system must take into account not only the technical points of view but also the economic, social, and human points of view in an integrated way;

• The knowledge which is necessary to design a system cannot be found in a single person: designing requires team work;

• The initial status of the system is not a matter of chance: this initial status must be taken into account to understand well the specific constraints under which the system operates on the one hand and, on the other hand, to avoid the redesign of those parts of the system which are satisfactory. Because of today's economic situation, this point becomes more and more relevant: many industrial systems must evolve faster and faster but with as low a cost as possible.

'The term "methodology" means a consistent set of components which are:

• A reference model (the architecture as described here) globally and generically showing the structure of the project system to be studied,

• One or more modeling formalisms enabling the build up of the model in order to study and evaluate it,

• A structured approach for the overall program leading step by step from an existing system to a future system taking into account evolution objectives and specific constraints,
• Performance evaluation criteria with which the system can be evaluated in relation to several points of view (economic, reliability, etc.).

'Generally speaking, a structured approach is a set of steps to be followed to solve a problem. Within the framework of an integrated manufacturing system design methodology, the structured approach must cover all of the life cycle of the integration project which is split into states (analysis, design, development, implementation, operation). Every step of the methodology must be precisely defined and based on a standardized project structure giving the set of actors for which the work must be precisely defined as well.

'Defining very precisely all steps which are required to go from concept to specification to implementation is essential. During each step some models are built and checked. The consistency of these models and results for each step must be constantly checked.'"

The above statement illustrates succinctly the industrial requirements which need to be considered in the field of Enterprise Integration in terms of an application methodology. In order to fulfill the requirements proposed by each major architecture candidate [1, 5, 16, 19], and also because of the inclusion of evaluation aspects within the Enterprise Reference Architecture defined in this study (see Section 2.3), the components of the Enterprise Integration Methodology can be defined or listed as follows:
Definition 2.3 (Components of Enterprise Integration Methodology) An Enterprise Integration Methodology consists of:

- A step-by-step integration development application guide; and
- An accompanying Enterprise Reference Architecture which models and illustrates the Methodology involved.

2.5.3 Formalism Required by Methodology

Formalism is a disciplined approach to the development of engineering and related concepts. Likewise, the Methodology and its accompanying Architecture must present a sufficiently disciplined formalism so that complete and ready understandability and certainty of use is improved.

The formalism used with the Methodology and the Architecture may include:

- Standardized or rigidly defined naming conventions;
- Standardized or rigidly defined symbology;
- Standardized or rigidly defined models;
- Standardized or rigidly defined analytical or modeling procedures;
- Standardized or rigidly defined analytical or modeling tools;
- Etc.

The broad approach provided by the Enterprise Reference Architecture and its associated Methodology requires that the formalism defined above be developed and applied to describe and interpret the Reference Architecture and the Methodology, along with the specific enterprise development program, in a complete, unambiguous and consistent way.
This will make it possible for the multidisciplinary approaches incorporated in all integration projects to communicate and associate with each other through all of the kinds of interfaces involved. At the same time, the formalism should help limit the complexity of the system representations. Human comprehension and a universal acceptance will then be facilitated. Such a formality helps assure a full and accurate interpretation, by all discussing parties through the existing interfaces, of all points under discussion.

2.5.4 A Short Review of the Available Type Two Reference Architectures and their Associated Methodologies

Although CIMOSA, GRAI-GIM and Purdue are Type Two Reference Architectures, their different approaches to this solution have emphasized different features of their Methodologies. The following is a quick look at these differences.

2.5.4.1 Purdue — Comprehensive Master Planning and Implementation Program

The Purdue Enterprise Reference Architecture and its associated Methodology represent one of the major approaches to the field of Enterprise Integration. Because of the long-term exposure of its developers to a wide range of industrial applications and their heavy involvement in large engineering programs, this approach emphasizes strategic planning, and the completeness of an enterprise development program. It also requires a full implementation perspective, including both technological and human and organization aspects, for the enterprise development [19, 27]. Among the three Type Two Architectures and their Methodolo-
gies, it has been recognized that the Purdue Enterprise Reference Architecture, as well as its Methodology, is the one which is most broad in its scope, while it also is the most complete and easily understood by industry users with practical and detailed application instructions [1]. Its recognized weakness is a major lack of formality [1].

Because the Purdue Enterprise Reference Architecture and its associated Methodology are the research subject of this study, a special description of them will be given in the next chapter.

2.5.4.2 CIMOSA — Computer Supported Modeling

CIMOSA represents another major approach to the field of Enterprise Integration. Its initial motivation was to promote the application of information technologies in the discrete manufacturing industry [35]. In the CIMOSA official documents, its modeling methodology is described as follows.

"System design using a finite set of generic building blocks for computer supported modeling and computer supported enterprise operation" [3]

CIMOSA provides its own reference architecture to specify its modeling procedure and to guide its users, system organizers and implementors to go through the CIMOSA proposed system life cycle. However, it does not directly support the task of "business objectives determination" [3].

One of the important contributions of CIMOSA is its definition of a generic integrating infrastructure required by their implementation of Enterprise Integration Programs. Another strong point of CIMOSA is that among the three Type Two reference architectures, it has defined the most formal-
ized modeling structure. This is recognized as a means of promotion for the development of computer processable enterprise system models [1]. The current incompleteness of its reference architecture and methodology is CIMOSA's weakness since it does not as yet cover the full life cycle and all implementation aspects of an integration program as does Purdue [1]. A more complete description of CIMOSA is given in Chapter 4.

2.5.4.3 GRAI-GIM — User Oriented System Design

The GRAI-GIM models give a generic description of a manufacturing system focusing on the details of the control part of this system. GRAI-GIM is intermediate between Purdue and CIMOSA in its completeness and its espousal of formalism. In its methodology, the importance of the initialization of the program based on the current system is recognized. In its modeling framework, users are allowed to initially propose requirements for the new system. A set of user oriented modeling tools are developed and presented to promote an ease of understanding by noncomputer science educated users with the degree of formality used [16]. However, like CIMOSA, its architecture and methodology are not very complete in terms of their adherence to a full life cycle and other aspects of an integration program [1]. A more complete description of GRAI-GIM is given in Chapter 4.
3. A DESCRIPTION OF THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE AND ITS ASSOCIATED METHODOLOGY

The Purdue Enterprise Reference Architecture and its associated Purdue Methodology [19, 20, 27, 36] are recognized, as noted in Chapter 1 and Chapter 2, as the most comprehensive of the available architecture candidates in defining a generic architectural specification and integration methodology for enterprise integration or other enterprise development programs. This chapter will give a description of the basics of this Architecture. Note that discussions of the Purdue Enterprise Reference Architecture and its capabilities and applications are all based upon the concepts of enterprise integration (EI) rather than on computer integrated manufacturing (CIM) [19, 27, 36]. Enterprise integration is a much expanded use of the concepts of CIM (applicable to any enterprise). In addition, CIM as a technology has been discredited by many parts of industry because of its narrow scope (applied to the computer and control system only) [36].

3.1 The Purdue Enterprise Reference Architecture

The Purdue Architecture employs both prose and graphical tools to present the structural arrangement of the development and implementation of a Enterprise Development Program. Its layered conceptual graphical presentation may be called
a "wind-chime" after its shape (see Figure 3.1). The following description of the Purdue Architecture will begin with the general life cycle of an integration program in a manufacturing enterprise.

3.1.1 A Life Cycle Presentation on the Purdue Architecture

Enterprises are organizational business entities built to produce goods and/or services in response to customer needs. The Purdue Architecture considers that an enterprise must face two dynamic environments [20]:

- "The external environment characterized by rapidly changing market needs and competition from other vendors serving the same field of endeavor";
- "The internal environment characterized by global objectives, management attitudes, and technological changes."

Therefore, the Purdue Architecture considers the enterprise from two distinctive viewpoints, that of the external environment or business environment as seen by the business user and that of the internal environment or the arrangement of the enterprise's resources, such as manufacturing technologies and information technologies. These two viewpoints are reflected in the conceptual layers of the Architecture shown in Figure 3.1 which presents a simple block diagram of the Purdue Architecture to help in its explanation. The details of Figure 3.1 will be presented below as the discussion proceeds.

In terms of the types of integration tasks which occur in the several different areas of the graphical representation of the Purdue Architecture, Figure 3.1 shows that the
FIGURE 3.1 THE "WIND-CHIME"—A PRESENTATION OF THE PHASES (OR LAYERS) OF THE PURDUE ARCHITECTURE IN TERMS OF
THE TYPES OF THE TASKS WHICH OCCUR THROUGHOUT
THE LIFE CYCLE OF THE ARCHITECTURE
layered diagram starts with the identification of the Enterprise Integration Business Entity (EBE) under the conditions of both the external environment and the internal environment. This considers the future business use of the integration project to be proposed and how the management identifies the necessity of the initiation of the project or program. This leads to the Concept Layer of the Purdue Architecture where a description of the management’s mission, vision and values for the entity plus any further philosophies of operation or mandated actions, such as choice of processes, vendor selection, etc., will be presented. Figure 3.1 relates to a manufacturing plant specifically. The expansion to the application to any enterprise entity will be shown later in this report.

From the above description, the operational policies of the company for this business entity needed to carry out the mission, vision and values of the entity will then be derived for all the units and areas of potential concern. They are then further classified into two functional categories, one for information and control related policies and another for manufacturing related policies (see Figure 3.2). These two categories complete the Concept Layer.

Following this, a series of functional analyses of the project or program can be carried out in the next layer corresponding to the two viewpoints noted above. In a manufacturing plant, the above prescription and selection by management of possible options leads to the definition of operational or functional requirements for the plant, i.e., the Definition Layer of the Purdue Architecture, where the statement of requirements for all equipment and for the methods of operation, etc., for these equipments will be
FIGURE 3.2 DEFINITION OF THE COMPONENTS OF THE CONCEPT AND DEFINITION LAYERS FOR THE MANUFACTURING CASE
established, and all the related functional decomposition and composition will be performed.

All the analyses in the Definition Layer of the Purdue Architecture are independent of the specific method of actual implementation of the task or function ultimately chosen, no matter whether it is machine- or human-based. The Purdue Architecture considers that functionally, all tasks can be initially defined without reference to their method of implementation, i.e., it makes no difference whether they are conducted by humans or machines/computers, etc. All of these latter considerations are implementation details. Therefore, any discussion of the particular functions carried out can be postponed in the system development until after all tasks and functions are defined.

At the Definition Layer, based on the requirements established, a bottom-up functional approach will be carried out starting from detailed task definition and collection. The Purdue Architecture defines each of its task modules as used here as:

"The lowest level of functional decomposition of an enterprise which usually corresponds to the work of a single person or machine at a point in time."

and defines each function as:

"A group of tasks which can be classified as having a common objective within a company."

As shown in Figure 3.2, the tasks become collected into functions, and these building block modules in turn can be connected into a functional network of information flow on the left hand side, or of material and energy flow on the right hand side. These latter then form the Information
Functional Network or the Manufacturing Functional Network respectively.

The Specification Layer, where the implementation considerations start, immediately follows the Definition Layer. If the future system were a totally automatic one and no humans were involved such as a automatic space station, the two functional networks defined in the Definition Layer would translate here directly into two unmanned implementation architectures: the Information Architecture and the Manufacturing Architecture (see Figure 3.3).

However, because of the necessity of human involvement in any implementation, the first need at the Specification Layer is to define which tasks, either informational or manufacturing, will be fulfilled by people. In other words, the place of the human in both the Information Architecture and the Manufacturing Architecture will be defined. Even though the two types of the tasks to be carried out by humans are different, the economical, social, cultural and educational requirements of humans are common, and so are their physical and psychological capabilities. Thus, all human-based tasks can be considered together. As a result, the two functional networks are actually converted into three implementation architectures as follows (see Figure 3.2):

- The Information Systems Architecture (the computers, communication equipment, interfaces, database facilities, etc.);
- The Human and Organizational Architecture;
• The Manufacturing Equipment Architecture (the processing and material and energy handling equipment, etc.).

All of these Architectures are sub-architectures of the Purdue Enterprise Reference Architecture itself. They are called architectures because they themselves form frameworks for extensive sets of tools, models, etc., for the development of their own contributions to the Enterprise Integration program under study.

The following are two important views which relate to the development of the architectures described above (see Figure 3.4):

• Functional View — that collection of task modules (including their interconnectivity) which describes and illustrates the functions assigned to a business entity and their relationship to each other. The Functional View is reflected in the development of the functional networks in the Definition Layer. The Functional View comprises a set of generic tasks carried out by all, or at least most, Enterprise Integration Business Entities. Management and government directives, policies, mandates, standards, etc., will not affect what tasks are carried out but will only affect how they are carried out. Thus they are not vitally important at this stage, i.e., the directives, etc., and should only take effect when a particular implementation methodology is considered later after the Functional View is complete.

• Physical or Implementation View — a collection of the human organizations and of the physical hardware
and software used for carrying out all or part of the functions described and illustrated by the Functional View of a business entity. This Physical or Implementation View is reflected by the three
FIGURE 3.3 DEVELOPING THE RELATIONSHIPS OF THE SEVERAL SUB-ARCHITECTURES OF THE PURDUE ARCHITECTURE FOR MANUFACTURING SYSTEMS
Figure 3.4 Development of an Enterprise Integration Program
As shown by the Purdue Architecture
(Phases and Layers of the Program)
FIGURE 3.4 (CONT.) THE LATER PHASES IN ENTERPRISE INTEGRATION SYSTEM EVOLUTION AND THEIR ACTIVITIES IN RELATION TO THE PURDUE ARCHITECTURE
architectures starting from the Specification Layer through the rest of the layers in the Purdue Architecture. In contrast to the Functional View, in the Implementational View, management and government directives, policies, mandates, standards, etc., are now vitally important since they affect the how of the implementation of the Enterprise Integration functions.

As a result of the establishment of the three implementation architectures at the Specification Layer, the integration of all of the functions of an integration program with considerations of implementation can be properly planned and specified in the form of a Master Plan. The Master Plan will provide to the integration program the further necessary coordination and integration actions. Thus the actual implementation of such an integration may be broken up into a series of coordinated projects, any and all of which are within the financial, physical and economic capabilities of the company. These can then be carried out as these resources allow as long as the requirements of the Master Plan are followed. When these projects are completed, the integration desired will also be complete.

After the Specification Layer, the following life history of the Implementation View will continue through its Detailed Design Layer, Manifestation Layer, and Operation Layer, and finally to its obsolescence. It can be seen that although the Purdue Architecture itself is not a problem solver for a particular integration program, it will provide a set of well defined problem or subproblem boundaries for the detailed project development under the internal environ-
ment. Therefore, the whole integration program will be well organized within a complete framework.

The above discussion has gone through the major integration tasks in the life history of an integration program. In agreement with the steps of human activities involved in the integration program, the life history of the Purdue Architecture can also be divided into six phases, from the Concept Phase to the Operations Phase. The Phases are one-by-one mappings of the Layers.

Figure 3.4 expands the abbreviated sketch of the Purdue Architecture of Figure 3.1 as applied to a Enterprise Integration system to show the functions carried out at each stage or phase of the life history of the integration program. It will be noted later that with minor modifications, such a life history would apply to any enterprise.

3.1.2 The Choice of Human Tasks

The last section went through the structure and important views of the Purdue Architecture along the vertical direction of its life cycle. Along the horizontal direction, however, the Purdue Architecture has made another important contribution to the study of Reference Architectures, i.e., the split of the Implementation View of the Architecture into three sub-architectures, which allows an extensive discussion of all of the human aspects of the integration program as well as that for the computer or information system and the manufacturing system.

In order to show the true place of the human in the implementation of the enterprise functions just discussed, we need to assign the appropriate functions to the human ele-
ment of the system. This can be done by defining and placing sets of three lines in the graphical architecture representation. This will separate the two implementation architectures into three as shown in Figure 3.5 and thus assign the tasks or functions involved. There is the line called the Automatability Line which shows the absolute extent of pure technologies in their capability to actually automate the tasks and functions of the Enterprise Integration system of the Enterprise Integration Business Entity. It is limited by the fact that many tasks and functions require human innovation, etc., and cannot be automated with presently available technology.

There is another line which can be called the Humanizability Line (see Figure 3.5) which shows the maximum extent to which humans can be used to actually implement the tasks and functions of the Enterprise Integration system of the Enterprise Integration Business Entity. It is limited by human abilities in speed of response, breath of comprehension, range of vision, physical strength, etc.

Still a third line is presented which can be called the Extent of Automation Line (see Figure 3.5) which shows the actual degree of automation carried out or planned in the subject Enterprise Integration system. Therefore, it is the one which actually defines in Figure 3.5 the boundary between the Human and Organizational Architecture and the Information Systems Architecture on the left hand, and the boundary between the Human and Organization Architecture and the Manufacturing Equipment Architecture on the right hand.

The location of the Extent of Automation line has

- Economic
- Political
• Social
FIGURE 3.5 INTRODUCTION OF THE AUTOMATABILITY, HUMANIZABILITY AND EXTENT OF AUTOMATION LINES TO DEFINE THE THREE IMPLEMENTATION ARCHITECTURES
as well as Technological factors in its determination in the ultimate split of functions between humans and machines. This is the line actually implemented in the Implementation View of the Purdue Architecture.

The Automatability Line showing the limits of technology in achieving automation will always be outside of the Extent of Automation Line with respect to the automation actually installed (see Figure 3.5). That is, not all of the technological capacity for automation is ever utilized in any installation for various reasons. Thus, the Human and Organizational Architecture is larger (i.e. more tasks or functions) and the Information System and Manufacturing Equipment Architecture are smaller (less functions) than technological capability alone would allow or require.

Note that for a completely automated plant as an extreme case, both the Automatability Line and the Extent of Automation Line would coalesce together and move to the right edge of the Information Architecture block and correspondingly to the left edge of the Manufacturing Architecture block. Therefore, the Human and Organizational Architecture would disappear and the Information Systems Architecture and the Manufacturing Equipment Architecture would coincide with the unmanned Information Architecture and the unmanned Manufacturing Architecture respectively as noted in Section 3.1.1 (See Figure 3.3).

The actual distribution of tasks and functions of the Human and Organizational Architecture over the Information
Architecture and the Manufacturing Architecture can be represented as shown on Figure 3.6 for convenience.

Another major finding of the architecture diagram is that, as long as the specifications for accomplishment of each of the tasks is honored, the three implementation architectures can be developed relatively independently.

3.1.3 Development of the Modular Task and Function Representation Method

The key to the overall applicability of a Reference Architecture is its espousal of a set of generic tasks and functions and their assignment to certain levels in its functional architecture for any integrated system where this set of tasks and functions must be carried out to operate a Enterprise Integration Business Entity or Manufacturing Facility effectively. Here, again the Purdue Architecture ressorts to informal textual and graphical tools to present its method of modularity.

3.1.3.1 The Task Modules and Function Representation Method

The Purdue Architecture provides the following method for the development of the generic tasks and functions (see their definitions in Section 3.1.1) to specify its functional architectures and its implementation architectures for any integrated system.

Generality of the Task Modules

All the tasks are defined in a modular fashion along with their required interconnections such that they may be interchanged with other tasks carrying out a similar transformation but implemented in a different way. Figure 3.7
shows such a conceptual task module defined by the Purdue Architecture.

The choice of the relevant implementation methods can be governed by independent design and optimization techniques as long as these task specifications are honored. The interconnections between the task or function modules can be considered interfaces which, if specified and implemented by using company, industry, national or international standards, will facilitate the interchange between the modules or the substitution of a particular module for another similar one in the determination of the implementation methods.

In other words, the Purdue Architecture requires that the use of the set of tasks and functions as a generic list be completely independent of

- The method of implementation of the task or function ultimately chosen (whether technological or human-based);
- Any uncertainty as to the availability, maturity, compatibility or stability of the solutions chosen; or
- The ability of the chosen solutions to meet all current and future requirements.

Classification of Task Modules

Transformations in a Manufacturing Enterprise are of two and only two types,

- An informational (data) process;
- A physical (material and/or energy-based) process.

Accordingly, any task and/or function within a Enterprise Integration system can be classified as either an informational (i.e., moving, storing or transforming
FIGURE 3.6 DEFINITION OF HUMAN TASK TYPES IN THE SEVERAL AREAS REPRESENTED IN THE HUMAN AND ORGANIZATIONAL ARCHITECTURE
ALL TASKS CARRY OUT A TRANSFORMATION

FIGURE 3.7 DEFINITION OF THE TASK MODULE
information), or conversely as a manufacturing or physical one (i.e., moving, storing or transforming material and/or energy). Every task should be expressible as a module based on the definition in Figure 3.7.

Every task should be generic in its basic description. All the informational tasks can be considered as control in its very broadest sense, either immediately or at some future time. Likewise, all the tasks on the manufacturing side can be considered conversions, the transformation (chemical, mechanical, positional, etc.) of some quantity of material or energy. Again, all of them are initially defined without reference to their method of implementation.

All task modules of either or both types can be interconnected by the flow of data (information) and commands, and/or material or energy. Informational (i.e., computational) tasks may be interconnected with each other and informational with manufacturing tasks. Manufacturing tasks or functions may be interconnected with each other only through material or energy flow.

Information tasks have only information inputs and outputs to effect their functionality. They also use the information outputs (sources—sensors, etc.) and information inputs (sinks—control commands, etc.) belonging to the production equipment to effect their connectivity to the manufacturing tasks and to allow their monitoring and control of them.

Manufacturing tasks thus may be considered to serve as sources and sinks for information to and from the information tasks (i.e., operational or state data, and operational commands, etc.).
In the Purdue Architecture, sets of function modules and task modules further compose macro functions as shown in Figure 3.8.

In an Enterprise Integration system, typical macro functions are

- Master Production Scheduling;
- Business Planning;
- Factory Coordination;
- Etc.

Bottom-Up Definition Approach

A bottom-up approach (i.e., detailed to general, or, task to function) will be taken to apply the modular method. This bottom-up approach is necessary to assure an accurate definition and description of the operation of the task or function at hand and to assure the commonality of these tasks or functions when comparing them with those arising from different sources.

The bottom-up definition system also assures the completeness of the implementation requirements for the task or function and thus provides for the development of accurate specifications for the implementation media.

Once a particular individual task, function, or macro-function has been thoroughly and correctly described by this bottom-up approach, it may then be used in any subsequent top-down analysis as a separate distinct, self-contained block. Among these blocks, the most basic one is shown in Figure 3.7.
FIGURE 3.8 FUNCTIONAL COMPOSITION AND DECOMPOSITION OF THE TASK MODULES OF PURDUE ARCHITECTURE
3.1.3.2 The Task Modules with Implementation Considerations

Although the initial definitions of tasks and functions are independent of implementation methods, their applications in the Functional Design Phase ask for the capabilities to handle different application methods undertaking the same transformation in the implementation architectures. The following is the representation method developed in the Purdue Architecture to specify the major tasks in the implementation architectures based on the definition of the basic task module shown in Figure 3.7.

Scheduling and Control Tasks (i.e., Information Handling)

The transformation process for a scheduling and control task involves operations on information (data) enabled by an algorithm, which may be represented by algebraic, logic, expert system, neural net, differential equation, table look-up, a computer programmed procedure, or any other method of expressing the relationship involved. The transformation may be considered as a computational block where the algorithms are replaceable along with their associated parameters.

Note that the word, algorithm, is used here in its broadest sense: both as a mathematical expression and as a computer programmable procedure. It may be Deterministic (continuous or discrete) or Stochastic (statistical). The key is that the operation should be totally definable in mathematical and/or computer programmable terms.

Where no definable algorithm or computer program exists, such as in a human, thought-based, innovative process, the transformation process may be described by a written scope or other text which characterizes the transformation in-
volved. All other rules concerning modules as discussed here prevail for the scope modules. This condition also applies for the case where an algorithm exists, but, for whatever reason, it has been chosen to use a human implemented method to carry out a particular function. This module then becomes the non-automated or non-mechanized task or function of the system.

Table 3.1 presents an overview of the breakdown of the tasks in the Information Architecture at their highest level (least detail or minimum breakdown). The parameters needed by the information transformation processes, as an example, may be

- Tables of constants;
- Operational variable value-dependent selections from tables of constants (precomputed adaptive control);
- Results of active on-line or periodic recalculation by adaptive tuning, etc., algorithms;
- Expert systems outputs;
- Neural nets outputs;
- Etc.

Manufacturing-Based Tasks

For manufacturing processes, the basic tasks will comprise generic, irreducible operations, such as the "unit operations" of chemical engineering, or the "machining processes" of mechanical engineering, or other related types of operations or similar classes of operations in each of the other fields of engineering. That is, all manufacturing modules, where applicable, follow a set of basic rules or procedures which is the counterpart of the algorithm in an information task module noted above.
Typical examples can be seen in chemical engineering like a unit operation and its associated equipment or in mechanical engineering like a turning operation and its associated lathe. Each operation and its associate(s) would be a module in the context used here.

Manufacturing modularity will be carried out as follows:

- All modularity will be task or function related;
- All physical, chemical, positional, dimensional and geometrical, i.e., manufacturing transformations are carried out by or in physical equipment, which are all located at Level 0 of the Purdue Functional Hierarchy as described in the Purdue Reference Model for CIM [37] as diagrammed in Figure 3.9;
- Modularity dictates that a module at this level will comprise one or a group of such closely related transformations carried out in the same piece of equipment and will include that required equipment.

The expression of manufacturing modules are generic in the chemistry, physics, position, dimension and geometry of their operation but are specific in terms of the values of their parameters (dimensional and operational) and in their connectivity to form the manufacturing plant.

The Task Module with Human Task Representation

In the Purdue CIM Reference Model [37], which actually presents the Information Functional Network and the Information Systems Architecture of the Purdue Architecture (see Figure 3.10) without considerations of system construction and operation, humans and their relevant functions are treated as "external entities", and communication facilities
are made available to send and receive all necessary information and data from and to the plant and the human worker.
### TABLE 3.1 BASIC CLASSES OF TASKS OF THE INFORMATION ARCHITECTURE OF THE ENTERPRISE

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Communications — the transmission of information between the elements (modules) of the Information Architecture (human workers or components of the Information System Architecture) in the form of messages adhering to a preestablished protocol. Messages may be verbal (agreed upon language), visual (agreed upon nonverbal actions or symbols), or written (agreed upon language and format) if to, from or between human workers. Messages between the elements of the non-human components of the Information System Architecture are through electronic, light, pneumatic or other media in the agreed-upon standard protocol.</td>
</tr>
<tr>
<td>II</td>
<td>Information Storage — the retention of information and data for periods of time (specified or indeterminate) for use in later actions of the Information Architecture components. Storage may be in a human-readable form (written — libraries, etc.) or machine-readable (electronic or magnetic or other related form in databases or related depositories).</td>
</tr>
<tr>
<td>III</td>
<td>Mission Fulfillment — guidance (control) of the components of the Manufacturing elements of the Manufacturing Architecture in carrying out their assigned tasks to assure an optimal completion of the assigned mission of the enterprise.</td>
</tr>
</tbody>
</table>
FIGURE 3.9 THE PURDUE SCHEDULING AND CONTROL HIERARCHY FOR AN INDUSTRIAL PLANT (DISCRETE MANUFACTURING ASSUMED)
FIGURE 3.10 MAPPING OF THE PURDUE REFERENCE MODEL ONTO THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE
Although this arrangement served the purpose of the Purdue CIM Reference Model well and was adopted, it does not provide the solutions to the representation of human involvement in the Enterprise Integration system, nor the representation of the elements and functions of the manufacturing system of the plant, needed by overall manufacturing plant integration.

During the development of the Implementation Procedures Manual project [27], both of these problems were solved by defining the general task representation as shown in Figure 3.11, which may be seen with implementation considerations as being an elaboration of the pattern of Figure 3.7.

Information system tasks (the algorithmic control of the CIM Reference Model), manufacturing tasks, and human-based tasks were then defined as shown in the preceding discussion (see Figure 3.12–3.14). The general applicability of the task module is shown by giving the definition of an expert system in Figure 3.15.

It should be noted that the modules presented here and their connectivity into networks comprises a type of data flow or material and energy flow mechanism. These networks define the functional requirements of both sides of the Purdue Architecture (see Figure 3.2).

A revised version of the task module representation which uses object-oriented methods can be found in [19, pp. 161–175]. The concept of resources needed in implementation is presented there based on the object-oriented method proposed by Süssenguth and Jochem [38, 39]. Only a few modifications to the previous PERA nomenclature are needed to develop the applicability of object-oriented technology to PERA [19].
FIGURE 3.11  GENERAL TASK MODULE WITH IMPLEMENTATION CONSIDERATIONS
FIGURE 3.12 INFORMATION TASK MODULE WITH IMPLEMENTATION CONSIDERATIONS
FIGURE 3.13 MANUFACTURING TASK MODULE WITH IMPLEMENTATION CONSIDERATIONS
FIGURE 3.14  HUMAN AND ORGANIZATIONAL TASK MODULE WITH IMPLEMENTATION CONSIDERATION
FIGURE 3.15  EXPERT SYSTEM OR OTHER RULE-BASED MODULE
3.1.4 Expansion of the Purdue Enterprise Reference Architecture to Cover Any Type of Enterprise

The presentation so far has mainly focused on the background of a manufacturing enterprise. However, in terms of the two dynamic environments (see the definition in Section 3.1.1), enterprises may also be defined as entities which exist to carry out any mission or missions. The mission involves one or more functions implemented with the aid of the resources to satisfy one or more desired ends (goals or objectives) in the face of difficulties and limitations (constraints).

Note that all of the above statements apply to every enterprise regardless of size, physical content, or mission carried out. Therefore, it should be possible to develop a way of expressing these generic concepts for all enterprises in one form. The Purdue Architecture holds this to be one of the goals of an Enterprise Reference Architecture. Table 3.2 is a set of principles concerning the expansion of the Purdue Architecture to include all types of enterprises in its capability to model the development of a program, project or system.

Every enterprise, as noted above, must have a mission in order to justify its existence. Almost always this mission involves the carrying out for, or delivering to some customer, specific products or services or both. These products or services satisfy a need, desire, expectation, etc., of the customer in one way or another. No matter how far fetched, someone, somewhere, receives a perceived benefit from the work of the enterprise or it must fail. This is the customer Product and Service Equipment Architecture
TABLE 3.2 EXPANSION PRINCIPLES OF THE PURDUE ARCHITECTURE

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>All functions and operations on the right hand (Manufacturing) side of the Purdue Reference Architecture Diagram can be represented as relating only to Level 0 of the Purdue Scheduling and Control Hierarchy view of the CIM Reference Model [37] (see Figure 3.9 which illustrates the Purdue Hierarchy diagram with a discrete manufacturing facility as an example).</td>
</tr>
<tr>
<td>2.</td>
<td>Conversely, all functions and operations shown on the left hand (information) side of the Purdue Reference Architecture diagram related only to Level 1 and higher of the Purdue Scheduling and Control Hierarchy view [37].</td>
</tr>
<tr>
<td>3.</td>
<td>All functions and operations on the right hand (Manufacturing) side can also be shown to relate only to the supplying of products or services for the customers, i.e., operation and maintenance of the manufacturing facility to produce products or services for sale to customers.</td>
</tr>
<tr>
<td>4.</td>
<td>Again conversely, all functions and operations on the left hand (Information) side relate only to the well-being of the Business Entity itself, i.e., to its operational control in order to achieve the optimal operating conditions at hand.</td>
</tr>
</tbody>
</table>
concept expressed by the Purdue Architecture. Thus, no organization (enterprise) can long exist for itself alone, it must render some acknowledged service in order to generate the wherewithal to continue its existence.

At the same time, because others are also attempting to supply the same or competing services, these services must be rendered at a competitive cost and quality. Thus, the enterprise must be operated in such a way as to achieve this minimal cost of operation and high quality of service. The achievement here requires control of the operation via information concerning its current and future status, results, etc. This is the expanded Information Architecture concept of the Purdue Architecture. Success for the enterprise is therefore measured by how well the execution of the mission satisfies that specific need, desire, expectation, etc., on the part of the customer.

In the case of the manufacturing system, the need, etc., is obviously for the product being manufactured. For non-manufacturing systems, the above expansion of the potential customer's need for non-manufacturing product(s) allows one to correspondingly expand the definitions of the Architecture to model any kind of an enterprise regardless of its specialized field of endeavor.

As noted earlier, the right hand side of the Business Entity life history diagram (Figure 3.4) relates to customer products and services, while the left hand side relates to informational services to the Enterprise itself. Thus, the diagram can be expanded in its coverage to treat all enterprises—not just manufacturing in an Enterprise Integration implementation program. Figure 3.16 and 3.17 further explain this graphically.
Figure 3.16 Further explanation of the definition of the generic enterprise by the Purdue architecture.

Functions and operations here devoted to service to the business entity, i.e., operation in an optimal manner for the situation at hand.

Functions and operations here devoted to service to the customer, i.e., production of goods and services for sale.
FIGURE 3.17  FURTHER EXPLANATION OF THE DEFINITION OF THE GENERIC ENTERPRISE BY THE PURDUE ARCHITECTURE
Customer response or the provision to the customer of products and services may be carried out in many ways such as:

- By physical things (i.e., manufactured products [the type of Business Entity presented in all the earlier parts of this discussion]);
- By pure physical services (transportation of goods or persons, availability of goods or service for purchase, rental [lease], etc.) or
- Through the supply of information services (data, information) to be used by others;
- Etc.

Figure 3.18 uses the information just developed to modify the labeling of the overall enterprise evolution diagram, Figure 3.4, to cover the corresponding generic enterprise, i.e., the production of either customer goods or services or both. With the same notation, it is extended to cover the total life history of that enterprise as discussed with Figure 3.4 in the case of the manufacturing system.

Figure 3.19 shows how the definition of the tasks of the humans in each section of the Human and Organization Architecture would change in order to be able to carry out the designated mission. For example, an information services company, whose missions are strictly of an information services nature, would employ information services type personnel to perform the desired customer service and thus would have these kinds of personnel in both compartments of the Human and Organizational Architecture.
3.2 The Purdue Methodology

The developer of the Purdue Methodology realized that not only a methodology is necessary, but also this Methodology must be simple and believable in concept, direct in execution, and within the available resources of those needing it. The basic concepts of the Methodology are presented in Table 3.3.

So far, it can be seen that the Purdue Enterprise Reference Architecture follows Definition 2.2 of Enterprise Reference Architecture in Chapter 2. The components of the Purdue Methodology also follow the Definition 2.3 of the Components of Enterprise Integration Methodology in Chapter 2.

All investigations to date have repeatedly confirmed that, as part of the methodology, the Reference Architecture is a superb vehicle for explaining, organizing and guiding the development of a Enterprise Integration system or indeed any type of Enterprise project.

In addition, the Purdue Architecture itself has a major potential for further expansion in terms of detail, explanation, application examples, etc., which will greatly aid its use for Enterprise Integration systems and other Enterprise Integration projects.

The Master Plan of the Purdue Methodology compares the present state of the business enterprise (the As-Is) with the desired future state (the To-Be) to characterize the modification path between them (the Transition). The planning process necessary to define the As-Is, the To-Be and the necessary Transition is readily shown on the architecture diagram by mapping. All the major steps in the master
FIGURE 3.18  THE MODIFIED EVOLUTION DIAGRAM FOR THE GENERIC ENTERPRISE
FIGURE 3.18 (CONT.)  THE MODIFIED EVOLUTION DIAGRAM FOR THE GENERIC ENTERPRISE
FIGURE 3.19  DEFINITION OF HUMAN AND ORGANIZATIONAL ARCHITECTURE FOR AN INFORMATION SERVICES ENTERPRISE
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong></td>
<td>An overall detailed Master Plan for the desired transition is absolutely necessary before attempting to implement any Enterprise Integration program;</td>
</tr>
<tr>
<td><strong>2.</strong></td>
<td>An extensive and detailed instructional manual is necessary to guide and simplify the preparation of the Master Plan;</td>
</tr>
<tr>
<td><strong>3.</strong></td>
<td>Included in the Master Plan is a Enterprise Integration Program Proposal, a prioritized set of integrated projects, each within the resources of the company involved, whose ultimate completion will assure the success of the finally desired operational integration of the enterprise;</td>
</tr>
<tr>
<td><strong>4.</strong></td>
<td>A Reference Architecture is necessary to provide the framework for the development and use of the Instructional Manual, the resulting Master Plan and the ultimately implemented Enterprise Integration Program Proposal;</td>
</tr>
<tr>
<td><strong>5.</strong></td>
<td>Much of the detail necessary in developing the Master Plan follows the example material in the associated document, the Purdue CIM Reference Model which provides a vision of the generic operational structure of an integrated enterprise (see Figure 3.9).</td>
</tr>
</tbody>
</table>
planning process are laid out and identified. No short cuts from this should be considered and are really not possible if good results are expected.

As shown in Figure 3.20, the major flows of information in the typical Master Plan each separate into three major streams — one for each implementation architecture. The numbering in this figure refers to the proposed chapter numbers in the Master Plan. The legends refer to suggested chapter titles and subjects.

The subject matter of the several chapters maps onto the relative subject as outlined by the Purdue Enterprise Reference Architecture. This correlation of the subject matter of the Architecture sections with the corresponding subject matter of the Master Plan has been a major factor in the acceptance of the Architecture by industry.

Based on the Purdue Methodology, what is needed therefore, for each company contemplating a major Enterprise Integration efforts, is for the company to develop a Master Plan covering all of the anticipated efforts required to integrated the whole of the company or factory operation.

After this, smaller projects within the monetary and personnel resources capability of the company can be initiated with the knowledge that the sum of this and all succeeding projects will result in the final total integration of the company’s activities. This will be possible provided that the requirements of the initial planning efforts or the Mater Plan be followed in each one of the resulting projects. Therefore, in terms of an actual program implementation, the Master Plan is the key in the application of the Purdue Methodology.
Step-by-step instructions on developing the Master Plan is documented in the "Implementation Procedure Manual for Developing Master Plans for CIM" [27] and its revision known as the "Guide to Master Planning and Implementation for Enterprise Integration Programs" [36] which are practical guides using textual and graphical tools, including questionnaire systems, to illustrate the processes involved.

3.3 Tools, Etc., Available for Exploiting the Architecture

As has been noted several times, the Purdue Architecture is a framework under which one can discuss any one or more of the many important aspects describing an Enterprise Integration Program. Each of the subarchitectures and, indeed, even the several regions within many of them become localities for the grouping of answers to the topic to be explored in that particular region or subarchitecture.

This applies to such topics as the development and assignment of policies and their resultant requirements; the formulation of tasks and functional modules; and the assignment of the tasks and functional modules for implementation. At each stage or phase of completion of implementation, such as functional design or specification or detailed design, among many others, the regional or local constructs of the Purdue Architecture can be used to collect and assign the very wide variety of manual, graphical and computer aided tools for developing and carrying out such tasks as initial designs, detailed designs, hardware commissioning, software design and preparation, manufacturing equipment design, etc., throughout the whole of the enterprise integration's considerations.
FIGURE 3.20 THE PURDUE GUIDE TO MASTER PLANNING AND IMPLEMENTATION FOR ENTERPRISE INTEGRATION PROGRAMS AND MASTER PLAN INFORMATION FLOW
This discussion is in no way to be considered as a through review of this very wide field, or even an appreciable beginning. It is only intended to point out another facility of the Architecture, which should also be expressed by the Architecture formality. This facility provides a format for the classification and ordering of these various tools and design aids needed to help the development, design and use of the Enterprise Integration system or even the whole Enterprise.

Figure 3.21 provides an abbreviated sketch of the structure of the Architecture and provides the skeleton framework upon which to base the example collection of tools and design aids to be employed there. Included on the diagram are numerical references to Table 3.4 which with the numerals of Figure 3.21 lists each of the important areas on the framework diagram. Important development and implementation aids can then be listed by the areas under discussion. Table 3.4 also presents examples lists of the aids available for the development of implementation endeavor normally associated with each area of the diagram. Note that these lists are examples only and not intended in any way to be comprehensive or complete.
FIGURE 3.21 ABBREVIATED SKETCH TO REPRESENT THE STRUCTURE OF THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE IN RELATION TO TABLE 3.4
### TABLE 3.4 AREAS OF INTEREST ON THE ARCHITECTURE FRAMEWORK
FOR DISCUSSING DEVELOPMENT AND IMPLEMENTATION AIDS
FOR PROGRAMS AND ENTERPRISE STUDIES
(REFER TO FIGURE 3.21)

<table>
<thead>
<tr>
<th>Area</th>
<th>Subject of Concern</th>
<th>Types of Aids Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Mission, Vision, and Values of the Enterprise, operational philosophies, mandates, etc.</td>
<td>Examples sets of Mission, Vision and Values expressions from company annual reports or the basic documents themselves. These are valuable to the extent that they are generic</td>
</tr>
<tr>
<td>2.</td>
<td>Operational policies related to the Information goals and objectives, etc. of the Enterprise</td>
<td>Generic lists of: Policies and requirements related to such topics as: control capabilities; degrees of performance of processes and equipment; adherence to classes of regulations and laws (environmental, human relations, safety, etc.); compliance in the above to a degree of good community behavior (good neighbor, citizen of the world, etc.); quality, productivity, and economic return goals; and related</td>
</tr>
</tbody>
</table>

(continue)
### TABLE 3.4 (CONT.) AREAS OF INTEREST ON THE ARCHITECTURE FRAMEWORK FOR DISCUSSING DEVELOPMENT AND IMPLEMENTATION AIDS FOR PROGRAMS AND ENTERPRISE STUDIES (REFER TO FIGURE 3.21)

<table>
<thead>
<tr>
<th>Area</th>
<th>Subject of Concern</th>
<th>Types of Aids Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>Operational Policies and goals related to the Customer Product and Service or Manufacturing goals and objectives etc. of the Enterprise</td>
<td>Example scopes of the tasks for development and operation of specific processes and plants or other corresponding Customer Product or Service Operation, if for the same process or type of plant, may be directly used or otherwise will be used as an example of types of requirements needed</td>
</tr>
<tr>
<td>4.</td>
<td>Requirements to be fulfilled in carrying out the Information Policies of the Enterprise</td>
<td>Generic lists of requirements necessary to carry out the policies listed in Area 2, probably in the form of scopes of the macrofunctions to be listed in Area 6</td>
</tr>
<tr>
<td>5.</td>
<td>Requirements to be fulfilled in carrying out the Customer Product and Service or Manufacturing related Policies of an Ent-</td>
<td>Example sets of operational requirements for specific processes and process plants or other corresponding Customer</td>
</tr>
<tr>
<td>enterprise</td>
<td>(continue)</td>
<td></td>
</tr>
</tbody>
</table>
TABLE 3.4 (CONT.) AREAS OF INTEREST ON THE ARCHITECTURE FRAMEWORK FOR DISCUSSING DEVELOPMENT AND IMPLEMENTATION AIDS FOR PROGRAMS AND ENTERPRISE STUDIES (REFER TO FIGURE 3.21)

<table>
<thead>
<tr>
<th>Area</th>
<th>Subject of Concern</th>
<th>Types of Aids Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>Product or Service Operation; would include general safety requirements, fire rules, etc., that will influence plant design and process and equipment selection later in the program development; OSHA regulations and Fire Safety Underwriters Rules</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Sets of Tasks, Function Modules, and Macrofunction Modules required to carry out the Requirements of the Information or Mission Support side of the Functional Analysis</td>
<td>Generic lists of Control and Information Tasks, Function Modules, and Macrofunctions</td>
</tr>
<tr>
<td>7.</td>
<td>Sets of Tasks, Function Modules and Macrofunction Modules required to carry out the Requirements of the Manufacturing or Customer Product</td>
<td>Lists of generic unit process operations of chemical engineering, or, of the manufacturing features from group technology for the discrete products industry;</td>
</tr>
<tr>
<td>and Service Mission of</td>
<td>(continue)</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>Subject of Concern</td>
<td>Types of Aids Available</td>
</tr>
<tr>
<td>------</td>
<td>--------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>7.</td>
<td>the Enterprise</td>
<td>corresponding requirements for other types of Customer Product or Service Operations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Connectivity diagrams of the Tasks, Function Modules, and Macrofunction Modules of the Information or Mission Support Activities, probably in the form of data flow diagrams or related modeling methods</td>
<td>Data flow diagram techniques [40, 41]; the generic data flow diagram of the Purdue Reference Model [37]</td>
</tr>
<tr>
<td>9.</td>
<td>Process flow diagrams showing the connectivity of the Tasks, Function Modules, and Macrofunctions of the Manufacturing or Customer Product and Service processes involved</td>
<td>Example flow diagrams for commonly available processes showing material and energy balances and example process operating procedures are likely types of aids here [42]; corresponding requirements for other types of Customer Product or Service Operations</td>
</tr>
</tbody>
</table>

(continue)
TABLE 3.4 (CONT.) AREAS OF INTEREST ON THE ARCHITECTURE FRAMEWORK FOR DISCUSSING DEVELOPMENT AND IMPLEMENTATION AIDS FOR PROGRAMS AND ENTERPRISE STUDIES (REFER TO FIGURE 3.21)

<table>
<thead>
<tr>
<th>Area</th>
<th>Subject of Concern</th>
<th>Types of Aids Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
<td>Functional Design of the Information Systems Architecture</td>
<td>Generic representations of typical control and information systems; functional design aids; lists of sensors, actuators, control functions for particular process equipment examples [43, 44]; data base design techniques [45–47]; entity relationship diagrams [48]; the Purdue Scheduling and Control Hierarchy [37]; example hardware architectures from various vendors; networked communications are also very important [49–56]</td>
</tr>
<tr>
<td>11.</td>
<td>Functional Design of the Human and Organizational Architecture</td>
<td>Example lists of generally required personnel tasks; auditing methods for skill level determination (required vs. available); methods for cultural status assessment and correction [57–64]</td>
</tr>
</tbody>
</table>

(continue)
<table>
<thead>
<tr>
<th>Area</th>
<th>Subject of Concern</th>
<th>Types of Aids Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.</td>
<td>Functional Design of the Manufacturing or Customer Product and Service Equipment Architecture</td>
<td>Example specifications of process equipment to be required; P and I, Ds (piping and instrumentation diagrams) indicating control systems capabilities necessary to accomplish suggested task assignments and degree of automation indicated, obtained from the current literature; computer-based process plant layout and design optimization programs available from a wide variety of vendors for almost any industry, corresponding examples from other types of Customer Product or Service Operations</td>
</tr>
<tr>
<td>13.</td>
<td>Detailed Design of the equipment and software of the Information Systems Architecture</td>
<td>Computer control systems components selection aids from control system vendors; configuration software packages from these same (continue)</td>
</tr>
</tbody>
</table>
### TABLE 3.4 (CONT.) AREAS OF INTEREST ON THE ARCHITECTURE FRAMEWORK FOR DISCUSSING DEVELOPMENT AND IMPLEMENTATION AIDS FOR PROGRAMS AND ENTERPRISE STUDIES (REFER TO FIGURE 3.21)

<table>
<thead>
<tr>
<th>Area</th>
<th>Subject of Concern</th>
<th>Types of Aids Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.</td>
<td></td>
<td>vendors; software project management techniques [65-68]</td>
</tr>
<tr>
<td>14.</td>
<td>Detailed Design of the task assignments, skills development training courses, and organizations of the Human and Organizational Architecture</td>
<td>Example lesson plans and syllabi for necessary training courses; example organizational charts for equivalent groups in terms of numbers of people, skill levels and tasks required; team building [63, 69]</td>
</tr>
<tr>
<td>15.</td>
<td>Detailed Design of components, processes, and equipment of the Manufacturing or Customer Product and Service Equipment Architecture</td>
<td>Detailed design techniques for physical processes and equipment from the major handbooks of the various engineering fields; computerized versions of these design methods available from a wide variety of software vendors; corresponding examples from other types of Customer Product or Service Operations</td>
</tr>
</tbody>
</table>

(continue)
<table>
<thead>
<tr>
<th>Area</th>
<th>Subject of Concern</th>
<th>Types of Aids Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.</td>
<td>Construction, check-out, and commissioning of the equipment and software of the Information Systems Architecture</td>
<td>Project management techniques such as critical path method. These and related techniques are readily available as computerized project management aids from a wide variety of vendors</td>
</tr>
<tr>
<td>17.</td>
<td>Implementation of organizational development training courses, and on-line skill practice for the Human and Organizational Architecture</td>
<td>Continuation of the work under Area 14 in terms of training and staffing of the members of the Human and Organizational Architecture</td>
</tr>
<tr>
<td>18.</td>
<td>Construction, check-out, and commissioning of the equipment and processes of the Manufacturing Equipment Architecture</td>
<td>Continued improvement of the operation of the plant and its associated control system (continue)</td>
</tr>
<tr>
<td>19.</td>
<td>Operating of the Information and Control System of the Information systems</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>Subject of Concern</td>
<td>Types of Aids Available</td>
</tr>
<tr>
<td>------</td>
<td>--------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>19.</td>
<td>Architecture including its continued improvement</td>
<td>involving such techniques as Statistical Quality Control, Statistical Process Control, Total Quality Management, and other related techniques [70–73]</td>
</tr>
<tr>
<td>20.</td>
<td>Continued organizational development and skill and human relations development training of the Human and Organizational Architecture</td>
<td>Tasks here are continued improvement of workers' skills and training of replacement workers; same aids as for Area 11, 14, and 17 prevail</td>
</tr>
<tr>
<td>21.</td>
<td>Continued improvement of process and equipment operating condition to increase quality and productivity and to reduce costs involved for the Manufacturing or Customer Product and Service Equipment Architecture</td>
<td>Requirements and aids available as noted under Area 19 above</td>
</tr>
</tbody>
</table>
4. CIMOSA, GRAI-GIM AND THEIR FORMALISMS

As noted in Chapter 2, CIMOSA and GRAI-GIM take different approaches to establish their Enterprise Integration strategies. This chapter will give a brief review of their respective approaches. The discussion will demonstrate the ways formalisms were introduced into their modeling structures.

4.1 Introduction to CIMOSA

CIMOSA (Computer Integrated Manufacturing – Open System Architecture) was defined and developed as an architecture for the definition, specification and implementation of Computer Integrated Manufacturing systems under an ESPRIT (the European Strategic Programme for Research and Development in Information Technology) project by AMICE (the European Computer Integrated Manufacturing Architecture [reverse acronym]) [35]. Technically speaking, the main goal of CIMOSA [2-15] is to support the process oriented modeling of manufacturing enterprises and to provide execution support for the operation of enterprise systems based on the resulting computerized models so that a flexibility of real-time enterprise operations can be obtained and multi-disciplinary information integration and system integration can be supported.

As a Type Two reference architecture, CIMOSA provides a framework to guide CIM users and CIM vendors to take part in
enterprise modeling. It also provides architectural constructs for the structured description of business requirements and for CIM system implementations. It is expected that the application of CIMOSA will result in a complete description of the enterprise business processes and its day-to-day operations, which is then stored on and manipulated by the information technology base of the enterprise. Real time enterprise monitoring and control are also established based on the CIMOSA enterprise modeling system. Engineering and execution of such models will be heavily supported by an Integrating Infrastructure developed within the CIMOSA project.

Note that the above statements are all futuristic in nature. CIMOSA is still under development and most of these capabilities are not yet complete although the potential has been proven. The reader should also note that CIMOSA and GRAI-GIM are still restricted to discussion of discrete manufacturing systems and to CIM in its narrow sense. They have not been expanded to cover any enterprise integration programs although these seems to be no built in restriction to such an expansion of application [1].

The AMICE project aims at establishing standards for enterprise modeling and supporting environments for enterprise model engineering and operation. In addition, the ESPRIT Consortium AMICE intends to develop these standards to a level of detail facilitating the derivation of compliant specifications. These will enable consistent information technology applications and physical system integration in heterogeneous manufacturing and information technology environments.
With computer-based methods and techniques, CIMOSA will provide means to [13]:

- develop enterprise models in an evolutionary mode (start with a part rather than the complete enterprise);
- define, describe and structure enterprise requirements in a consistent and meaningful way;
- derive from those requirements the system design and relevant and sufficient enterprise component specifications;
- describe their implemented versions and support enterprise component implementation and release for operation;
- operate enterprise systems in heterogeneous environments; and
- maintain the enterprise model (adapt to external and internal changes).

Report [1] lists two important CIMOSA contributions to Enterprise Integration: the definition of its modeling framework and its generic Integrating Infrastructure. The same report [1, p. V-1-5] also points out several strong points of CIMOSA as follows:

- "The AMICE Consortium which initially developed CIMOSA is to be highly commended for their early decision to be as formal as possible in their definition and description of all aspects of this architecture. This was done with the ultimate aim of achieving complete computer executability of all constructs, models, tools, techniques, etc., associated with the architecture."
"Because of its declared goal of formality and eventual computer executability, CIMOSA is the most formally described of the three candidate architectures. ..."

"In order to increase the potential for executability, CIMOSA is developing two major environments, the Enterprise Engineering Environment and the Enterprise Operation Environment and a set of specified system services known as the Integrating Infrastructure. The first of these environments formalizes the development of enterprise models and their conversion to working programs for the system. The second formalizes the testing, proving and acceptance of the resulting programs as new additions or changes to the operating systems. The Infrastructure defines how all such programs, as just noted, work together to carry out the overall functions of the integrated computer system."

The discussion in the next section will focus on the way in which CIMOSA formalizes its definitions and specifications with computer executability as its ultimate goal. This is the way in which CIMOSA will be of the most benefit to this study. A detailed description of CIMOSA can be found in references [2-15] and [22] which are also the main sources of the following description.

4.2 CIMOSA Approach and its Modeling Formalism

The CIM solution provided by CIMOSA is actually a model based systems solution, i.e., enterprise modeling systems supported by computerized information technologies. Such
modeling systems will readily permit the verification of proposed solutions through simulation responses to imposed environmental changes. Similar models will also be used directly for monitoring and control of the business processes and activities.

In order to establish the particular implementation model, CIMOSA provides its modeling framework, modeling methods and modeling constructs to support CIMOSA business users in developing their own requirements definition models. In addition, CIMOSA also supports translation from the user-defined requirements models into the enterprise implementation model, a description of the CIM system and its components [3]. For such modeling processes, the CIMOSA modeling formalism was developed to ensure CIMOSA model applications by computerized techniques.

4.2.1 CIMOSA Modeling Framework and its Life Cycle

CIMOSA explains its architectural modeling framework, which outlines the general concepts of steps or levels of its modeling approach, in natural language with the help of graphical diagrams in a way similar to Purdue.

CIMOSA architectures [3] are defined as:

• The CIMOSA Particular Architecture: the structure and contents of a specific enterprise;

• The CIMOSA Reference Architecture: the collection of constructs which allow the user to structure and engineer a specific enterprise model.

The so-called Reference Architecture (generic and partial levels together) provides the set of constructs which aid the engineer to derive the model for any enterprise. Ge-
eneric constructs (Generic Building Blocks and Building Block Types) are applicable to all industrial enterprises, whereas more specific macro constructs (Partial Models) are aimed at modeling specific enterprise domains and special industries, that is, they can be further re-used and customized in different Particular Architectures.

CIMOSA separates its modeling framework into four parallel view models, i.e., Function, Information, Resource and Organization Views, which are expected to be able to analyze and optimize a particular aspect of the enterprise operation. These four Views are defined by CIMOSA as follows [3]:

- The Function View describes the functional structure required to satisfy the objectives of the enterprise and related control structure, i.e., the rules which define the structures and the principles of the underlying business processes, control sequences or flow of action within the enterprise.

- The Information View lists the information required by each function and how it is collected, handled and stored.

- The Resource View describes the resources, i.e., the real physical equipment such as machines, people, and application programs required for the enterprise, and their relationship to the functional and control structures, and to the organizational structures.

- The Organization View is a description of the responsibilities assigned to individual resources of the enterprise for functional operation and control
of the enterprise and their relationship to each other.

In order to provide modeling support for system specifications and designs based on the user requirements CIMOSA also defines three Modeling Levels [3] as follows:

- The Requirements Definition Modeling Level defines what has to be done in a business sense and terminology. This Level is the domain of the CIMOSA business user and his business requirements. It corresponds to the Functional View of Information in PERA.

- The Design Specification Modeling Level structures and optimizes the business requirements, and specifies the required components. This Level is the domain of system designers. It corresponds to the Functional Design and Specification in PERA.

- The Implementation Description Level describes the means and/or rules to be used in executing the enterprise operations according to requirements of the Requirements Definition Modeling Level. This Level is the domain of system implementers. It corresponds to the Detailed Design in PERA.

In its graphical presentation, CIMOSA puts its two sub-Architectures, its four Views and its three Levels noted above together into a cubical shape called the "CIMOSA-Cube" shown in Figure 4.1. Although this cube was proposed years ago [2-5], not all of its building elements have as yet been defined. This is because there exist inherent weakpoints in its definition. Some researchers have attempted to modify this cube [74] by enlarging or reducing the dimensions of
the CIMOSA cube. However, it is not the task of this report to criticize CIMOSA. Thus, discussions here will focus on the CIMOSA modeling formalisms and their computer process-ability.

4.2.2 CIMOSA Modeling Constructs and Modeling Methods

Although the presentation of the CIMOSA modeling framework takes only a textual description and graphical form, those CIMOSA modeling constructs and modeling methods which are employed inside the CIMOSA framework are introduced in a more formal way. This is necessary to guide the CIMOSA modeling approach for all parties involved, such as business users with different specialties, system designers and system implementers in setting up their own specialized models at each step or level where their positions are identified in the CIMOSA framework. As a result, at each modeling level, only one total model, which is accessible from each of the different views, is expected [14].

The modeling paradigm of CIMOSA relies heavily on the object oriented modeling approach. CIMOSA has made extra efforts to define and formalize its modeling constructs by various means. Beside textual definitions and associated graphs, which are the tools frequently used to define the basic terminology (semantics) of the constructs, mini-examples are shown in CIMOSA documents to help this understanding. In addition, CIMOSA templates with their syntax are introduced for detailed descriptions of these constructs [3].

Take the Function View as an example. CIMOSA has developed a unified function description method used for the
functional decomposition at the Function View. That is, CI-MOSA has created a unified construct, called the Enterprise
FIGURE 4.1 THE CIMOSA-CUBE: AN OVERVIEW OF THE CIMOSA ARCHITECTURAL FRAMEWORK
Function which will itemizes the basic contents of the CIMOSA templates for the formal specifications of the CIMOSA constructs. The basic constructs of this view at the Requirement Definition Level are Domain Process, Business Process and Enterprise Activity. They are used together with constructs of the other views to model enterprise functionality and behavior. Figure 4.2 shows these constructs and their relationship [3].

An Enterprise Function is defined as the business user's view of those tasks which are required to achieve a particular enterprise objective. Here the word task is used as a general term which represents the basic constructs, i.e., Domain Process, Business Process and Enterprise Activities. The Enterprise Function consists of three major parts: the functionality, the behavior and the structure parts, which are presented in Figure 4.3. Functionality as used here describes the static part of the Enterprise Function. Behavior describes the dynamic part of the Enterprise Function, and Structure shows the location of the Enterprise Function within the hierarchy of the Function View decomposition.

CIMOSA believes that the three parts of the Enterprise Function ensure [3]:

- a separation of functionality and behavior, making it possible to revise behavior in order to meet changing circumstances without altering the installed functionality;

- the tractability of functional decomposition as well as the possibility of a break-down into Objectives, Constraints, Inputs and Outputs;
• modeling at different levels of abstraction which therefore allows the execution of the model (for
FIGURE 4.2 OVERVIEW OF BASIC CONSTRUCTS AND THEIR RELATIONSHIPS IN THE CIMOSA FUNCTION VIEW
FIGURE 4.3 DESCRIPTIVE PARTS OF CIMOSA ENTERPRISE FUNCTION
simulation) for different purposes such as strategic, tactical and operational planning.

Furthermore, as noted above, the detailed specification of a CIMOSA Enterprise Function can be presented in the form of its CIMOSA template. Figure 4.4 shows the functional part of the CIMOSA template as defined for the description of any CIMOSA Enterprise Function. Each major item of the template, such as the objectives, constraints, functional descriptions, required capacities, and inputs and outputs, etc., is defined according to the constituent parts of the CIMOSA Enterprise Function (see Figure 4.3) with a rigid syntax attached. As a result, the template is actually an ordered list with a set of grouped entries each of which has a clearly defined syntax [3] including the rules used for any structured language, definition of data type and event table, etc., to be included.

An event-driven modeling approach is taken in the CIMOSA Function View. In other words, the execution of the Enterprise Functions is started by Enterprise Events which will initiate the processing of the set of Procedural Rules associated with the Behavior Part of the related Enterprise Function. Enterprise Events may be external (those which cross Domain boundaries) or internal (those which are triggered by Procedural Rules, etc.). Therefore, in the description of an Enterprise Function, if necessary, a record of the Enterprise Event(s) will also be kept. A CIMOSA template with its relevant syntax can also be defined for the Enterprise Event [3].

In more recent literature [14, 75, 76], a formal method of specification has been introduced to describe CIMOSA constructs with mathematical notations. Reference [14] reveals
that the form of an ordered list with \( n \) entries as is taken by all CIMOSA templates provides a close connection between the specifications of the CIMOSA constructs and the notation of an \( n \)-tuple in set theory [77]. This is because there is a one-to-one mapping between each entry in the CIMOSA template and each element in the \( n \)-tuple. Since the \( n \)-tuples have a more succinct form for expression than the templates, the following discussion will use the materials about the \( n \)-tuples provided by Reference [14] to show the way in which CIMOSA defines its building blocks and sets up its models with them. When there is a need for specifying the model in detail, however, the more descriptive forms, the CIMOSA templates, will still have to be used.

4.2.2.1 An Example of the CIMOSA Modeling Approach Using the \( n \)-Tuple Method

According to the CIMOSA definition, Enterprise Events describe real-world happenings or requests (i.e., orders for products) of the enterprise, which require certain actions. In many cases, events carry information (e.g., the customer order, machine indications, the management order, etc.). This information is described, by following the object-oriented modeling presentation, in the form of a so-called Object View which is a basic construct, i.e., a subset of the Information View. Thus, an Enterprise Event (EE) can be defined as a 4-tuple

\[
\text{EE}: \langle \text{EE-name}, \text{source-list}, \text{object-view}, \text{timestamp} \rangle \quad [4.1]
\]

where

\( \text{EE-name} \) — the Event name;
Enterprise Function

Type: [relevant category - select from list <name>]
Identifier: [FU-<Unique Identifier>]
Name: [name of Enterprise Function in the form: <adjective> <noun> 
<adjective>: qualifying the Enterprise Function 
<noun>: related to the scope of the Enterprise Function]
Design Authority: [name of a person and department with the authority to design/maintain this particular instance]

A. Functional Description

OBJECTIVES: [list of identifiers of all Objectives which must be fulfilled by this Enterprise Function]
CONSTRAINTS: [list of identifiers of all Constraints applicable to this Enterprise Function]
DECLARATIVE RULES: [list of identifiers and names of Declarative Rules applicable to this Enterprise Function in the form of: <identifier of Declarative Rule> / <name of Declarative Rule>]

NB: Declarative Rules can be used to combine OBJECTIVES and CONSTRAINTS together with conditions

FUNCTIONAL DESCRIPTION [textual description of the tasks performed by this Enterprise Function]

REQUIRED CAPABILITIES [identifier of a set of Required Capabilities applicable to this Enterprise Function]

INPUTS
FUNCTION INPUT: [list of Enterprise Function Inputs in the form of <Identifier><name>]
CONTROL INPUT: [list of Control Inputs in the form of: <Identifier><name>]
RESOURCE INPUT: [list of Resource Inputs in the form of: <Identifier><name>]

OUTPUTS
FUNCTION OUTPUT: [list of Enterprise Function Outputs in the form of: <Identifier><name>]
CONTROL OUTPUT: [<Identifier><name>, of list of Control Outputs in the form of <Identifier><name>]
RESOURCE OUTPUT: [<Identifier><name>, of list of Resource Outputs in the form of <Identifier><name>]

FIGURE 4.4 THE CIMOSA TEMPLATE FOR SPECIFICATION OF THE FUNCTIONAL PART OF THE CIMOSA ENTERPRISE FUNCTION
source-list — a list of names of constructs which can generate the event, e.g., resource objects, activities, etc.;

object-view — the name of the object view containing the information attached to the event; and

timestamp — the instant in time of origin of the event.

Enterprise Activities are elementary tasks, i.e., the basic unit of functionality of an enterprise. They are to be performed to achieve one of the basic objectives of the enterprise while operating under some constraints. The Activities require the allocation of time and resources for their full execution. They use function inputs and resource inputs to produce function outputs according to their transfer function definition. They operate under the influence of their control inputs and may also produce control outputs and resource outputs. Thus, an Enterprise Activity (EA) can be defined as a 13-tuple

$$EA: \langle EA\text{-name}, Aobj, Aconst, \{DR_i\}, FI, FO, CI, CO, RI, RO, \{ES_j\}, \{RC_k\}, \delta \rangle$$

$$Aobj \neq \emptyset; FI \cup FO \neq \emptyset; \{ES_j\} \neq \emptyset; RI \neq \emptyset \quad [4.2]$$

where

EA-name — the Activity name;

Aobj — a non-empty set of activity objectives;

Aconst — a set of activity constraints;

\(\{DR_i\}\) — a set of Declarative Rules defined as a combination of objectives and constraints to model the imposed business rules, i.e., management rules, regulations, and internal
policies, etc., acting as pre-conditions of the Activity;

$FI$ — the function input defined as a set of Object Views;

$FO$ — the function output defined as a set of Object Views;

$CI$ — the control input defined as a set of Object Views which can provide run-time information to the Activity (e.g., the Object View associated to the event triggering the process in which the activity is involved) or which can constrain the execution of the Activity (e.g., a work schedule);

$CO$ — the control output indicating the Events which can be generated by the Activity during its execution;

$RI$ — the resource input defined as a non-empty set of resources (including the functional entities able to perform the functional operations of the Activity);

$RO$ — the resource output defined as an Object View on the resources and indicating their status after the execution of the Activity;

$\{ES_j\}$ — a non-empty set describing the ending status of the Activity, i.e., logical predicates defining all the possible termination states of this Activity (e.g., complete, aborted, terminated, etc.);

$\{RC_j\}$ — a set of capabilities required by the Activity; and
δ – the transfer function modeling the task of the Activity.
Activities can be categorized and are subject to standardization. For instance, it can be verified that any manufacturing activity can be classified according to generic classes: move, make, verify, rest.

Processes, including both the Domain Processes and the Business Processes, are constructs used to model the behavior, i.e., the control flow of the enterprise. Processes are used to chain activities and/or sub-processes (only Business Processes can be such recursive constructs) to model large business functions achieving major objectives of the enterprise under management, administrative or operational constraints and rules. Processes are triggered under some triggering conditions, i.e., Events, and operate according to their set of Procedural Rules, which are control structures relevant to CIM activities and covering sequential control, conditional control, parallelism, rendezvous and iterative control. Control structures operate according to values of the ending status of the processes and activities they govern. Thus, any Process \( P \) can be defined as a 7-tuple

\[
P: \langle P\text{-name}, Pobj, Pconst, \{DR_\alpha\}, \{PR_\beta\}, \{EE_\gamma\}, \{ES_\delta\} \rangle \quad [4.3]
\]

\[
Pobj \neq \emptyset; \{PR_\beta\} \neq \emptyset; \{ES_\delta\} \neq \emptyset;
\]

\[
\alpha \in [1, u]; \beta \in [1, v]; \gamma \in [1, r]; \delta \in [1, t]; u, v, r, t \geq 0
\]

where

- \( P\text{-name} \) – the Process name;
- \( Pobj \) – a non-empty set of objectives;
- \( Pconst \) – a set of constraints;
- \( \{DR_\alpha\} \) – a set of Declarative Rules;
- \( \{PR_\beta\} \) – an ordered set of Procedural Rules;
{EEᵣ} — a set of Events involved in the triggering
condition of the Process; and

{ESδ} — a set of ending statuses of the Process,
i.e., a function of the ending statuses of
the last activities of the Process.

Each Procedure Rule above, \( PR_β \), is defined in a form of 3-
tuple by a unique sequence number in the set, a triggering
condition and the next action(s), which are called trig-
ger(s), to be executed when the triggering condition is sat-
ished, i.e.,

\[ PR: \langle Rule-number, triggering-condition, trigger \rangle \]  \[4.4\]

By using the constructs provided above, a simple example
model of a part manufacturing and assembly system may be
graphically presented as given in Reference [14] and as
shown in Figure 4.5. The original system components include
two conveyors C1 and C2, two machines M1 and M2, the assem-
bly station A1, a robot R, and three storage units S1, S2,
and S3. This system is supposed to produce part P made of
one of its components, P1, and another, P2. C1 and C2 move
workpieces from S1 to M1 and from S2 to M2 respectively. A1
requires one occurrence of P1 and one occurrence of P2 to
make one assembly P. Finally, R1 moves P from A1 to S3.

This example system function can be represented as one
Manufacturing Domain with a Domain Process defined as "DP
Produce P" (see upper part of Figure 4.6). The DP can be
further decomposed into four Business Processes defined as
"BP Make P1", "BP Make P2" (the latter does not appear in
the Figure to avoid duplication), "BP Assemble P", and "BP
Store P" (see lower part of Figure 4.6). The BP's are then
decomposed into chains of Enterprise Activities (EA's).
Reference [14] uses the CIMOSA template to give detailed specifications (see Figure 4.7), by following the related descriptive structure with the attached syntax (see Figure 4.4 for references to the functional description part). The Procedural Rules connecting the Enterprise Activities together are defined by the template of the Domain Process, which also gives the specifications of the related information constructs, such as the triggering Events of the Process, and the resource constructs, such as Resource Input/Output, etc.

The example above shows one of the methods that CIMOSA uses to produce a functional model based on building its constructs during the requirement definition process at the Function View. Meanwhile, the overall structure of the information input/output involved will define the Information View of the CIMOSA Enterprise Model. Thus, it is expected that an integrated enterprise model will be generated with a number of the enterprise aspects covered. CIMOSA claims [13, 15] that the flow of control, flow of information and resource management can be analyzed from one total model. Thus consistency, coherence and completeness can be achieved and verified by computer-aided tools (see Figure 4.8).

CIMOSA defines its modeling methods as "the formal description of model building techniques" [3]. It can be seen from many CIMOSA publications that the approach taken by these CIMOSA modeling methods resembles the one used by high level languages and adopted in programming techniques [75]. CIMOSA actually makes use of its modeling constructs as language constructs to attempt to develop a programming system which could be directly used by relevant CIMOSA users to engineer enterprise models. No matter how much more this
FIGURE 4.5 SKETCH OF AN EXAMPLE MANUFACTURING SYSTEM
FIGURE 4.6 PROCESSES AND ACTIVITIES FOR THE EXAMPLE MANUFACTURING SYSTEM
DP Produce P

Objectives: To produce and store occurrences of part P
To produce one part at a time
Declarative Rules: Production cannot exceed 50 parts per day
Function Input: OV1 / P1_Blank, OV2 / P2_Blank
Function Output: OV11 / Part_P
Control Input: Nil
Control Output: Nil
Resource Input: Nil
Resource Output: Nil
Events: EV1 / Make P
Procedural Rules:
  1. ON (START OR Make_P) DO Make P1 & Make P2
  2. ON (ES(Make P1) = ‘P1 made’ AND ES(Make P2) = ‘P2 made’)
     DO Assemble P
  3. ON (ES(Assemble P) = ‘P made’) DO Store S3
  4. ON (ES(Store S3) = ‘ok’) DO FINISH
Comprises: BP-1 / Make P1, BP-2 / Make P2, BP-3 / Assemble P, BP-4 / Store

EA Machine P1

Objectives: To machine an occurrence of P1
Constraints: Nil
Declarative Rules: Use NC equipment only
Required Capabilities: RC10 / Machine P1 Capability
Function Input: OV1 / P1_Blank
Function Output: OV5 / Part_P1
Control Input: OV8 / NC_data_P1
Control Output: EV9 / Maintenance Request
Resource Input: M1 / Milling_Machine_M1
Resource Output: Nil
Ending Statuses: ‘ok’: one occurrence of P1 has been machined
Functional Operations:
Mount M1 (P1_Blank, Ok);
Get (NC_data_P1, Ok);
Run M1 (NC_data_P1, Ok);
Release M1 (Part_P1, Ok);
Where-used: BP1 / Make P1

FIGURE 4.7 CIMOSA TEMPLATES FOR THE EXAMPLE MANUFACTURING SYSTEM
FIGURE 4.8 MAJOR CONSTRUCTS OF CIMOSA ENTERPRISE MODEL AND THEIR RELATIONSHIPS
overall language approach may achieve at the overall enterprise system level, the CIMOSA modeling formalism has already shown the possibilities to formally specify an enterprise system.

In order to manage rapid change, however, the enterprise engineering model needs much more than a programming system. Other capabilities which are required may include, on the one hand, real-life information availability for system analysis, design and verification, etc. On the other hand, it also need the immediate processability or executability of its model system in the real-life operation environment. Therefore, in its implementation model, CIMOSA has defined two Integrated Environments based upon one computerized Integrating Infrastructure (IIS) to provide the necessary operational support for its model engineering.

4.2.3 CIMOSA Implementation Modeling Structure and IIS

After the CIMOSA enterprise model system has been developed at the Requirement Definition Level and has been transformed at the Design Specification Level, it will be mapped onto a distributed processing environment consisting of computers, machines and people linked together by multiple communication systems to realize the required functionalities. Therefore, this structure has to be reorganized into an implementable structure of distributed processing entities at the Implementation Description Level [3].

Major constructs defined for this level are:

- Implemented Functional Operation (IFO) — the smallest controllable unit of work specified by an imple-
mented Enterprise Activity and implemented by real data processing objects, or Functional Entities;

- Implemented Functional Entity (IFE) — the functional resource required to perform the requested Functional Operations.

At the Implementation Description Level, the entire processing system of a CIMOSA Enterprise is described as a nested hierarchy of Functional Entities as shown in Figure 4.9. In the first level of the decomposition, the Enterprise can be decomposed into three cooperating Functional Entities as follows.

- CIMOSA Integrated Enterprise Engineering (IEE) which covers build-time aspects, such as the specification, design, modification and release of the enterprise system;

- CIMOSA Integrated Enterprise Operations (IEO) which covers run-time aspects, such as the execution of the Business Processes and Enterprise Activities; and

- Integrating Infrastructure (IIS) which provides the basic distributed enterprise services.

At the next level, further decomposition is made to provide specialized functional resources to carry out either functions or services as required by enterprise engineering, enterprise operation or computer operation services involved.

The CIMOSA IEE can be decomposed into a set of computer aided engineering (CAE) tools and models which help perform engineering design tasks through the whole design process and allow computer simulation based on the CIMOSA modeling constructs and related modeling methods.
FIGURE 4.9 FUNCTIONAL DECOMPOSITION AT THE IMPLEMENTATION DESCRIPTION LEVEL
The CIMOSA IEO can be again further decomposed into a set of so-called IFE which actually are application software. CIMOSA classifies the IEO as following:

- Implemented Data Storage Functional Entity;
- Implemented Communication Functional Entity;
- Implemented Application Functional Entity;
- Implemented Human Functional Entity; and
- Implemented Machine Functional Entity.

Both the CIMOSA IEE and IEO require the computer operating support which is provided by the CIMOSA IIS. The IIS which is a software system can then be decomposed into four primary Service-Functional Entities (SFE) as:

- Business Related Services (B) — for control of Business Processes and Enterprise Activities and management of resources, such as dispatching the execution of Enterprise Activities and scheduling the provision of resources, which may be considered technically as interfaces with computerized tools, or management of application programs and procedures;

- Information Related Services (I) — for management of information, such as information administration required by application processes, which may be considered technically as interfaces with data or management of database;

- Front End Related Services (F) — for management and control of the Implemented Functional Operations, i.e., representing various types of manufacturing resources to the Business Related Services, which may be considered as interfaces with end-users
and/or manufacturing equipment or, in other words, management of user interfaces; and

- Communication Related Services (C) — for system wide information exchange between distributed instances of the IIS services B, I and F, which may be considered technically as a network interface manager or management of a network.

One of CIMOSA's most important contributions to enterprise integration is its identification and definition of the Integrating Infrastructure (IIS) which is built on top of OSI-based communications facilities to provide the following operational capabilities for integrated manufacturing systems [3, 8, 11,15].

- Events and Behavior Management

A CIMOSA enterprise is subject to a great number of events, generated internally or externally. The IIS must be capable of responding to events based on the definition of the domain model and the use of this model to launch and monitor the business processes on a set of available information and resources. This is the aim of the Business Related Services which mainly use the information contained in the Function and Resource Views of the CIMOSA model.

- Manufacturing Resource Management

Typical resources recognized by CIMOSA are application programs, machines and humans. The IIS must be able to support all computerized applications and interact with manufacturing machines to schedule these resources to the related functional operations. The IIS should allow humans to interact
with the processes, to instruct them, and to receive or input information. These basic requirements are ensured by the Front End Related Services which use the information contained in the Resource View of the CIMOSA model for this purpose.

- Information Management

Enterprises have to handle a wide variety of information and the IIS must be capable of providing this information in a transparent fashion, such as hiding the location of the data, and the manner in which the data is assembled, etc. These basic requirements are ensured by the Information Related Services which use the information contained in the Information View of CIMOSA.

- Communication Management

Enterprises must also develop network infrastructures to link computer systems, machines and humans together in order to support the information exchanges required. The liaison with these infrastructures is the role of the Communication Related Services.

The IIS services are system wide services. Their functionality is available from everywhere on a network of computers and is not physically localized to any one machine. Functionally though they can be expressed as one block of the IIS diagram (see Figure 4.9 and Figure 4.10). The relationship between these services is structured according to the Client-Service Model.

Figure 4.11 gives an overview of the Client-Server relationships. Within the IIS, the Business Related Services
are the center of this operation. The main purpose of the Front End Services is to present every manufacturing
Figure 4.10 Distributed Implementation of the Integrating Infrastructure

The diagram illustrates the distributed implementation of the integrating infrastructure, showing how different nodes (CIMOSA Node 1, CIMOSA Node 2, and External Node) interact through front end services, business process services, information services, communication services, and basic data processing and communication resources.

Key points:
- **Front End Services** are connected to **Communication Services**, which in turn are connected to **Basic Data Processing and Communication Resources**.
- **Business Process Services** and **Information Services** are shown to behave in the same manner.
- **CIMOSA Node 1** and **CIMOSA Node 2** are linked through **Agent Protocol** with **Communication Services**.
- **External Node** is connected to **CIMOSA Node 2** through **SFE.5**.

Access to implementation interfaces are indicated with a dotted line, while channels (protocol links) are shown with a solid line.

**MAP** and **PROF1** are used to denote access points or interfaces within the network.
FIGURE 4.11 CLIENT-SERVER RELATIONSHIP IN THE IIS
capability (e.g., humans, machines, application programs) to the Business Services in a homogeneous fashion, and vice versa.

As a whole, the IIS is the integrating facility (Functional Entity) identified by CIMOSA to provide the functional service capabilities for system-wide administration of information, system-wide scheduling of resources and system-wide control of operations. Therefore, all specific applications will be integrated through the Front End Services in order to form an entirely co-operative CIM system [11, 15] (see Figure 4.11).

4.2.3.1 CIMOSA Integrated Environments

CIMOSA considers that an enterprise integration has to be an on-going process rather than a one time effort. While today's Business Processes are in operation, their successors may be in the process of development. Strategic business plans under development today will set the stage for the system implementation tomorrow.

Therefore, CIMOSA has further arranged its two Implemented Functional Entities, IEE and IEO, into two mutually independent environments which cooperate with the IIS so that the relevant processes of the CIMOSA life cycle can be executed in an evolutionary mode and CIMOSA enterprise modeling can be decoupled from the day-to-day enterprise operations. The key operating support which is shared by both of the environments is provided by the IIS. In CIMOSA, these two environments are called (see Figure 4.12) [10, 27]:

- The Integrated Enterprise Engineering Environment (IEEE);
• The Integrated Enterprise Operation Environment (IEOE).

By adopting an iterative development scheme between the real-world functional entities and their CIMOSA models as supported by the two Integrated Environments, one may create a Particular Implementation Model in the IEEE. Then this model with its executability will be ready for an immediate release for execution in the IEOE [73]. This is because the model was developed under a unified framework for its modeling methods. It is then to be executed in the Integrated Environment by sharing the same operating support from IIS which Is able to interpret the CIMOSA constructs used for the modeling [3]. This is the way that the CIMOSA modeling system is expected to be used in order to render a more active and effective management of change.

4.3 Introduction to GRAI-GIM

The work on the GRAI-GIM Architecture and Methodology was started in the 1970’s at the GRAI Laboratory of the University of Bordeaux. It was found that the decisional (i.e. control) aspects of the production management of manufacturing enterprises were not well accounted for in the methods used for designing advanced manufacturing systems at that time. Therefore, the GRAI methods were developed to address these production management decisional aspects of the manufacturing systems. The first objectives were to model a production management system in order to define the specifications needed to chose a software package for a Computer Aided Production Management system [1, 78].
FIGURE 4.12 CIMOSA INTEGRATED ENVIRONMENTS
GRAI-GIM initially appeared to be more like a collection of modeling tools. However, in its subsequent development, the GRAI model has been extended to model the whole manufacturing system, and the GRAI method has been extended to cover the design of such a system. These GRAI methods and their extension were developed in cooperation with industry. The GRAI Laboratory was also involved in several ESPRIT projects [1, 79]. The results of these efforts became GIM (GRAI Integrated Methodology) which together with its Architecture is called GRAI-GIM in this study.

The domain of GRAI-GIM is to help the designer of a CIM system to develop the model of an Integrated Manufacturing System in order to prepare the specifications for the system. The components specified can then be bought or developed in-house. The need for the development of a methodology to aid the design a CIM system is based on the following concept [[16] (see also Section 2.5.2):

- A CIM system can not be bought;
- A CIM system must be built for each enterprise; and
- A CIM system must be developed based on existing and reliable components.

GRAI-GIM provides the following for its users (1) a reference model, (2) a modeling framework, (3) a set of modeling formalisms and (4) a structured approach. The reference model describes a generic CIM system and its components with emphasis on a hierarchical decision or control system. The modeling framework presents all the models to be used for system analysis, design and implementation during the development of the CIM system. The modeling formalisms define the ways in which these models are presented and manipu-
lated. The structured approach used shows the order of use of the steps necessary to develop the CIM system.

Report [1] considers that "GRAI-GIM is intermediate between CIMOSA and Purdue in terms of the degree of formality implied and used and the consequent ease of understanding by the noncomputer science educated users." It further points out

- "GRAI-GIM has a well developed and described methodology for the application of its architecture and related tools and technology for the development of an integration program by the user through the specification stages." [1, p. V-1-7]

- "The GRAI-GIM architecture allows an explicit modeling and design of the decisional structure of manufacturing systems, while most of the other existing approaches are more information processing oriented." [1, p. IV-2-109]

- "The GRAI Laboratory in producing GRAI-GIM has developed several tools and techniques of potential wide use in enterprise integration studies such as the GRAI-GRID, GRAI-NET, ECOGRAI, the GRAI Model and others. They have also advanced the use of tools and techniques developed by others such as MERISE, Petri Nets, etc." [1, p. V-1-8]

- The philosophy of CIM and of the carrying out of a CIM development project as expressed by the GRAI-GIM descriptive documents is the best of three candidate architectures in terms of its adherence to the philosophy adopted by the Task Force and used in this Report." [1, p. V-1-8]
The following section will discuss the way that GRAI-GIM presents its Methodology and Architecture with emphasis on its modeling formalisms. A detailed description of GRAI-GIM can be found in references [1, 16-18, 78, 79].

4.4 GRAI-GIM and its Formalisms

An important feature of GRAI-GIM is its principle of a structured approach where the two parts of the life cycle of an enterprise program are presented as follows (see Figure 4.13) [1, 16].

- "The first part which is user-oriented"
- "The second part which is technical"

Thus, GRAI-GIM differentiates the user-oriented specifications from the technical oriented ones. Because the first part emphasizes that the user develops the specifications for their own system, these specifications will also be validated by the users under GRAI-GIM. GRAI-GIM believes that these models (see Figure 4.13) and all the modeling methods involved must therefore be readily understandable by the same users. This understandability is achieved through the descriptive documentation of GRAI-GIM as noted by Report [1].

4.4.1 GRAI Reference Model and Structured Approach

Although it is stated in GRAI-GIM that models are based on mathematical formalisms or graphical tools [16], most parts of its presentation are based on textual and graphic descriptions. This is a common practice among all of the three major Reference Architectures.
FIGURE 4.13 USER ORIENTED APPROACH AND TECHNICAL ORIENTED APPROACH: THE TWO PARTS OF THE GLOBAL GIM STRUCTURED APPROACH
A GRAI reference model is aimed at modeling the invariable parts of the CIM system and the links between them. As such a model, the GRAI global model (see Figure 4.14) splits a manufacturing system into three parts [1, 16]:

- the physical system composed of people, equipment, materials and techniques which transforms raw materials into final products;
- the control system which directs the physical system in accordance to its objectives defined.
- the information system which is effectively the system database.

The control system is further split into two subsystems as:

- the decision system which makes the control decisions needed to define the orders to be transmitted to the physical system;
- the operating system which provides direct control to the physical system components using the control decisions from the decision system.

The reader should note the similarity between the GRAI model and the Purdue Reference Model of Figure 3.9.

The life cycle of a manufacturing system is illustrated by the GRAI structured approach as shown in Figure 4.15. Four phases are identified in this GRAI life cycle [16]

- initialization which is equivalent to Mission, Vision and Values or Concept in Purdue;
- analysis which is equivalent to Definition or Functional Analysis in Purdue;
- user oriented design or preliminary design or Specification in Purdue; and
FIGURE 4.14  GRAI GLOBAL MODEL
FIGURE 4.15  GIM STRUCTURED APPROACH
• technical oriented design which is equivalent to Detailed Design in Purdue.

Note that the manifestation and operation phases of Purdue are actually not included here.

Most steps or contents in each phase of the structured approach are presented by either textual forms or graphics. For example, the initialization phase includes the following eight main steps:

1. Overall definition of the company;
2. Presentation of the GRAI Method to the company managers;
3. Definition of the domain of the study;
4. Diagnosis;
5. Initialization milestone;
6. Definition of the people involved in the study;
7. Training of the participants; and
8. Planning and resources for the next phase.

For the sixth or definition step, Figure 4.16 illustrates the introduction of the several groups of people involved in the GIM method application.

In the presentation of the analysis phase and the user oriented design in the design phase, an informal tool which is called "frame specifications" by Reference [78] is employed to present the concepts and elements of each sub-phase in GIM view modeling [16]. Figure 4.17 is such a frame which describes functional view modeling, a sub-phase of the analysis phase which presents four common items:

• Name of the sub-phase;
• Objectives;
• People involved and location; and
• Description of the process which also includes the responsibilities of the people and tools involved.

Figure 4.18 is another frame which shows the functional view design at the user oriented design sub-phase. From Figure 4.17 and 4.18, it can be seen that the expertise of the people and the specialties of the modeling tools are emphasized in the frame specifications.

In terms of the format, the frame is somewhat similar to the CIMOSA template (see Figure 4.7) though there is no syntax attached to the frame method.

4.4.2 GRAI-GIM Modeling Formalisms and Modeling Framework

A modeling formalism, method or tool, by GRAI-GIM's definition, is a means to show a method for transmitting without any ambiguity those parts of knowledge which must be established. It also allows one to build models according to the concepts associated with the formalism [16]. The following modeling formalisms or methods are listed in Reference [16]:

• The entity relationship method;
• GRAI modeling formalisms;
  − The GRAI grid;
  − The GRAI nets;
• The IDEF0 modeling method.

Note that formalism as a word is defined somewhat differently here than in CIMOSA.

As stand-alone modeling tools, they are presented separately with their individually established formalisms, most of which are based on textual descriptions and graphics [16]. This is similar to the way in which the Purdue
Define the objectives of the study
Orient the study
Evaluate the results

Project Board

Define the objectives of the study
Orient the study
Evaluate the results

Working group
- GIM method specialist
- analyst

Synthesis group
responsible of:
- methods
- manufacturing
- control
- management
- computer science

Users - Specialists
all concerned domains

Analysis & Validation

To collect information

FIGURE 4.16 GROUPS INVOLVED IN THE GIM METHOD APPLICATION
<table>
<thead>
<tr>
<th><strong>Name of the sub-phase:</strong></th>
<th>Functional view modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objectives:</strong></td>
<td>To have an overall description of the system</td>
</tr>
<tr>
<td><strong>People involved and location:</strong></td>
<td>An meeting with the working group (GIM method experts and analysts) and the synthesis group (some key users: Production management functions responsible people, manufacturing responsible people).</td>
</tr>
<tr>
<td><strong>NB:</strong></td>
<td>In the same meeting are defined the GRAI grid and the physical system model</td>
</tr>
<tr>
<td><strong>Description of the process:</strong></td>
<td>The definition of the domain made during the initialization phase is taken as basis for the modeling. The modeling is a top down description from the definition of the domain. All the people present in the meeting are involved and can make proposals. The GIM method expert is in charge to translate the descriptions done by the key users into an IDEF0 representation on a blackboard.</td>
</tr>
</tbody>
</table>

**FIGURE 4.17 FUNCTIONAL VIEW MODELLING AT ANALYSIS PHASE**
**Name of the sub-phase:**  Conceptual functional view design

**Objectives:**  To elaborate the functional model of the further system at the conceptual level

**People involved and location:**  The synthesis and the working group for the first two levels and the working group alone for the detailed levels

**Description of the process:**  The IDEF0 modeling is a top down description from the definition of the domain. All the people present in the meeting are involved and can make proposals. The GIM method expert is in charge to translate the proposal done by the Synthesis group into an IDEF0 representation on a blackboard. The design of the functional model is made according to the design of the physical system and in coherence with the design of the GRAI grid.

When the first two levels have been performed during the meeting, the analysts detailed each boxes.

---

**FIGURE 4.18  FUNCTIONAL VIEW AT USER ORIENTED DESIGN SUB-PHASE**
"Implementation Procedures Manual" [27] presents its modeling methods and tools.

GRAI-GIM also recognizes that with the use of multiple modeling tools, a modeling framework is necessary to define the places where an individual modeling tool should fit in the process of enterprise integration. Based on the functional decomposition in the GRAI reference model, GRAI adopts four views, information, decision, physical and functional views, to describe different aspects of any manufacturing system. In order to help deal with the complexity involved in development of the specifications of any manufacturing system, GRAI introduces three abstraction levels as follows [1, 16].

- conceptual level: the level which asks the question "What?" without any organizational or technical consideration;
- structural level: the level which asks the questions "Who?", "When?" and "Why?" to integrate an organizational point of view; and
- realizational level: the specific level which asks the question "How?" to integrate technical constraints.

By adding the principle of the GRAI structured approach, i.e., the user oriented approach and the technical oriented approach as used in developing specifications (see Figure 4.13), the GRAI-GIM modeling framework as shown in Figure 4.19 is obtained. Eight corresponding modeling domains can be identified on the modeling framework to indicate the proper locations for defining the modeling tools in Figure 4.20.
FIGURE 4.19  GIM MODELING FRAMEWORK
FIGURE 4.20  GIM MODELING FRAMEWORK WITH MODELING FORMALISM USED
5. APPROACHES TO FORMALIZATION OF PERA AND THE PURDUE METHODOLOGY THROUGH INTERFACE CONCEPTS

Based upon the discussions in the previous chapters, the basic tasks of formalizing PERA can be identified and initiated as discussed in this chapter.

5.1 Formalism Depends on Concerns for Enterprise Integration

As implied in Chapter 4, the degree of formalism taken by a reference architecture and its methodology fundamentally depend on the way in which they define and treat the subject programs. For example, CIMOSA establishes its modeling formalism by specifying a computer supported modeling approach to its integration project, and this formalism is therefore strongly influenced by the formal methods which have developed for the specification of hardware or software for computer systems. At the same time, GRAI-GIM presents a guideline to illustrate the way in which its users can use a series of individualized methods jointly in order to prepare system specifications themselves for their integration projects. Therefore, user-understandability in the use of these methods is emphasized whenever the formalisms involved are developed and presented within GRAI-GIM. In either case, it is the purpose of the approach involved that decides the
special way to present the relevant information required for the different aspects of the integration projects.

Similarly, there are historical reasons why there should be an informality of expression for the Purdue Methodology and the Purdue Enterprise Reference Architecture and the resulting criticism from more theoretically minded individuals for its informality as mentioned in Chapter 1. First, the Purdue Laboratory for Applied Industrial Control has had a long-term exposure to a wide range of industrial applications. Second, in this new field of Enterprise Integration, there is so far no well-organized theoretical basis available for developing a unified formalism capable of presenting every aspect of the knowledge needed for integration projects. Third and finally, during its preparation, the authors of the Purdue Methodology and Purdue Architecture were obliged to foster a wide degree of understanding of the material involved among its Industrial Consortium members most of whom are practitioners from industry rather than theoreticians. An important consideration of the Consortium members was, of course, to first present their ideas and express their thinking and experience in a way that their industrial colleagues would easily understand rather than to find a rigorous formalism.

This does not say, however, that formalism should not be an important consideration for application to the Purdue Methodology and its Reference Architecture. On the contrary, formalism, which helps convey the information about an enterprise integration program or other development program in a prescribed form with its conformity to a related set of definitions, rules and standards, etc., is one of the
keys to a successful further extension of the Architecture's application.

5.2 Formalization of Purdue Architecture for Information Presentation between Program Participants

There is always a distance in time and in space between those people who develop and document the Methodology and Architecture and those who must apply them. Therefore, the Reference Architecture needs to be unambiguous, consistent, and concise in order to minimize potential misunderstanding and misinterpretation of its message and meaning.

The necessity for the application of a disciplined formality to the use of an enterprise development program can be readily identified in the context of the program life cycle and its stepwise approaches as shown by the Purdue Reference Architecture. As noted in Section 2.5.4.1 and Section 3.2, PERA and its Methodology emphasize the important role played by a comprehensive master planning program at the beginning of an enterprise development program (see Figure 3.20). The first three Phases or Layers of the Purdue Reference Architecture (see Figure 3.2 and Figure 3.18) outline the procedures for preparing a Master Plan for the Program. The Architecture helps assure the overall completeness of the program, the satisfaction of Management Requirements for performance and economic gain of the program, and the overall satisfaction of all the other pertinent considerations.

During the preparation of the Master Plan, there must be frequent communications and/or information presentations between the master planning team and higher management. The planning team will collect, record and expand the higher
management's mission, vision and values and other expectations that they may have for the program. They will also take management's sets of policies and use these for the preparation of the Functional Requirements of the subject enterprise. In addition to providing the planning team with the management developed information needed for the planned program, the management will closely monitor the planning processes through program reviews.

There may be a semantic gap between the everyday business language used by management people to express management concepts and expectations on the subject program and the technical language used by the professionals in the planning team to develop the Master Plan. In order to ensure that the important concerns of higher management are transformed correctly and consistently into technical requirements, the exact concepts and meanings (i.e. semantics) should be captured by applying a widely accepted dictionary or glossary and standard graphics with clearly defined graphical semantics and syntax. At the same time, reinforced by the same dictionary and standard graphics, those reports to the management on the program as it proceeds should be understandable or even "informal" enough to ensure that management is kept informed of the program progress. That is why Figure 5.1 indicates that the final form of the Master Plan is shown as being developed in two parts, one to give an informal report to the management which explains and interprets the other part which may be as formal as required for the information presentation or transformation between the two groups of professionals, the master planning team and the detailed design team.
FIGURE 5.1  FORMALITY AND THE MASTER PLAN IN THE PERA STRUCTURE
Not only does the stepwise structure of PERA indicate the necessity for the enforcement of a certain degree of formality in the information presentation or transformation between different working groups, but the derivation of the three Implementation Architectures also needs a multidisciplinary coordination such as in defining the Extent of Automation Line.

As shown in Figure 5.2, the Concept Layer deals only with management type activities. These concepts are all the results of high-level decision making functions. These must rely completely on human innovation and no major formality is possible here. In the Definition Layer, starting from the functional requirements, most required activities become automatable or programmable. Therefore, as shown in Figure 5.2, their descriptions or specifications can probably be defined formally or in a computer interpretable form.

5.3 Semantic Basis for Further Development of Enterprise Reference Architectures

Each major reference architecture and its methodology has its own features and style which reflect the different academic or industrial backgrounds of their developers. As a result, there is no common semantic basis between them for further mutual development, or to jointly develop a universally acceptable architecture and methodology.

One of the first needs is to develop a widely accepted dictionary as a common semantic basis to establish cross-reference between them. Such a basis will provide support for further inter-architecture studies. This study provides later an initial glossary which could be used for this purpose.
FIGURE 5.2  EXTENT OF FORMALISM IN RELATION TO THE INTEGRATION FUNCTIONS
5.4 The Importance of Interfaces in Enterprise Integration and their Treatment

5.4.1 Introduction

Review of the National and International Standards available and being prepared for the industrial communication and control field shows the major place of interfaces in relation to their consideration in terms of the application of standards to an enterprise integration program. It is readily apparent there that the standardization of interfaces and their application is a major, if not the primary, concern of all standards developments.

Further consideration of the interface question will show that they are indeed a major, if not the primary, concern in all enterprise integration efforts and not just in regard to standardization alone.

5.4.2 Interface Standards in Enterprise Integration [36]

It was long the policy of the International Purdue Workshop on Industrial Computer Systems that the establishment of a set of interface standards (communications and programming) would be the easiest and best way of assuring the interconnectivity of the elements of an Enterprise Integration scheduling and control hierarchical computer system and the transportability of all computer programs between the several computer nodes of the system. [37]

There is a decided trend in industrial control at this time to develop the concept of "configuration" rather than direct computer programming for many of the vendors' prod-
ucts in the industrial control field, particularly micro-
processor-based systems for use at Level 1 (Figure 5.3).
Figure 5.3 Present Extent of Configurability in Computer Systems

**NEEDED:**
- Configurable Packages (i.e., Defined Interfaces)

**LEVEL 4**
- Programmed Technology Probably Now Available for Configurability

**LEVEL 3**
- Technology Now Configurable

**LEVEL 2**

**LEVEL 1**
While these latter systems are very easy to use, they do generally prevent the user from directly programming the system, i.e., altering the available menu of possible functions. They also tend to be less standard than the languages from which they have been developed.

The developer of a Master Plan for a proposed enterprise integration program should therefore prepare a specification for the communications interface between system units and a companion specification for the programming interface between the systems developer and the implementation of the system itself. An example of some of the important points in such a dual specification is presented in Table 5.1. This specification becomes part of the Master Plan and should be agreed to by all vendors involved.

Figure 5.4 shows how the physical interfaces characterized in the Purdue Enterprise Reference Architecture may be modeled. Note that sensors and other data gathering devices are considered to be part of the Information Systems Architecture. Likewise consider the devices by which the human worker communicates with the factory equipment through the Information System Architecture. Note also, that sensor receptacles such as thermowells are considered part of the Manufacturing Equipment Architecture. Similarly all valves and other actuators are also considered part of the Manufacturing Equipment Architecture.

Let us now consider the general subject and nature of interfaces within the reference architecture and its various subarchitectures themselves.
TABLE 5.1  A SUGGESTED SET OF PROGRAMMING AND COMMUNICATIONS INTERFACE STANDARDS

I. Programming Interfaces [80]. See also Figure 5.3.

1. Level 1 - All work at this level should be carried out by "configuration" using the available configuration aids developed by the control system vendors. These programs tend to be proprietary and restricted to one model of control system. They are subject to change by the manufacturers as competition dictates. They comprise a set of menus of possible functions from which the user chooses those desired for the case at hand.

2. Level 2 - Some configuration tools are available at this level but more actual programming is required of the users. All necessary programming should be carried out using high level languages as far as possible. The minimum possible number of such languages should be specified to minimize the learning required for system developers and to promote the transportability of the resulting programs between the computer nodes of the overall system.

3. Levels 3 and 4 - As one progresses higher in the hierarchy menu type programming aids become less available and more direct programming is necessary. In many cases, however, preprogrammed packages are available from vendors to carry out single tasks or groups of tasks at these levels. Compatibility of these "packages" with each other and with the overall system becomes the overriding factor in their selection. The selection of languages may be somewhat modified for these levels compared to Level 2 because of the differing tasks and the different backgrounds of the personnel involved. Again, the overall list of languages involved should be kept to minimum.

* At the time of preparation of this report some of the discussed interface standards are still under preparations. Those preparing a Master Plan should consult the latest version of these standards documents before attempting to establish their own company's standards in these cases.
II. Communication Interfaces [81].

1. Levels 1 and 2 - The distributed, microprocessor-based, control system now comprising the major offerings of the control system vendors usually incorporate a proprietary communications system unique to that vendor's offering and often to the particular models involved. These are usually bit serial systems closely resembling the MAP and IEEE 802 systems discussed in [81]. Efforts are underway to make them completely compatible with the standards being developed by the MAP/TOP group. In any case, the CIM program developer must assure himself that the chosen Level 1 system is or can be made readily adaptable to the other computers and communications systems with which it must communicate either by adherence to accepted standards or through "gateways" which achieve the same purpose. [81] outlines the standards and interfaces involved.

2. Levels 3 and 4 - The MAP/TOP proposals handle these levels as well as those discussed above. Here we are generally discussing computer to computer interfaces because of the types of tasks involved. Again the enterprise integration program developer must assure himself that the chosen computer communications systems follow the MAP/TOP proposed and accepted standards discussed in [81] or that suitable "gateways" are made available by the respective manufacturers to assure the same compatibility of communications promised by the standards themselves.

3. As noted in the Programming Interfaces section above, these above standards or alternate gateway solutions should be specifically stated in an enterprise integration specification as part of the Master Plan and agreed to by all vendors involved.
FIGURE 5.4 INTERFACES IN THE OVERALL ENTERPRISE INTEGRATION SYSTEM
5.4.3 Categories of Interfaces

Just as there are two types of Enterprise Reference Architectures, Type 1 describing a physical enterprise integration system and/or its components, and Type 2 describing the development process of producing the enterprise integration system, there are also two types of interfaces to be considered. They are (1) those carrying out the interconnection and exchange between the components of the enterprise integration systems to be appropriately called Type 1 Interfaces, and (2) those carrying out the transfer of information between the separate phases of the enterprise integration development process itself again to be appropriately labeled as Type 2 Interfaces.

Type 1 Interfaces will also be called Internal System Interfaces. In addition, they can be called Interarchitectural or Intraarchitectural Interfaces depending upon whether they are connecting components within one implementation architecture or the components of two separate implementation architectures. Human/Machine Interfaces would be classified as Interarchitectural Interfaces while Communication Interfaces within the Information Systems Architecture would be Intraarchitectural Interfaces.

5.4.4 A Discussion of Internal Interfaces

Internal, or Type 1, Interfaces can be divided into two further categories as follows:

1. Transfer of information between components of the Information Systems Architecture and/or between those and components of the other Architectures is classed as a communications function. (Such communi-
cations functions would also occur in information handling elements of the Customer Product and Service Architecture.

All of these Interfaces are Communications Interfaces, i.e., the propagation of a satisfactory interpretation and comprehension of information transferred through an appropriate transformation of the signal involved.

2. Transfer of materials and/or energy between modules of the Product and Customer Service Architectures through material handling and energy conversion and transfer.

The following general statements can thus be made concerning interfaces within and between the several implementation architectures.

5.4.4.1 Interfaces within the Manufacturing Equipment Architecture

In this Architecture, all of the interfaces are those which permit the physical flow of either material or energy or both across the boundaries between the individual units of the manufacturing equipment. For example, a heat exchanger works as an interface between the fuel combustion system and the heated fluid system; an in-process storage works as an interface between two units of the manufacturing equipment.

5.4.4.2 Interfaces between the Manufacturing Equipment and Information System Architectures

As noted in Figure 5.4, the interface problems here will basically involve only the wiring problems between the
equipment of the two Architectures. Under PERA, the wiring configuration directly connects the sensors or controllers on the side of the Information System Architecture to the related subject element (such as a control valve) of the Manufacturing Equipment Architecture.

5.4.4.3 Interfaces between the Manufacturing Equipment and Human and Organizational Architectures

Human involvement with the Manufacturing Equipment Architecture always involves the human operator acting as another machine component (direct labor) of the process equipment, such as an additional material moving and placing device. The special position and capabilities of humans enables them to directly handle the material flow of the manufacturing process. Physically, the human operator joins the material handling system within the Manufacturing Equipment Architecture. Traditional Human Factor researches have provided many popular and detailed instructions on the compatibility design of such Man/Machine interfaces to facilitate and safeguard the physical inputs from the human operator and to further assure his physical safety. Under PERA, this Human-Factor Man/Machine interface would be considered as part of the Manufacturing Equipment Architecture.

5.4.4.4 Interfaces within the Human and Organizational Architecture

In this architecture, only one type of interface exists, that of the human to human interface. This interface relates to that information representation and communication where an established dictionary, or other standard semantics, and a standard graphical semantics and symbology may
be necessary when people from different backgrounds are involved for informational interactions.

5.4.4.5 Interfaces between the Information System and Human and Organizational Architectures

Human interaction with the Information Systems Architecture always involves the transfer of data or information across an interface which allows the information system device and the human to interpret (process the results of) the operations of the other with exactly the same meaning in passing information between these entities across the interface. Therefore, each of them has to follow certain prescribed syntax and semantics, or communication protocol(s) to ensure the consistency and unambiguity required for the interaction or the transaction. The Man/Machine interface is here considered to be part of the Information System Architecture as noted in Figure 5.4.

5.4.4.6 Interfaces within the Information System Architecture

In order to execute the input of information or to process data within the computer system, interfaces between the elements of the information system are always involved to allow the computer(s) or other devices on either side of the interface to correctly interpret or process the results of the operations of the other party across the interface. Therefore, these interfaces have to meet certain requirements from the aspects of the communication protocols involved which specify the functional, electrical, and mechanical features of all signal and power lines which cross that boundary. Thus, the parties on each side of the inter-
face will be able to follow the preagreed-upon syntax, semantics or communication protocol(s) as insured by the interface.

However, because of the extremely wide range of transfer requirements, it is difficult to define a uniform interface or develop one standard to cover all computer peripherals. Although there are many workable alternatives to define certain classes of interconnections that are suitable for both computers and peripherals, this very proliferation means that decisions on interface standardization must still be carefully made in order to obtain the maximum degree of compatibility for possible system expansion.

5.4.5 The Use of the Architecture Diagrams for the Representation of Interfaces in Integrated Enterprises

The diagrams presented in Figures 5.5 to 5.11 can be used to represent the interarchitectural interfaces between each of the implementation architectures and either of the other two. The first four figures show some representations of the interfaces that must exist between the separate implementation architectures. These present the interfaces which must exist between the control system and the processes being controlled as well as the interfaces between the plant operator and both the control system and the manufacturing equipment. Figures 5.5 and 5.6 show the ideal interfaces using the functional block diagrams of the Purdue Implementation Architectures. Figure 5.7 corresponds to that of Figure 5.5 and 5.6 but uses the sketch of the Scheduling and Control Hierarchy [37] as a medium instead. Figure 5.8 shows the varied nature of real computer system applications and would be a detailed design phase representation. This
FIGURE 5.5 IDEAL INTERFACES BETWEEN THE HUMAN AND ORGANIZATIONAL AND THE OTHER ARCHITECTURES
FIGURE 5.6 IDEAL INTERFACE BETWEEN THE INFORMATION SYSTEM AND THE MANUFACTURING EQUIPMENT ARCHITECTURES
FIGURE 5.7  IDEAL INTERFACES AS REPRESENTED ON THE PURDUE SCHEDULING AND CONTROL HIERARCHY DIAGRAM (CORRESPONDS TO FIGURE 5.5 AND 5.6)
FIGURE 5.8 HUMAN AND ORGANIZATIONAL ARCHITECTURE INTERFACES MODIFIED TO MATCH THE NON-IDEAL CASE FOR THE INFORMATION SYSTEM ARCHITECTURE
FIGURE 5.9 DEVELOPMENT OF STILL ANOTHER METHOD OF REPRE-
SENTING EACH OF THE INTERFACES BETWEEN THE SEVERAL
IMPLEMENTATION ARCHITECTURES
FIGURE 5.10 USE OF THE PRISM-BLOCK REPRESENTATION OF THE INTERFACES BETWEEN THE SEVERAL IMPLEMENTATION ARCHITECTURES
Human and Organizational Architecture

- Human-Factor Man/Machine Interface design available
- Manufacturing Equipment Interface design available

Interfaces between:
- Human and Information System
- Manufacturing Equipment
- Human and Manufacturing Equipment
- Manufacturing Equipment and Information Equipment

Semantics, Syntax, and Standard Protocols

Industrial Wiring Standards, etc., available
latter sketch represent the penetration of computer systems into all aspects of individual systems today.

Figures 5.5, 5.8, 5.10 and 5.11 diagram the ubiquitous presence of interfaces involved Human-Machine interactions throughout the modern enterprise due to the implementation methods resulting from the major research and design work carried out on the technical and office use of computers and on plant operator interfaces [59, 60, 82]. These implementations are now commonly carried out on CRTs or similar operating devices. Corresponding systems are now regularly used by all individuals who must work with computer systems, regardless of their function within the enterprise integration system as noted by these figures.

The other major class of interfaces is that of data interfaces, i.e., the derivation of operating data from the manufacturing system and the return of control commands to it from the information system (Figures 5.6, 5.7, 5.10 and 5.11). In addition Figure 5.4 presents another view of these same concepts.

The earlier figures (Figures 5.9 and 5.10) show how these interfaces might be shown on the Purdue Enterprise Reference Architecture’s functional design phase or functional specification block on the architecture’s structural diagram.

Thus Figures 5.9 and 5.10 show how a new form, a prism block, readily illustrates all three possible interfaces. Note on the diagrams of these figures how the prism block is physically formed and how the former Extent of Automation line becomes a representation of the interarchitectural interfaces. Thus with this new interpretation the same diagram can have several meanings. In Figure 5.4 the actual
connectivity of the sensor and valve interface as well as the operator's access to the control of the process is illustrated. Note that valves themselves and sensor installation fixtures such as thermowells are considered part of the Manufacturing Equipment Architecture. However, sensors as data sources are considered part of the Information Systems Architecture as are transmitters and other related control output devices. Correspondingly, operator's stations, keyboards, control buttons, etc., are also part of Information Systems, while manual valves and related devices are part of Manufacturing Equipment. The operator is considered to be working across the interface. Only people are considered to be part of the Human and Organizational Architecture. No devices are represented there.

It must be kept in mind that this discussion of interfaces in the diagrams up to this point does not include intraarchitectural ones, i.e., those between the equipments comprising any one architecture. The major examples of intraarchitectural enterprise are, of course, those communications interfaces which interconnect the elements of the Information Systems Architecture [36, 49, 51, 52, 54, 55, 82]. These must be represented by models and tools which break down the Information Systems Architecture into its constituent elements such as the Scheduling and Control Hierarchy, Communications Systems Connectivity Diagrams, Hardware Architectures, etc. [37, 40].

Figure 5.11 presents one means by which each of the types of Intraarchitectural Interfaces together with the Interarchitectural interfaces can be represented by the prism. Note that interfaces are not required between the functional architectures. Interfaces are physical entity or im-
implementation dependent only. The following additional points can also be made on the subject of interfaces:

1. Interfaces are interconnections between elements of the physical architectures (Information Systems or Manufacturing Equipment) or between elements of these physical architectures and the Human and Organizational Architecture or between the two physical architectures themselves.

2. Interfaces are the electronic and/or mechanical interconnections which permit two or more physical (or Human and Organizational or both) modules to carry out the information or material and energy transfer functions of the two or more functional modules which are interconnected.

3. The elements of a functional network or two functional networks require no interfaces between them as only virtual information transfer is involved.

4. The actual form in which the information exists is a function of the Physical or Human and Organizational Architecture involved.

5. With only interfaces standardized vendors are free to innovate as they desire in computer hardware design, etc. The only remaining requirement is a guaranteed compatibility of that innovative design with the standard interface prescribed.

The following are important recommendations concerning the design of the above interfaces:

1. The method of processing information requests and then transmitting the result back to a user should be transparent to the users.
2. All users should have the same method of accessing information, regardless of where they are connected to the network.

3. Interfaces to the outside world (customers, suppliers, government agencies, capital providers, etc.) should be standardized.

4. Interfaces, protocols and interactions across all levels of management, support and operational systems should be standardized for control, data storage, retrieval, and communications.

5. All major functions should have a standard format for I/O which would be used by any vendor supplying software to perform that function.

6. Enterprise integration system communications should follow OSI, MAP, Field Bus and other established ISO and IEC Standards to promote the open system interconnect capability.

7. Each of these interfaces must be designed and developed as part of the work of design and development of the Integrated Enterprise Entity.

8. Type 1 interfaces should be treated as internal components of the Integrated Enterprise Entity and enter the program documentation as such.

5.4.6 Some Further Definitions of Internal Interfaces

Figures 5.12 to 5.17 present a series of sketches which model the above sets of statements concerning internal interfaces and other interesting points about them as well.
5.5 A Discussion of Program Interfaces

In contrast to the internal interfaces just discussed, program interfaces represent the connectivity of the phases of the enterprise integration program itself. Since a Type Two Reference Architecture describes the conduct of an integration program by means of a set of major Phases, these Phases must be connected by a set of interfaces which describe the handling of the information transmitted between the phases involved. The following are a set of concepts and requirements concerning the information transfer taking place:

1. The transmissions involved are all transfers of information between groups of humans. Where automated design, etc., techniques are contemplated, machine readable forms of the transferred documentation may be specified. However, all must have a human interpretable form as well. This latter may be a separate document.

2. Work within a phase of the architecture to develop this documentation should be carried out in the same media that will be eventually transmitted in order to avoid the extra work and potential errors of translation, etc.

3. The description which is required is that necessary for ready understandability and use of the material transmitted across the Program Interfaces by those carrying out the next Phase of the work.

4. Although the Phases or steps separated by the Program Interfaces are consecutive, the tasks of each Phase are focused on its own special aspect of
A MODEL FOR A GENERIC BIDIRECTIONAL INTERFACE

FIGURE 5.12 REPRESENTATION OF A GENERIC INTERFACE
FIGURE 5.13 ANOTHER GENERIC INTERFACE MODEL

GENERAL INTERCONNECT
OF DATA FLOW FOR HUMAN AND MACHINE
INTERACTIVE TASKS

CLOSE COUPLING OF HUMAN AND MACHINES

FIGURE 5.13 ANOTHER GENERIC INTERFACE MODEL
FIGURE 5.14  INFORMATION AND DATA FLOW IN THE ENTERPRISE
FIGURE 5.15 RELATIONSHIP OF A STANDARD ARCHITECTURE TO A SPECIFIC SYSTEM OF ENTERPRISE INTEGRATION
FIGURE 5.16 DIAGRAM OF THE INTERFACES BETWEEN THE INFORMATION SYSTEM AND THE CUSTOMER PRODUCT AND SERVICE ARCHITECTURES WHILE IN THE OPERATION MODE
FIGURE 5.17 THE PLACE OF THE HUMAN IN THE INTEGRATED ENTERPRISE
system development. For example, prior to the functional analysis, these tasks are business-oriented. The tasks in the Definition Layer are function-oriented without consideration of implementation factors.

5. Information transfer across the program interfaces always involve transmission of this information between humans or groups of humans with a maximum interpretability and understandability.

6. The degree of formality to be used in PERA must be phase specific, and must involve the minimum complexity of expression needed to assure a non-ambiguity of interpretation and understanding of the information transferred across the phase-to-phase interface. Whenever possible, the media and format used should match the requirements of the using phase.

Figure 5.18 shows the assignments of different types of individuals or teams to each phase of enterprise integration program development and implementation. Titles alone indicate a very wide variance in personnel skills, interests and of the contribution they will make to the development of the program. Figure 5.1 indicates some of the formality which might be imposed on the program. However, the reader should carefully note the required involvement of management and other relatively non-technical personnel which will serve to severely limit the degree of mathematical or information technology based formality which may be imposed.

On the other hand, where computer interpretability of the information concerned is involved, a high degree of formality may be necessary. Figure 5.18 points out their re-
quirement from relevant parties involved. Figure 5.1 lists the type of modes and formalism involved. Figures 5.2 and 5.9 continue this discussion in relation to the formalism and types of models which will most probably be used with PERA.

Figure 5.20 is a reworking of earlier material, Figure 3.21 and Table 3.4 [19, 36, 37] to show an overall chart of the methodology described in the Guide [36]. Consideration of it will point out the wide diversity of tasks involved in the separate phases and the requirements that succeeding phases impose on any earlier ones. Table 5.2 defines each of the interfaces involved and their relationship to the overall task. Note that the symbol T on each phase location discussion indicates the Task carried out at that location on the Architecture while the symbols M and A indicates an abbreviated discussion of the Methodology involved there and the computer based and other Aids available to help accomplish the work.

5.6 Additional Architectural Considerations

5.6.1 Representation of Entity Interactions

Each separate organizational entity (enterprise or sub-unit of another larger enterprise) would have its own architecture as described by the Purdue Enterprise Reference Architecture. Enterprises with different missions will interact with each other as noted on the attached diagrams to accomplish a mission requiring multiple capabilities as described in these diagrams.

In Figure 5.21 the development of a particular manufacturing entity is shown [83]. In the center the integrated
FIGURE 5.18 WHO CARRIES OUT THE TASKS INVOLVED
FIGURE 5.19 TYPES OF MODELS AND TOOLS INVOLVED
FIGURE 5.20 THE PROGRAM INTERFACES
TABLE 5.2 DEFINITIONS OF PROGRAM INTERFACES INVOLVED IN AN ENTERPRISE INTEGRATION PROGRAM

<table>
<thead>
<tr>
<th>Interface</th>
<th>Capability and Tasks Involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.</td>
<td>Capability requirements for the <em>functional</em> information network.</td>
</tr>
<tr>
<td>B.</td>
<td>Capability requirements for the <em>functional</em> customer product and service network.</td>
</tr>
<tr>
<td>C.</td>
<td>The information network, the listing and interconnectivity of the information tasks — remains functional to this point.</td>
</tr>
<tr>
<td>D.</td>
<td>The customer product and service network, the listing and interconnectivity of the customer service tasks — remains functional to this point.</td>
</tr>
<tr>
<td>E.</td>
<td>The master plan — first phase of implementation, distribution of tasks of both types between humans and devices.</td>
</tr>
<tr>
<td>F.</td>
<td>Initial design (functional design or specification) of information systems architecture. Detail sufficient for completion of detailed design and for costing of overall project.</td>
</tr>
<tr>
<td>G.</td>
<td>Similar to (F) but for human and organizational architecture.</td>
</tr>
<tr>
<td>H.</td>
<td>Similar to (F) and (G) but for customer product and service equipment architecture.</td>
</tr>
<tr>
<td>J.</td>
<td>Overall detailed design document for the enterprise business entity. Detail sufficient for construction, purchase or other procurement of the total enterprise system.</td>
</tr>
</tbody>
</table>

(continue)
### TABLE 5.2 (CONT.) DEFINITIONS OF PROGRAM INTERFACES INVOLVED IN AN ENTERPRISE INTEGRATION PROGRAM

<table>
<thead>
<tr>
<th>Interface</th>
<th>Capability and Tasks Involved</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>K.</strong></td>
<td>Detailed design document for the information system architecture of the enterprise integration entity. Detail sufficient for construction, purchase or other procurement of this portion of the total system.</td>
</tr>
<tr>
<td><strong>L.</strong></td>
<td>Detailed design document for the human and organizational architecture of the enterprise integration entity. Similar to (K) in detail, etc.</td>
</tr>
<tr>
<td><strong>M.</strong></td>
<td>Detailed design document for the customer product and service equipment architecture. Of the enterprise integration entity. Similar to (K) and (L) in detail, etc.</td>
</tr>
<tr>
<td><strong>N.</strong></td>
<td>The completed integrated enterprise entity as developed under the enterprise integration program. Documentation of design, construction, and test details for legal acceptance by user group. Comprises the physical system itself plus the documentation noted above.</td>
</tr>
<tr>
<td><strong>P.</strong></td>
<td>Information systems architecture portion of completed enterprise integration program entity. Same type of documentation as (N), (Q) and (R).</td>
</tr>
<tr>
<td><strong>Q.</strong></td>
<td>Human and organizational architecture portion of completed enterprise integration program entity. Same type of documentation as (N), (P) and (R).</td>
</tr>
<tr>
<td><strong>R.</strong></td>
<td>Customer product and service equipment architecture portion of completed enterprise integration program entity. Same type of documentation as</td>
</tr>
</tbody>
</table>
(N), (P) and (Q).
THE CONCEPT OF INTERACTION OF ENTERPRISE ENTITIES

FIGURE 5.21 HOW DIFFERENT ENTERPRISE ENTITIES CAN INTERACT IN ORDER TO ACHIEVE THEIR INDIVIDUAL MISSIONS
enterprise in question such as is being described in this Guide is presented. On the left, we present the entity representing the enterprise integration planning team or the engineering design team who will be doing the Master Planning and Detailed Design work. Note that the design or planning team receives the management inputs required by the Guide. They produce the Master Plan and the Detailed Design documents for production by others. In this drawing the builder of the plant and/or plant equipment required is one or more separate entities which do the physical construction required. This methodology also shows the products developed by the so integrated plant.

Figure 5.22 shows the same information as presented by a Life-Cycle diagram but on an IDEF0 like presentation.

5.7 Promotion of a Minimum Formalism

So far, it can be seen that the following tasks have been carried out in this study to formalize PERA.

• Preparation of an Enterprise Integration Glossary.

• Provision for means or aids which help identify and prepare a processable syntax and semantics for the reference architecture.

• Provision for means or aids which help identify and define standard interfaces and related protocols for the reference architecture.

• Application of these findings to PERA to help formalize it and the Purdue Methodology.

The key issue here is that of the degree to which the architecture should be formalized for the specifications of an enterprise development program. The architecture formal-
ism should show the users the minimum that they have to do on system standardization under the reference architecture so as to establish an open system with the required connectivities as provided by information technologies across the interfaces involved to foreign systems or entities.
FIGURE 5.22 ANOTHER VIEW OF THE MATERIAL OF FIGURE 5.21 TO SHOW ENTERPRISE INTERACTION ON AN IDEF-TYPE DIAGRAM
6. THE USE OF THE AXIOMS OF ENGINEERING DESIGN AS A BASIS FOR DESCRIBING A GENERALIZED ENTERPRISE DEVELOPMENT PROGRAM'S CHARACTERIZATION AND EXECUTION

The Development of the Concepts of Enterprise Reference Architectures for Program Development (Type Two) and Physical System Descriptive Architectures of Enterprises and their Components (Type One) is described in this chapter based on the axiomatic approach proposed by Professor Suh in order to establish a theoretical basis for the Purdue Enterprise Reference Architecture and the Purdue Methodology.

6.1 The Axioms of Design

Professor Nam P. Suh, Director of the Laboratory for Manufacturing and Productivity, Department of Mechanical Engineering, Massachusetts Institute of Technology has written the following textbook [84]:

*The Principles of Design*

Oxford University Press

New York, NY

1990

in which he established a scientific base for engineering design.
This work is based on a fifteen year research by Professor Suh and his associates to establish the Axioms of Engineering Design. Axioms are formal statements either of what people already know or of the knowledge imbedded in many things that people do routinely. They cannot be proven, and they can only be disproved through the discovery of counterexamples.

The basic assumption of an axiomatic approach to design is that there exists a fundamental set or principles that determines good design practice. The only way to refute this assumption is to uncover counter examples that prove these axioms to be invalid. The knowledge in a given field can be axiomatized when a set of self-consistent logic based on the axioms can yield correct solutions to all classes of problems. So far, no-one has come up with evidence that the design axioms are invalid.

Suh initially proposed twelve axioms of engineering design and over the years he and his associates have reduced these to two which are:

**Axiom I:** The Independence Axiom

All of the Functional Requirements for a Design Must be Made to be Independent of Each Other.

**Axiom II:** The Information Axiom

That Design is Best Which Requires the Minimum Information for its Execution.

Prof. Suh later pointed out [85]:

"Axiom I distinguishes between good and bad design, or acceptable and unacceptable design. Axiom II is the criterion for selection of the optimum
design solutions from among those that satisfy Axiom I."

Under Suh's system, design is defined as the mapping process between the Functional Requirements (FRs) in the functional domain and the Design Parameters (DPs) in the physical domain. Their mapping relationships may be mathematically represented in a matrix form \([A]\) as follows:

\[
\{FR\} = [A] \{DP\} \tag{6.1}
\]

where

\[
\{FR\} \quad \text{the functional requirement vector;}
\]

\[
[A] \quad \text{the design matrix; and}
\]

\[
\{DP\} \quad \text{the design parameter vector.}
\]

The left-hand side of Equation (6.1) represents that "what we want in terms of design goals," and the right-hand side "how we hope to satisfy the DRs." [84, pp.54-55]

If the design matrix \([A]\) is a diagonal one, any single element of \(\{FR\}\) can be changed independently by just changing one element of \(\{DP\}\). A design that satisfies such a condition is defined as an uncoupled design.

The converse of an uncoupled design is the coupled design whose design matrix \([A]\) has mostly non-zero elements. That is, if we change just one design parameter in \(\{DP\}\), most probably every functional requirement will be changed at the same time. However, in this case, if we can arrange the \([A]\) into a triangle matrix and solve the equitation starting from the line where there is just one non-zero element (see also Section 7.4.3 and Equation 7.1), then this system is called a decoupled design which offers a conditional independence between the functional requirements. [84, pp. 54-56]
Among the corollaries of the two Axioms, there are two which can be related to the study of this thesis. They are [84, pp. 52–54]

- **Decoupling of Coupling Designs:**
  Decouple or separate parts or aspects of a solution if FRs are coupled or become interdependent in the designs proposed.

- **Use of Standardization:**
  Use of standardized or interchangeable parts if the use of these parts is consistent with FRs and constraints.

Although Prof. Suh expressed his Axioms in the mathematical form, many of his application examples in his book [84] and other related articles [85–87] were discussed qualitatively with help of the mathematical symbology he defined.

A small example in his book is a design of a can and bottle opener (see Figure 6.1). This bottle/can opener satisfies two FRs: (1) opens cans, and (2) opens bottles. If the requirement is not to perform these two functions at the same time, this physically integrated device satisfies Suh's first Axiom: they are independent FRs. [84, p. 51]

Prof. Suh especially stated [84, p. 51]:

"the identification of the ultimate optimal design having minimum information content cannot be guaranteed; also there is no method for generating all potential designs. We can only identify the best design on a relative basis among those proposed, using Axioms 1 and 2. However, the ability to eliminate unacceptable or unpromising ideas in their early stages enhances the creative part of de-
sign, and reduces the cost of development and the chance for failure."
FIGURE 6.1 TWO FUNCTION REQUIREMENTS OF THE OPENER ARE INDEPENDENT
6.2 Applicability of Suh's Axioms

Suh's book and its theme, the Axioms, will apply to the Enterprise Reference Architecture work if the following are considered:

1. What is really being done in the Architecture Project is to Design a Methodology for Enterprise Integration.

2. The Architecture is a Model of the Process or Methodology for Achieving Enterprise Integration.

3. It is interesting that the method of development in the Industry-Purdue University Consortium was the reverse. First, the Architecture Structure was developed and then the Methodology was written as a text explanation of the Architecture Sketch. Coincidentally, CIMOSA and GRAI apparently did likewise in their projects as well. This says something about how engineers' minds work.

4. Basically, the developers of PERA and the Purdue Methodology have summarized what they believe is a good engineering practice based upon general observation and put their summary into a framework and its guideline. The nature of this approach is developing a needed general theory to explain and to guide further development. The theoretical support needed to formalize this new theory must itself has a high generality and has a close connection with engineering practice.

The attached is an attempt at developing the Purdue Methodology for Enterprise Integration via the Axioms of Engineering Design. The system works as follows:
1. A minimum set of Functional Requirements is developed and the corresponding Design Parameters written for accomplishment of the Requirements.

2. Succeeding sets of Requirements and corresponding responses are then developed in a hierarchical fashion from the initial ones until the design description is finished as shown in the attached Figure 6.2.

3. The definitions accompanying Figure 6.2 present short titles for each Functional Requirement (FR) and its associated Design Parameter (DP).

The following notes are necessary for a fuller understanding of the axiomatic approach here:

1. There are four separate Functional Requirements which form the minimum set at the top level of the hierarchy as we see it. They relate as follows (see Figure 6.2 also):

(a) A Complete Life Cycle is necessary (FR(1)).

(b) The Methodology must be complete in that it must describe all systems components and their behavior (FR(2)).

(c) The objectives in executing the methodology are (1) the development of a Master Plan for the overall enterprise integration project and (2) the properly developed Master Plan must include a set of independent projects (FR(3)).

(d) These projects must be implementable as enterprise resources become available (FR(4)).
(e) This minimum set of Functional Requirements at the top level are all independent of each other.

2 In order to complete the development of the Methodology associated with the Purdue Enterprise Reference Architecture, a continuing set of subsidiary Functional Requirements and associated Design Parameters (1 for 1) under each main Functional Requirement must be developed in a recursive decomposition [84, pp. 36-39] until the Methodology is complete. This gives the hierarchical structures of Figure 6.2 where each of the boxes contains the serial number and short title of the Functional Requirement (FR) or Design Parameter (DP).

3 The term Design Parameter is used to name the corresponding means of satisfying the given Functional Requirement in order to match the nomenclature of Professor Suh’s book. This is not a good term to use for this example application, but it is necessary to follow the textbook as much as possible.

4 It is noted that the Master Plan and its associated Project Program Proposal are the Products for which the Methodology is being developed. In nature, they are part of a Design Parameter (see FR(3) and DP(3) in Table 6.21).

5 A Short Analysis of the results of this investigation is presented as a set of Questions and Answers in Table 6.24.
FIGURE 6.2  HIERARCHY OF FUNCTIONAL REQUIREMENTS AND DESIGN PARAMETERS FOR PURDUE METHODOLOGY FOR ENTERPRISE INTEGRATION
FIGURE 6.2 (CONT.) HIERARCHY OF FUNCTIONAL REQUIREMENTS AND DESIGN PARAMETERS FOR PURDUE METHODOLOGY FOR ENTERPRISE INTEGRATION
Professor Suh proposes an Information Theory type analysis of the total information necessary for a design as the answer sought with AXIOM II. He also points out though that many design decisions can be made qualitatively based on his Axioms if and when the basic knowledge of the subject is understood [84, pp. 190, 244]. Based on the understanding of Suh's Axioms and the Purdue Methodology, as noted in Question 4 in Table 6.24, the following is proposed to consider the information contents of the Purdue Methodology:

(a) Regardless of the Methodology used, the description of the Enterprise analyzed will require approximately the same overall amount of data.

(b) The important questions to be answered here are what are the critical questions asked in developing the Enterprise Models for further analysis. How many questions are asked? As noted in the Answer to Question 4, we believe only two are necessary in PERA. CIMOSA, for example, must ask several more to satisfy its required classifications.

6.3 Functional Requirements and Design Parameters of the Purdue Methodology

The contents of Functional Requirements and associated Design Parameters are presented in the following tables. Each pair of the Functional Requirement and Design Parameter is put together in a table which is named according to their symbols in Figure 6.2 and their short titles in Figure 6.2
are placed in parentheses after their long titles in the tables. The tables are arranged following the sequence of the boxes in Figure 6.2 similar to a depth-first pattern, which provide a suggested reading sequence to understand every aspect of the Purdue Methodology and its model PERA.

6.4 An Analysis: What is PERA and the Purdue Methodology?

By analyzing the work on PERA and the Purdue Methodology based on the Suh’s Design Axioms, the theoretical perspective and totality of the work can be validated and further summarized in a Q&A form in Table 6.24.
### TABLE 6.1 FR–(1) AND DP–(1)

**METHODOLOGY FUNCTIONAL REQUIREMENT 1**

**LIFE CYCLE OF THE ENTERPRISE DEVELOPMENT PROJECT AND ENTERPRISE ENTITY INVOLVED (Life Cycle)**

The Enterprise Development Program must cover the full Life Cycle of the subject Enterprise or part thereof under consideration.

<table>
<thead>
<tr>
<th>METHODOLOGY DESIGN PARAMETER 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIFE CYCLE OF THE ENTERPRISE DEVELOPMENT PROJECT AND ENTERPRISE ENTITY INVOLVED (Phases Named)</strong></td>
</tr>
</tbody>
</table>

The life history of Life Cycle of any Project for Enterprise Development and/or the associated Enterprise Entity must include the following six Steps or Phases in order to be complete: (see Figure 3.1)

1. Concept
2. Functional Analysis or Functional Definition
3. Functional Design
4. Detailed Design
5. Construction and Installation, or Manifestation
6. Operation to Obsolescence

Note: This is the first pair of FR and DP at the top level of the hierarchy to define the general concept of a full life cycle of an enterprise. In order to establish such a life cycle, this pair has six following FRs defined (see Figure 6.2).
TABLE 6.2 FR-(1-1) AND DP-(1-1)

<table>
<thead>
<tr>
<th>METHODOLOGY FUNCTIONAL REQUIREMENT 1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODELING THE LIFE HISTORY OF ANY PROJECT FOR ENTERPRISE DEVELOPMENT (Model Life Cycle)</td>
</tr>
<tr>
<td>The life history of any Project for Enterprise Development must be able to be modeled to show the interrelationship of the six Phases involved.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>METHODOLOGY DESIGN PARAMETER 1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODELING THE STRUCTURE OF THE ENTERPRISE DEVELOPMENT PROGRAM (Model Proposed)</td>
</tr>
<tr>
<td>The six separate regions and phases of the Enterprise Development Program as described in Methodology Design Parameter 1 are commonly executed in series. In their totality they will describe the life history of that part of the Enterprise involved in the Enterprise Development Program. This whole discussion can thus be named the LIFE CYCLE of the Enterprise or of that part of the Enterprise involved in the Development Program.</td>
</tr>
<tr>
<td>A model of the structure (i.e. interactions and progression of the Phases of the Program) can be represented as shown in the attached sketch (Figure 3.1). The meaning of the several areas and locations involved will be described in succeeding Functional Requirement/Design Parameter Pairs. Since this model represents the structure of the Program described, it can be termed as the Architecture of the Program.</td>
</tr>
</tbody>
</table>

Note: This pair and its following pair (Table 6.3) explain detailed concepts of Phases and Activities of the Phases.
TABLE 6.3  FR-(1-1-1) AND DP-(1-1-1)

METHODOLOGY FUNCTIONAL REQUIREMENT 1-1-1

DETAILING OF THE PROGRAM MODEL WITH THE PROGRAM ACTIVITIES (Model Detailed)

The model of the Life Cycle of the Enterprise Development Program must be detailed to show all activities occurring at each Phase of the Program.

METHODOLOGY DESIGN PARAMETER 1-1-1

DETAILING OF THE PROGRAM MODEL WITH THE PROGRAM ACTIVITIES (Proposed Detail Presented)

Figure 3.18 is one example of the possible detailing of the activities which are involved at each Stage of Progression of the Implementation of an Enterprise Development Program.

The diagrams of Figures 3.1 and 3.18 illustrating the Life Cycle of an Enterprise Development Program become the generic representation of the corresponding Life Cycle for any such program for any enterprise in any industry or field of endeavor since there is nothing in this discussion to limit the type of activity discussed, the size or characteristics of the enterprise or part thereof involved, or the industry or field of endeavor under discussion.

These diagrams are also the corresponding STRUCTURE for such a development program and thus their form can be called an ARCHITECTURE. Because of its generality and wide applicability it should be called an ENTERPRISE REFERENCE ARCHITECTURE.
### TABLE 6.4 FR-(1-2) AND DP-(1-2)

#### METHODOLOGY FUNCTIONAL REQUIREMENT 1-2

**THE FUNDAMENTAL BASIS FOR ANY ENTERPRISE DEVELOPMENT PROGRAM (Fundamental Basis for any Enterprise)**

Any Enterprise Development Program must be driven by the requirements which its parent enterprise has for its own future existence and/or growth. These requirements are reflected in the goals and aspirations which Enterprise Management has concerning the expected outcome of the proposed endeavor. These latter must then form the basis for planning for the proposed Enterprise Development Program.

#### METHODOLOGY DESIGN PARAMETER 1-2

**MANAGEMENT’S MISSION, VISION AND VALUES (Mission, Vision and Values)**

The first Action in establishing an Enterprise Development Program is to record Management’s Mission, Vision and Values for the program as expressed in their goals, objectives, mandates, critical success factors, etc., concerning that program. (See Section 3.1.1)

Note: This pair and its following pairs (Table 6.5-6.6) explain important concerns and actions related to the management involvement in the earlier stages of program development.
TABLE 6.5  FR-(1-2-1) AND DP-(1-2-1)

METHODOLOGY FUNCTIONAL REQUIREMENT 1–2–1
EXPANSION OF ENTERPRISE REQUIREMENT STATEMENTS
(Expansion of MVV)

The high level statements of goals, objectives, mandates, etc., for the proposed Enterprise Development Program as developed by Enterprise Management are by their very nature, abstract and general. They must be supplemented by appropriate material from external sources for use in later stages of the Program Development Process. Thus they must be expanded and further detailed to be readily useful by program development personnel in planning the proposed project.

METHODOLOGY DESIGN PARAMETER 1–2–1
EXPANSION OF ENTERPRISE REQUIREMENTS STATEMENTS
(Developing Policies, Requirements and Tasks)

Further stages of the detailing of management’s expectations for the proposed Enterprise Development Program should involve the conversion of the high level statements into a set of Policies concerning the means and expected results of the execution of the final implemented program. Preparation of the set of Policies is then followed by the preparation of a set of Enterprise Functional Requirements capable of satisfying the set of Policies. The Functional Requirements in turn must then be converted into a set of Tasks capable of satisfying these Functional Requirements. At each stage of this detailing process the higher level information available must be supplemented by external information sources to assume completeness of the next stage of development. The following diagram models the actions involved (see Figure 6.3 and Section 3.1.1).
FIGURE 6.3 DEVELOPMENT OF ENTERPRISE REQUIREMENTS
TABLE 6.6  FR-(1-2-2) AND DP-(1-2-2)

<table>
<thead>
<tr>
<th>METHODOLOGY FUNCTIONAL REQUIREMENT 1-2-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUPING OF MANAGEMENT RELATED ACTIVITIES</td>
</tr>
<tr>
<td>(Grouping of Management Functions)</td>
</tr>
<tr>
<td>The activities described under Management Mission, Vision and Values and the Policies Development part of the Program must be collectively grouped together and named as a distinct part or Phase of the Overall Enterprise Development Program. This is because of their specific relationship to management functions as contrasted with later parts of the Program.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>METHODOLOGY DESIGN PARAMETER 1-2-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUPING OF MANAGEMENT RELATED ACTIVITIES</td>
</tr>
<tr>
<td>(Naming of Concept Phase)</td>
</tr>
<tr>
<td>The work described under Mission, Vision and Values and Policies can be described as the Concept Phase of the Enterprise Development Program. As noted under Methodology Design Parameter 1, this Phase would be followed by a Definition or Functional Analysis Phase. The Concept Phase comprises the topics above the dashed line on Figure 6.3 and the so-indicated region on Figures 3.1 and 3.18.</td>
</tr>
</tbody>
</table>
## TABLE 6.7 FR-(1-3) AND DP-(1-3)

### METHODOLOGY FUNCTIONAL REQUIREMENT 1-3

**THE CLASSIFICATION AND FUNCTIONAL MODELING OF ENTERPRISE POLICIES, FUNCTIONAL REQUIREMENTS, AND TASKS**  
(Classifications of Policies, etc.)

The Policies, Functional Requirements and Tasks as developed under the previous Methodology Functional Requirement and Design Parameter pair must be able to be classified into a relatively small number of closely related Classes or Groups by using an appropriate criterion for such choice.

### METHODOLOGY DESIGN PARAMETER 1-3

**THE CLASSIFICATION AND FUNCTIONAL MODELING OF ENTERPRISE POLICIES, FUNCTIONAL REQUIREMENTS AND TASKS**  
(Control and Customer Service)

All of the Policies, Functional Requirements, Tasks, etc., carried out by an enterprise in its operation can be readily classified into two groups or classes:

1. Those related to the furthering of the well-being of the Enterprise, i.e., CONTROL of the enterprise’s operations in the broadest sense. These can also be expressed as Mission Support Tasks or Functions.
2. Those related to the accomplishment or fulfillment of the enterprise’s mission as established by management, i.e., those related to SERVICE to the Enterprise’s customers in supplying the goods and services needed by those customers.

(continue)
TABLE 6.7 (CONT.) FR-(1-3) AND DP-(1-3)

Those Tasks, Functions, etc., of Class 1 as developed under this Design Parameter all involve only the transformation and/or use of information (data, commands, etc.). All involve some aspect of CONTROL of the enterprise’s operation in the broadest sense.

Information storage and/or time-delay shall be considered a transformation in TIME to satisfy the meaning of this Design Parameter.

Those Tasks Functions, etc., of Class 2 as developed under this Design Parameter all involve only the transformation and/or use of material, energy or information (data, etc.) as needed to supply the products and/or services required for Enterprise Mission Fulfillment and/or Enterprise Customer Support. Note the distinct difference in the information definition and use between Classes 1 and 2. (see Section 3.1.1 and 3.1.4)

Note: This pair explains the mechanism of two-stream development in the earlier stages of PERA.
TABLE 6.8 FR-(1-4) AND DP-(1-4)

### METHODOLOGY FUNCTIONAL REQUIREMENT 1-4

**THE FUNCTIONAL MODELING OF ENTERPRISE TASKS AND FUNCTIONS (Modeling Technique for Tasks, etc.)**

The Functional Analysis of the Enterprise Program must be able to express the Enterprise's Constituent Activities as a set of Modularizable and Interconnected Tasks.

### METHODOLOGY DESIGN PARAMETER 1-4

**THE FUNCTIONAL MODELING OF ENTERPRISE TASKS AND FUNCTIONS (Modules and Their Connectivity)**

The transformation carried out by any of the Tasks, Functions, etc., under either class as defined in Functional Requirements 1-2-1 and 1-3 can be modeled as a Module showing all aspects of the inputs, outputs, parameters, etc., needed for the discussed transformations.

The resulting modules can be connected by their corresponding inputs and outputs to form multi-task Functional Modules. These can also be further connected for forming an overall Functional Network of these Tasks to give a Functional Model of the totality of the Tasks of either Class in that part of the Enterprise studied in the Enterprise Development Program under consideration. (Section 3.1.3)

Note: This pair and its followings (Table 6.9 and 6.10) illustrate the method of modularity of PERA.
TABLE 6.9 FR-(1-4-1) AND DP-(1-4-1)

METHODOLOGY FUNCTIONAL REQUIREMENT 1-4-1

THE FUNCTIONAL MODELING OF ENTERPRISE TASKS AND FUNCTIONS (Generic Module Model)

A method for presenting all aspects of the Functional Analysis of each of the separate Tasks and Functions of the Enterprise must be developed.

METHODOLOGY DESIGN PARAMETER 1-4-1

MODELING OF TASKS AND FUNCTIONS AS MODULES (Proposed Modules)

One possible form of the Modules used to model the individual Tasks and Functions as detailed in the discussion of this Functional Requirement and its subsidiaries is presented in Figures 3.7 for the Functional Analysis Case. The corresponding Implementation Descriptive Form would be as Figures 3.11-3.14.
METHODOLOGY FUNCTIONAL REQUIREMENT 1-4-2

THE FUNCTIONAL MODELING OF ENTERPRISE TASKS AND FUNCTIONS (Overall Functional Model)

An overall Functional Model of the enterprise must be able to be developed by connecting the Overall Functional Models of each of the two Classes of Functions of the enterprise together.

METHODOLOGY DESIGN PARAMETER 1-4-2

THE FUNCTIONAL MODELING OF ENTERPRISE TASKS AND FUNCTIONS (Network Models)

An overall Functional Model of that portion of the enterprise involved in the Enterprise Development Program under discussion can be obtained by connecting the two separate Network Models of each Class of Tasks, Functions, etc. This would be done by connecting them through the plant status information presented to the CONTROL side from MISSION FULFILLMENT and by the corresponding control activation information presented to MISSION FULFILLMENT from CONTROL. (see Section 3.1.1)
TABLE 6.11 FR-(1-5) AND DP-(1-5)

METHODOLOGY FUNCTIONAL REQUIREMENT 1-5

THE CLASSIFICATION AND FUNCTIONAL MODELING OF ENTERPRISE REQUIREMENT AND TASKS
(Functional Only, No Implementation)

As all of the Enterprise Requirements and Tasks discussed under Methodology Functional Requirement 1-3 and its subordinate Requirements are only functional in nature they must be able to be discussed without any reference to their ultimate means of implementation.

METHODOLOGY DESIGN PARAMETER 1-5

THE CLASSIFICATION AND FUNCTIONAL MODELING OF ENTERPRISE REQUIREMENTS AND TASKS
(Functional Analyses)

All of the activities discussed under Functional Requirement 1-3 and its Subsidiary Requirements can be considered as a FUNCTIONAL ANALYSES or DEFINITION PHASE of the process of development of the discussed program.

Note: Introduction to the concept of the Functional Analysis of PERA.
TABLE 6.12  FR-(1-6) AND DP-(1-6)

<table>
<thead>
<tr>
<th>METHODOLOGY FUNCTIONAL REQUIREMENT 1-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>THE VIEWS OF AN ENTERPRISE DEVELOPMENT PROGRAM</td>
</tr>
<tr>
<td>(Views of the Architecture)</td>
</tr>
<tr>
<td>The Functional Analysis versus the Implementation aspects of the Enterprise Development Program must be organized as two separate parts of the Program.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>METHODOLOGY DESIGN PARAMETER 1-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>THE VIEWS OF AN ENTERPRISE DEVELOPMENT PROGRAM</td>
</tr>
<tr>
<td>(Views of Development)</td>
</tr>
<tr>
<td>Once the Functional Analysis Tasks of the Methodology as described up to this point, has been completed, any succeeding work on the further prosecution of the Enterprise Development Program will involve aspects of its eventual implementation and operation. Thus, the overall work can be considered as involving only two views: FUNCTIONAL ANALYSIS and IMPLEMENTATION.</td>
</tr>
</tbody>
</table>

Note: Introduction to the concept of the two basic Views of development in PERA.
TABLE 6.13  FR-(2) AND DP-(2)

METHODOLOGY FUNCTIONAL REQUIREMENT 2

COMPLETENESS OF THE ENTERPRISE DEVELOPMENT PROGRAM
IN TERMS OF THE COMPONENTS OF THE ENTERPRISE ITSELF
(Complete in terms of Components and Behavior)

An Enterprise Development Program can be complete only when all aspects of the program in terms of all of the Components of the Enterprise and their Individual Behavior have been considered.

METHODOLOGY DESIGN PARAMETER 2

COMPLETENESS OF THE ENTERPRISE DEVELOPMENT PROGRAM
IN TERMS OF THE COMPONENTS OF THE ENTERPRISE AND THEIR BEHAVIOR (Activation)

Further Analysis and Design Activities concerning the several Stages or Phases of Enterprise Development as already described in Functional Requirement 1 and its Subsidiaries must include all activities of the enterprise during these further Stages or Phases and the Physical and Informational Components of the Enterprise which are involved.

Note: This pair and its four subsets (see Figure 6.2 (cont.)) describe the concept of completeness of development programs and how the Methodology and PERA develop a complete enterprise program.
### TABLE 6.14 FR-(2-1) AND DP-(2-1)

#### METHODOLOGY FUNCTIONAL REQUIREMENT 2-1

**EXPRESSION OF THE PLACE OF THE HUMAN IN AN ENTERPRISE THROUGH THE ENTERPRISE DEVELOPMENT PROGRAM**  
(All Aspects of Human)

The Program must reflect all aspects of the place of the Human in the subject Enterprise.

#### METHODOLOGY DESIGN PARAMETER 2-1

**EXPRESSION OF THE PLACE OF THE HUMAN IN AN ENTERPRISE THROUGH THE ENTERPRISE DEVELOPMENT PROGRAM**  
(Human in Implementation Only)

The place of the Human in the final form of that portion of the Enterprise involved in the Enterprise Development Program need not be considered in the Function Analysis or Definition Phase of the Program. However, the first duty under IMPLEMENTATION is to establish which of the Tasks, Functions, etc., developed under the FUNCTIONAL VIEW will be carried out by Humans and which will be carried out by Machines, etc. All succeeding design, etc. work must be based upon this resulting distribution of system tasks between Humans and Machines.

This action will be true independently for all of the Tasks, Functions, etc. under both classifications used in the Functional Analysis. (Section 3.1.2)

Note: This pair and its followings describe the principles and process of introducing human and organizational mechanism into program development.
TABLE 6.15  FR-(2-1-1) AND DP-(2-1-1)

METHODOLOGY FUNCTIONAL REQUIREMENT 2-1-1

ESTABLISHING THE PLACE OF THE HUMAN IN AN ENTERPRISE
(Boundaries of Human Activity)

The place of the Human in the enterprise must be able to be
established based upon the following factors:

1. The ability of Humans to carry out a particular Task
   or Function.

2. The associated relative economic and social costs of
   using Humans for Tasks which may be carried out by
   either Humans or Machines.

METHODOLOGY DESIGN PARAMETER 2-1-1

ESTABLISHING THE PLACE OF THE HUMAN IN AN ENTERPRISE
(Proposal for Establishment)

1. Some of the Tasks and Functions developed in the
   Functional Analysis Phase must be executed by Humans
   because they involve innovation or other capabilities which
   cannot yet be carried out by Machines.

2. Some of the Tasks and Functions developed in the
   Functional Analysis Phase must be executed by Ma-
   chines because they involve requirements of
   strength, speed, dexterity, cognition, or other fac-
   tors beyond the capabilities of Humans but within
   those of some types of Machines.

3. Some of the Tasks and Functions developed in the
   Functional Analysis Phase can be executed either by
   Machines or by Humans.

These divisions of assignment of Tasks and Functions can be
used to establish the boundaries of consideration of the
place of Humans and other entities of the overall Enterprise
in the Implementation Phase of the Development Program.
### TABLE 6.16 FR-(2-1-2) AND DP-(2-1-2)

**METHODOLOGY FUNCTIONAL REQUIREMENT 2-1-2**

**ESTABLISHING THE PLACE OF THE HUMAN IN AN ENTERPRISE**  
(Determination of Place of Human)

A method or procedure must be developed for the implementation steps of the program to separate those tasks or functions assigned to humans from those assigned to machines (i.e., where is automation or mechanization applied?).

**METHODOLOGY DESIGN PARAMETER 2-1-2**

**ESTABLISHING THE PLACE OF THE HUMAN IN THE ENTERPRISE**  
(Method of Distribution Detailed)

The Procedure for separating Human implemented Tasks from those assigned to Machines can be as follows:

1. Consider all of the Tasks, Functions, etc., developed under Class 1 (Information or Mission Support, see Table 6.7 of the Functional Analysis to be arranged in a continuum according to their requirement concerning their assignment to Information System Machine Implementation, Human Implementation or both with the degree of required Machine Implementation increasing toward the left and the degree of required Human Implementation increasing toward the right. Those in the middle of the continuum will then presumably be able to be implemented by either means.

2. There will be a line or point somewhere on the right side of the continuum which will mark the end of the potential application of Machines, i.e., all further consideration must be of Humans. This line or point will be considered the Automatability Line or Limit.  

(continue)
3. Likewise, there will be a line or point somewhere on the left side of the continuum which will correspondingly mark the limit of potential assignment of Humans to those Functions, i.e., all further consideration must be for Machines. This line or point will be considered the Humanizability Line or Limit.

4. All Tasks or Functions falling between these two lines or Limits can presumably be implemented by means of the use of either Humans or Machines.

5. Repeat the above analysis for all the Tasks, Functions, etc., developed under Class 2 (Provision of Customer Products or Services [otherwise known as Mission Fulfillment], see Table 6.7) except reverse the assignment. That is, put required Machine Implemented Tasks on the right and Human Implemented Tasks on the left. The Automatability and Humanizability lines will be established as before but their positions will be reversed.

6. In each case every point in the region between the two lines for each Class will be technologically feasible to be implemented by either Machine or Human based means. Thus management is free to establish the actual desired boundary between the separate groupings anywhere in either of these regions.

7. The actual placement chosen can then depend upon economic, social or political requirements rather than purely technological ones. The resulting line will be called the EXTENT OF AUTOMATION LINE.

(continue)
8. All Tasks to the left of the Extent of Automation Line in the Information or Mission support group of Tasks when implemented become the INFORMATION SYSTEMS ARCHITECTURE.

9. All tasks to the right of the Extent of Automation Line in the Customer Service or Mission Fulfillment group of Tasks when implemented become the CUSTOMER PRODUCT or SERVICE EQUIPMENT ARCHITECTURE (Manufacturing Equipment Architectures for manufacturing enterprise).

10. Place the remaining Tasks (those to the right of those placed in the Information Systems Architecture plus those to the left of group placed in the Customer Service Equipment Architecture) together. The boundaries of this remaining group will be the two Extent of Automation Lines. All those Tasks between the two Extent of Automation Lines can then be grouped together as the HUMAN AND ORGANIZATIONAL ARCHITECTURE, since all of these Tasks (regardless of Class 1 and 2) will have been assigned to Humans for their implementation and execution.

11. The distribution of the several separate Tasks and Functions from the Functional Analysis study between the Three implementation architectures as described by the Functional Requirement and associated Design Parameters can be readily illustrated as shown in Figure 3.5.
TABLE 6.17 FR-(2-2) AND DP-(2-2)

METHODOLOGY FUNCTIONAL REQUIREMENT 2-2

SEPARATION OF THE NON-HUMAN COMPONENTS OF THE ENTERPRISE IN TERMS OF THEIR CONSIDERATION IN THE ENTERPRISE DEVELOPMENT PROGRAM
(Distribution of other Components)

A procedure must be prepared for considering all of the Physical and Informational Components of the Enterprise (other than the Human Components already described) in carrying out the Enterprise Development Program.

METHODOLOGY DESIGN PARAMETER 2-2

SEPARATION OF THE NON-HUMAN COMPONENTS OF THE ENTERPRISE IN TERMS OF THEIR CONSIDERATION IN THE ENTERPRISE DEVELOPMENT PROGRAM
(Method Proposed)

The earlier separation in terms of the Mission Support (Control) and Mission Fulfillment as described in Functional Requirement Number 1-3 and its Subsidiaries provides the basis for further consideration. The Non-Human Entities responsible for Mission Support (Control) Tasks will be classified as an Information Systems Group or Architecture (to capture the structure of their organization and interaction together). Those Non-Human entities responsible for carrying out their Mission, Fulfillment Tasks will be Classified as a Customer Product or Service Architecture (to capture the structure of their organization and interaction together).

(See Section 3.1.2 and Section 3.1.4)

Note: This pair and its following one discuss the method of developing three Implementation Architectures of PERA.
TABLE 6.18 FR-(2-2-1) AND DP-(2-2-1)

METHODOLOGY FUNCTIONAL REQUIREMENT 2-2-1

ESTABLISHMENT OF THE IMPLEMENTATION ARCHITECTURES OF THE PROGRAM FOR ENTERPRISE DEVELOPMENT
(Naming of other Component Implementation)

Major differences in Task Type and technologies involved require that the Three Types of Tasks described by Functional Requirements 2-1 and 2-2 and their associated Design Parameters be Implemented by different means and therefore each type of Task Implementation must be treated separately in this Overall Methodology.

METHODOLOGY DESIGN PARAMETER 2-2-1

ESTABLISHMENT OF THE IMPLEMENTATION ARCHITECTURES OF THE PROGRAM FOR ENTERPRISE DEVELOPMENT
(Implementation Architectures)

If no Humans were involved there would be ONLY two different types of implementations (corresponding to the two Classes of Tasks in Functional Analysis) involved in the further execution of the Enterprise Development Program:

1. Those Tasks related to Class 1 — Information (Mission Support Tasks)

2. Those Tasks related to Class 2 — Customer Product or Service (Mission Fulfillment Tasks)

However, separation of the Tasks of the Functional Analysis according to their required means of implementation (Humans or Machines) will result in three separate groups of Tasks:

1. Those Information tasks (Control or Mission Support) to be implemented by information handling machines (computers, communications, controllers, etc.)
2. Those Customer Product or Service Tasks (Mission fulfillment) to be implemented by manufacturing equipment, etc.

3. Those tasks on both sides of the Functional Analysis Phase to be implemented by Humans.

Even though the two types of Tasks to be implemented by Humans are different, the social, etc., requirements of Humans are common. Therefore, these should be grouped together, hence three Implementation divisions result. Since the organization of the Machines, Humans, etc., to accomplish the assigned Tasks will result in Structures (layouts, flow patterns, organizations, etc.) their individual descriptions can be called Architectures. Thus there will be three of these Architectures involved:

1. Information Systems Architecture
2. Human and Organizational Architectures
TABLE 6.19  FR-(2-3) AND DP-(2-3)

METHODOLOGY FUNCTIONAL REQUIREMENT 2-3

THE PHASES OF IMPLEMENTATION OF THE ENTERPRISE DEVELOPMENT PROGRAM (Phases of Implementation)

The Methodology adopted must emphasize similarities in the Implementation Steps of the enterprise Development Program.

METHODOLOGY DESIGN PARAMETER 2-3

THE PHASES OF IMPLEMENTATION OF THE ENTERPRISE DEVELOPMENT PROGRAM (Phased Implementation)

Once the boundaries of the three separate Implementation architectures have been established by technological and management decree assignment, the planning and execution of the implementation of the Enterprise Development Program can proceed further.

The Implementation View can be arbitrably divided into four regions or Phases to be labeled and described as follows:

1. The Specification or Functional Design Phase. This will include the Task Assignments described in Functional Requirement 2 and its Subsidiaries. It will also include the determination of the Functional Design of the execution of each Task involved in each architecture to the point of writing the functional Specification defining each required machine or the skills necessary for each required Human worker.

2. The Detailed Design Phase. This involves the conversion of the Functional Design specification into a design capable of allowing the actual construction or purchase of Machines or the procurement and/or training of appropriately skilled Human employees.
TABLE 6.19 (CONT.)  FR-(2-3) AND DP-(2-3)

3. The Manifestation or Construction, Testing and Commissioning Phase (shortened to Manifestation Phase). Here all component parts are bought, assembled, tested and approved; all required individuals are hired and trained; and finally the whole Enterprise is certified for operation.

4. The Operations Phase involving the actual operation of the completed Enterprise System to produce the desired customer goods or services, including continued improvement and/or optimization of its operation through and until obsolescence.

Note: Introduction to the Phases in the Implementation View of PERA.
TABLE 6.20  FR-(2-4) AND DP-(2-4)

METHODOLOGY FUNCTIONAL REQUIREMENT 2–4

DESIGNATION OF THE PHYSICAL SYSTEMS RESULTING FROM USE OF THE IMPLEMENTATION ARCHITECTURES IN RELATION TO THE ENTERPRISE REFERENCE ARCHITECTURE (Type One vs. Type Two Architectures)

The Individual Physical systems resulting from the use of each of the Implementation Architectures will themselves each have a Physical structure. While this Structure can also be described by an Architecture of the placing and interconnections of its physical components such Architectures are fundamentally different from the overall Enterprise Reference Architecture in its meaning, etc. There must be a means of expressing this difference in their naming.

METHODOLOGY DESIGN PARAMETER 2–4

DESIGNATION OF THE PHYSICAL SYSTEMS RESULTING FROM USE OF THE IMPLEMENTATION ARCHITECTURES IN RELATION TO THE ENTERPRISE REFERENCE ARCHITECTURE (Classification of Reference Architectures)

The earlier Functional Requirements and their subsidiaries have established three so-called Implementation Architectures and discussed the execution of that part of the Enterprise Development Program contained within each one. The result of each case is a developed Structure of Components, Tasks and their execution capable of satisfying its share of the management requirements voiced at the beginning or Concept Phase of the overall program.
A description of the Structure involved as a result of the execution of each Implementation Architecture is itself an Architecture in the usual sense of the word — a description of the physical structure of a particular device, system, etc.

These, of course, are much different from the Enterprise Reference Architecture of Functional Requirement 1-1 and its Subsidiaries since the former are structures of actual physical entities rather than structures of projects or programs. The former type, the Structures of Systems, will be termed Type One Architectures while the latter, the Program Development Structure, will be termed a Type Two Architecture.

Note: This pair is an introduction to two different Reference Architectures involved in any enterprise development.
### TABLE 6.21 FR-(3) AND DP-(3)

#### METHODOLOGY FUNCTIONAL REQUIREMENT 3

**MASTER PLAN AND PROGRAM PROPOSAL PREPARATION**  
*(Master Plan and Program Proposal)*

The first overall activity of the execution of an Enterprise Program Methodology must be the preparation of a Master Plan and Associated Program Proposal for the Enterprise Entity involved as described in the first three Phases of the Methodology.

#### METHODOLOGY DESIGN PARAMETER 3

**MASTER PLAN AND PROGRAM PROPOSAL PREPARATION**  
*(Master Plan Preparation)*

The Purdue Enterprise Reference Architecture and Methodology should be first applied by using organization as the basis for planning the proposed implementation project. The First Three Phases of the Methodology outline the procedures for preparing a Master Plan for the Overall Project to assure project completeness, and satisfaction of Management Requirements for performance and economic gain and satisfaction that all pertinent consideration have been answered.

The Master Plan should include a Transition Plan and a Program Proposal. The Transition Plan outlines a viable series of Steps or Projects which together will implement the Master Plan for the Project. Each Project must be within the Resource Capability of the Company to implement in a reasonable time. The Program Proposal prioritizes these Projects in terms of the sequence of implementation in the overall Program. Implementation of each of the separate Projects as implemented in turn will assure that the overall Program is complete when all have been implemented. (see Section 3.2)

**Note:** This pair describes the requirements for Master Plan and its Preparation.
TABLE 6.22  FR-(3-1) AND DP-(3-1)

<table>
<thead>
<tr>
<th>METHODOLOGY FUNCTIONAL REQUIREMENT 3-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOOLS FOR USING THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE AND METHODOLOGY (Tools)</td>
</tr>
<tr>
<td>Each of the Nodes or Regions on the Graphical Representation of the Purdue Enterprise Reference Architecture represents a set of Tasks or Operations to be carried out during the pursuit of the Program. Many, if not all, of the Tasks must be able to make use of computer based aids in their execution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>METHODOLOGY DESIGN PARAMETER 3-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOOLS FOR USING THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE AND METHODOLOGY (Architectural Tooling Proposed)</td>
</tr>
<tr>
<td>Figure 3.21 and Table 3.4 present an outline of the Tasks and Associated Tools belonging to each Node or Area of the Purdue Enterprise Reference Architecture and its Associated Methodology. The Table can be populated with actual Product Offerings of Computer Software Houses to give a full set of Computer Aids available for these Tasks.</td>
</tr>
</tbody>
</table>

Note: This pair describes the requirements for development tools.
### TABLE 6.23 FR-(4) AND DP-(4)

**METHODOLOGY FUNCTIONAL REQUIREMENT 4**

**REQUIREMENTS FOR ACTUAL PHYSICAL IMPLEMENTATION OF THE ENTERPRISE PROGRAM**
*(Implementation of Program Proposal)*

The Implementation of the Program Proposal must proceed only as Enterprise resources permit.

---

**METHODOLOGY DESIGN PARAMETER 4**

**REQUIREMENTS FOR ACTUAL PHYSICAL IMPLEMENTATION OF THE ENTERPRISE PROGRAM (Method Detailed)**

Management will assure themselves that adequate resources in Money, Properly-Trained Personnel, and Time to complete the required Tasks are available before beginning any particular Enterprise Integration Project. They should also be assured that each such Project fits the previously prepared Integration Program Proposal and Master Plan. *(see Section 3.2)*

Note: This pair substantiates the concerns for resource availability in enterprise development.
TABLE 6.24 WHAT IS PERA AND ITS ASSOCIATED PURDUE METHODOLOGY

QUESTION 1. What have been really accomplished in developing PERA and the Purdue Methodology?

ANSWER 1. The DESIGN and development of an overall generic methodology for Enterprise establishment and operational fundamental.
Not enterprise integration — that is only a small part of it!
But something much bigger, a generic methodology for developing any and all enterprises!

QUESTION 2. What is the Purdue Enterprise Reference Architecture (PERA)?

ANSWER 2. PERA is a model of the methodology or PROCESS of carrying out the development and operation of any enterprise. It is a picture of the STRUCTURE of the steps involved in the methodology or process and of their interrelationships as these occur in that process. As such it is an ARCHITECTURE and because of its truly generic nature, it is, in fact, an Enterprise Reference Architecture.

QUESTION 3. How can we know this is true?

ANSWER 3. By Suh’s Axioms of Design in comparison with other proposed architectures.
The Axioms of Design are:
1. The Requirements Must be Independent,
2. The Information Required Must be the Minimum Possible.
The requirements for the Purdue Methodology are as follows:

1. The Program Must Cover the Full Life Cycle of the Subject Enterprise or Part Thereof.

2. The Enterprise Development Program Must be Based upon the Requirements of the Parent Enterprise for its own Future Existence and/or Growth. These Requirements are Reflected in the Goals and Aspirations of Enterprise Management Concerning the Expected Outcome of the Proposed Endeavor.

3. The Functional Analysis of the Enterprise Program can be Divided into Those Functions Related to the Control of the Enterprise itself and Those Related to the Accomplishment (Manufacture, etc.) of its Products or Services.

4. The Enterprise Development Program Must be Complete in Terms of the Components of the Enterprise itself.

5. The Functional Analysis of the Enterprise Program Must be Able to Express the Enterprise’s Constituent Activities as a Set of Modularizable and Interconnected Tasks.

6. The Program Must Reflect All Aspects of the Place of the Human in the Subject Enterprise.

7. The First Activity of the Enterprise Program Methodology is the Preparation of a Master Plan and Associated Program Proposal as Described in the First Three Phases of the Methodology.

8. The Implementation of the Program Proposal is carried out as Enterprise Resources Permit.
TABLE 6.24 (CONT.) WHAT IS PERA AND ITS ASSOCIATED METHODOLOGY

QUESTION 4. How does PERA and its associated methodology answer the requirements of these Axioms?

ANSWER 4.

1. The members of the preceding set of requirements for PERA and its Methodology are Independent of each other.

2. PERA uses the minimum information possible. It requires only two items of information:
   (a) Is a particular task functionally a part of CONTROL or of CUSTOMER SERVICE?
   (b) Is that task implemented by humans or machines?

The answer to the first Question above enables the PERA Functional View.

The answer to the second Question, 2(b), enables the PERA Implementation View.

SUMMARY

1. THE WORK IS DESIGNING A METHODOLOGY.

2. THEREFORE SUH'S AXIOMATIC DESIGN THEORY APPLIES TO THE RESULTING DESIGN ENDEAVOR.

3. PERA IS A MODEL OR ARCHITECTURE OF THAT METHODOLOGY
7. EXTENSION OF PERA

Based upon studies in the previous chapters, further extensions of PERA will be used in this chapter to develop a genealogical tree of enterprise development. Also program iteration will be presented within the lifecycle of PERA. The underlying meanings and importance of the results will be listed and explained.

7.1 Comparison of Models of System Lifecycle

If we consider any enterprise as a typical system which comprises an organized collection of people, machines and related processes needed to accomplish certain required functions of customer product or service operations, PERA can be considered as the conceptual model of that system's lifecycle. In the literature, there are many discussions about system life cycles which have been based on studies of information systems development. These have variously been carried out under the name of systems analysis and design [88–94], or systems engineering [95, 96]. The concerns related to the system lifecycle in these publications include:

- Process or program planning and risk management [95, 97];
- Project management [89, 90, 94–96];
• Project cost [91, 97]; and
• System development activities and the tools involved [88–90, 92, 93].

There are two distinctive features between the lifecycle models noted above and PERA. First, these models, despite their labels, do not cover the full lifecycle of a generic system development. Especially missing, in most cases, is that part of lifecycle where the initial ideas or concepts on the future subject systems requirements are collected from the enterprise’s management and enterprise’s customers involved and where these requirements are reorganized for the purpose of the detailing of further tasks and specifications analysis. By contrast, the developers of PERA [36] realized that for a successful implementation of an enterprise development program, top management involvement is the first critical factor. Not only is a persistent commitment needed from the top management, but also an active, ongoing involvement in program planning is needed as well.

Second, because these models do not show either an explicit demonstration of human and organizational functions or a functional split between the concept of system mission fulfillment and the concept of system mission support throughout the structure of these lifecycle models, only a single stream of system development is presented in these models. It does not matter here whether they have used the Waterfall Model (see Figure 7.1 [95, 98]) or the Spiral Model (see Figure 7.2 [95, 97]) or some other process model or methodology.

Although not all development in the Spiral Model is carried out in the sequence of a rigidly phased approach [88,
FIGURE 7.1 THE WATERFALL MODEL
FIGURE 7.2 THE SPIRAL MODEL
the iterative program development can still be seen as a refining process working on a single subject system. If we consider the case of a complete enterprise development program rather than only the implementation of an information system, such a single stream model will not be capable of illustrating the total complexity involved such as the multidisciplinary approaches which are required in the usual enterprise development program.

The central issue in developing any enterprise through its total lifecycle is the management of the complexity involved [21, 95]. Of course, the importance of information systems in enterprise integration cannot be ignored. However, as pointed out before in Chapter 2, the tasks involved in an enterprise development program will probably be much more complicated than those required in the implementation of an information system alone. The developers of PERA have realized through their industrial and academic experience the magnitude and complexity of such an enterprise development program [36].

PERA emphasizes that the most significant goal of enterprise engineering [2, 22] should be to engineer the total enterprise as a whole throughout its complete lifecycle. As shown in Chapter 6, the Purdue Methodology, an overall generic methodology, and PERA, a lifecycle model applicable for any and all enterprise development, have been designed and developed according to Professor Suh's axiomatic approach [84]. Based on the discussion carried out in Chapters 3, 5 and 6, the contributions of PERA to the generic system lifecycle model can be summarized as follows (see [99] also):
• Definition of the full lifecycle of system development in general engineering practice;

• Separation of the functional analysis phase into a two-parallel-stream development where the concepts of system mission fulfillment and system mission support are introduced to reveal that two and only two sets of generic system functional tasks are used throughout its lifecycle; and

• Definition of the place of the Human in system implementation to introduce three and only three implementation architectures, i.e., the Information System Architecture, the Human and Organizational Architecture and Customer Product and Service Architecture, in order to carry out the remaining system lifetime phases and functions.

Based on the general framework of PERA, further explanation and use have been made of the enterprise program lifecycle in [19, 36] and in this report as follows:

• Major tasks or activities involved in the lifecycle of an enterprise development program have been illustrated (see Figure 3.18 and reference [19]);

• A step-by-step program planning and implementation guide have been developed (see Figure 3.20 and [36]);

• Development and implementation aids or tools available for the conduct of the program have been listed (see Table 3.4);

• Identification of the significance of inter- and intra-architectural interfaces has been established (see Figures 5.4, 5.10 and 5.11);
• Identification of Program Interfaces has been made (see Figure 5.20);

• Illustration of the place of all human program participants has been demonstrated (see Figure 5.18);

• Identification of the probable formalism of program model and tools required was made (see Figure 5.19); and

• Extent of formalism possible in program products has been listed (see Figures 5.1 and 5.2).

As noted in the above list, almost every concern related to the common overall system lifecycle as shown at the beginning of this section can be studied in the bigger domain of an enterprise development program by adopting the lifecycle model and methodology of PERA as the basis for that study. For example, if relevant cost coefficients are available for the cost estimation of program activities and tools, etc., the program lifecycle cost can be estimated. If information on the personnel productivity of the program participants can be obtained, it will be a great help for the planning activities in the project management involved. With help from the Guide [36], development of an enterprise's Master Plan will be properly oriented to the minimum of resources necessary.

Furthermore, because PERA is a truly generic lifecycle model of any and all enterprise development programs as noted before, it is able to capture and express more complicated concepts inherent in enterprise development programs. In the next section, an enterprise genealogy will be developed for presenting enterprise entities of all types and
the relationships between them by using the lifecycle model of PERA.

7.2 An Enterprise Genealogy

7.2.1 Types of Enterprise Entities Related to Enterprise Life Cycles

As pointed out by PERA, every enterprise in order to justify its existence must have a mission [19]. In this vein, enterprises may share common goals among themselves while each has its own specific goals as well. For example, individuals and departments within a single company may have to address some particular requirement or task of their own in addition to the coordination required between them. Figure 7.3 [19] shows how PERA separates a single overall Enterprise Entity into several subentities by going through a separation of the overall mission into its different components. So far in this report, the treatment for enterprise development programs has been focused on integration within one single enterprise program which can be called Intraenterprise Integration as in reference [100]. The relationships between coordinating enterprises or those involved in Interenterprise Integration [100] have as yet been only briefly considered in this report.

When the discussion on program interfaces in Chapter 5 was extended to cover the interactions between enterprise entities, Figure 5.21 [83] showed three different enterprise entities acting together through the interfaces between them. These three entities were:

- An Engineering (Design) Entity which was responsible for the design of a subject enterprise;
• A Manufacturing Entity which was the subject enterprise being developed; and
FIGURE 7.3  SUBDIVISION OF THE ENTERPRISE INTO SEPARATED ENTERPRISE BUSINESS ENTITIES
• A Construction Entity which was responsible for the physical manifestation of the subject enterprise.

The Engineering Entity and the Construction Entity provided their Customer Products and Services to different stages/phases in the lifecycle of the Manufacturing Entity as shown there.

To illustrate further, Figure 5.21 has pointed out that the operations carried out at a certain phase of the lifecycle of an enterprise development program may actually involve the Customer Product(s) or Service(s) which can be delivered by some other substantive Enterprise Entity or Entities. What kind of Enterprise Entities will deliver these products or services is related to the particular phase of the receiving Entity which is involved.

At its final phase, the Operations Phase, any Manufacturing Entity is itself a substantive Enterprise Entity which can then contribute to another downstream entity. However, it itself is the result of all the efforts made earlier by other entities during the whole of its own development program. Thus, on the one hand, as noted above, each phase of development of the new Enterprise Entity needs another separate Enterprise Entity to provide the needed products and services required at that point. On the other hand, whatever concrete products or services the Enterprise Entity first developed delivers to its own customers must then be based on their own development life history which can also be described by the framework of PERA.

In this context, the following four different classes of Enterprise Entities can be named in order to develop a genealogical tree of the enterprise development programs and
relationships between any and all Enterprise Entities which might be involved [101]:

1. A Strategic Management Enterprise Entity, or #1 Enterprise Entity, which is responsible for the strategic planning of the subject manufacturing Enterprise Entity, for example, as shown in Figure 5.18, the product of Enterprise Entity #1 may be produced by the upper management and the steering committee. Enterprise Entity #1's relationships with the other Enterprise Entities involved into the program can be as shown in Figure 7.4.

2. The Engineering Enterprise Entity, or #2 Enterprise Entity, is responsible for all kinds of enterprise development issues, such as the engineering design, construction and consulting services involved, etc., as required by the Manufacturing Enterprise Entity #3.

3. The Manufacturing Enterprise Entity, or #3 Enterprise Entity, is responsible for the production of physical or informational products, including both semi-products ready for further processing or assembly and final products serving the ultimate users.

4. The Enterprise Product Entity, or #4 Enterprise Entity, represents the life history of any semi- or final products delivered by a Manufacturing Enterprise Entity. The semi-products may be provided to other #3 Enterprise Entities for further processing, or the final products may be supplied to the ultimate users or customers of the Entity.

Bernus and Nemes [101] were the first to point out this recursive nature of a lifecycle model such as PERA. This
occurred in their definition of a proposed Generic Enterprise Reference Architecture and Methodology (GERAM) for the IFAC/IFIP Task Force on Architecture for Enterprise Integration. PERA has a major place in these authors' assignment of the characteristics and capabilities of existing architectures to GERAM [101].

As noted directly above [83], these authors were involved with work which showed the same properties of the lifecycle model but did not note the extent of its recursive nature as did Bernus and Nemes in the proposal [101]. Bernus and Nemes proposed that there were only four classes of enterprises noted above and that any chain containing them would be only four entities long. However, as noted below, our work shows that we agree with the four entities but the chain can be of unlimited length [102].

7.2.2 A Genealogical Tree of PERA Enterprise Development

In presenting their proposal, Bernus and Nemes based their discussion on a lifecycle diagram developed earlier by the Task Force [103] and called by them the "Matrix Lifecycle Model" since it combined aspects of CIMOSA, GRAI-GIM, and PERA to show the multi-functional contributions of each of them to the definition of GERAM. They did not actually illustrate the interconnectivity of the functions of the several enterprise entities as was done here earlier in Figure 5.21. In this section we will illustrate the Bernus and Nemes concepts using the PERA lifecycle diagram. It would be much more difficult to develop a similar diagram using the Matrix form (see next chapter for details).
Based on the previous discussion on the types of Enterprise Entities, Figure 7.5 presents a possible structure of
FIGURE 7.4 RELATIONSHIPS BETWEEN THE STRATEGIC MANAGEMENT ENTERPRISE ENTITY AND THE ENGINEERING OR MANUFACTURING ENTERPRISE ENTITIES
FIGURE 7.5  A GENEALOGICAL TREE OF ENTERPRISE ENTITIES
an enterprise genealogical tree describing the development of a series of Enterprise Entities and the relationships between their life cycles. Although different Enterprise Entities play different parts in establishing an enterprise lifecycle, they are all connected together by the lifecycle to form an enterprise chain.

The existence of the chain shows that no enterprise can spring full-blown from a vacuum. There must be somewhere a perceived need for it to occur. Every enterprise must be able to locate its position somewhere on the tree which will go on ad infinitum in both directions from #1 and #4 Enterprise Entities.

Each Enterprise Entity can be considered as a node on the tree where each arrow there represents interfaces between the Entities. Its direct connections with other Enterprise Entities is through communication through the program interfaces which are established in the enterprise life cycles involved. For example, as shown in Figure 7.5, the Strategic Management Enterprise Entity will be in charge of developing both information and manufacturing policies ready for further Requirement Analysis carried out by the Engineering Design Enterprise Entity. These two Entities will communicate through the program interface between the Concept Phase and Definition Phase of the Manufacturing Enterprise Entity in developing the lifecycle of the subject Entity.

Because of the important part played by the Engineering Design Entity in developing the engineering design for the Manufacturing Entity, the Strategic Management Entity may take over the detailed tasks of preparing not only the sets of policies but also the functional networks required for
the Engineering Design Entity. These detailed requirements may be publicized as bid documents for potential Engineering Design Entities which are going to compete for the bid. In order to satisfy all the requirements involved and develop its own engineering capabilities, the Engineering Design Entity who takes the bid may itself charge an Engineering Consulting Entity to look after its own Design and Construction Phases, as is also shown on Figure 7.5. Therefore, communications between the Strategic Management Entity and the Engineering Consulting Entity will take place through the program interface between the Definition Phase and the Specification Phase of the lifecycle of the Engineering Design Entity.

Another important connection between the Enterprise Entities can be created during their life cycles when they participate in the development programs of others. An example of this is potential customers helping establish the functional and operational requirements for the products of a manufacturing enterprise which they will later use themselves. This connection is illustrated by the sketch of Figure 7.6 [102] which shows how users' requirements or reports on the product, a vehicle for instance, may have input to the product development processes carried out by the manufacturer of the vehicle. First, the way that PERA presents the lifecycle of a physical product needs to be explained as follows.

In order to describe the vehicle's lifecycle under the framework of PERA, as shown in Figure 7.6, a proposal for the new product will be made as the very initial idea for this product. This will comprise the beginning of the PERA framework for that product as the entity identified. Then,
FIGURE 7.6 USERS' INPUT TO A PRODUCT DEVELOPMENT PROCESS CARRIED OUT BY THE MANUFACTURER OF THE PRODUCT
the goals, policies, requirements, etc., for the development of the proposed vehicle project can be further examined. Here, a recall of PERA's concepts of Control and Physical Manufacturing is necessary to introduce PERA's multi-stream development into the product lifecycle model.

The concept of control in PERA refers to the controlling operations through information concerning the current and future status of the subject system (see Section 3.1.4 in Chapter 3). That is, this control is a combination of decision making and all information necessary [99]. By following this concept, Figure 5.4 has shown that in the case of a manufacturing plant, CRT, sensors and other data gathering devices are considered as part of the information system, while all valves, actuators and sensor receptacles are considered as part of the manufacturing equipment. In the case of the vehicle, by following the same lead, all operations required to collect information and make decisions during the vehicle driving, such as those of speed control, direction control, navigation, etc., will be considered as Left Hand Side Activities (or Tasks) of the vehicle architecture, while all other operations required to carry passengers and cargoes, such as those of loading, power application, etc., will be considered as Right Hand Side Activities (or Tasks) as shown in Figure 7.6.

When the two streams of the Functional Analysis are split into three Implementation Architectures, the designer of the vehicle will decide among his/her options to what degree the tasks needed by the operations of the future vehicle will be shared between human and the on-board devices involved. For example, on the Left Hand Side of the Product Architecture, the designer may consider whether cruise con-
control, traction control, electronic instrument-panel display with a trip computer monitor, etc., are introduced into the vehicle to ease the control tasks of the human driver. On the Right Hand Side, the designer may decide what kinds of "power devices" such as power steering, power windows, power seat, an on-board crane on the vehicle, etc. are to be adopted to take over some of the manual labor of operation. Depending on all these decisions, instruction manuals and special training programs if needed will be prepared for the future drivers and other laborers who may be involved in the maintenance of the vehicle.

Therefore, the requirements, even involvement, from potential users of the vehicle such as the freight hauling company shown in Figure 7.6 have to be considered as early as possible in the product development processes so that proper arrangements could be made in the design and manufacturing process to take care of the users' interests. If the designer and manufacturer of the vehicle fail to respond to the needs of their customers, there will be a much greater chance for them to be forced out of the competition in today's manufacturing world.

7.2.3 The Purdue Guide as an Information Product and Tool Serving for Enterprise Development

The Purdue Guide to Master Planning and Implementation for Enterprise Integration Programs [36] can provide another example of the interaction of Enterprise Entities. If the way that the Guide has been created and developed is considered, the Guide itself is seen to be an information product of some Engineering Enterprise Entity since it is a refined recording of the type of activities carried out by a #2 En-
tity. As a small application case of the enterprise genealogical tree, the role of the Purdue Guide on the tree can be discussed as follows (see Figures 7.7 and 7.8):

- The Purdue Guide can be considered as both a product and a tool. As a product, it is developed by some #2 Information Services Enterprise Entity to serve as a tool used by #2 Enterprise Entities involved in enterprise development programs (see Figure 7.7). During the actual act of analysis, design, etc., for the development of the Purdue Guide itself, all activities involved in producing the management and control contents of the Guide will be considered at the Right Hand Side of the PERA Diagram (see Figure 7.7). It is an information product of the first #2 Information Service Enterprise Entity.

- As a tool, it can serve as a guide to help engineer an enterprise development program by #2 Enterprise Entities joint with #1 Enterprise Entities (see Figures 7.5 and 7.8). In this case, all operational actions involved in developing the management and control requirements of that program or project according to the Guide would be the Left Hand Side Activities for the new #3 Enterprise Entity (see Figure 7.7).

- Considering only the simple production process (printing, binding, etc.) of the Purdue Guide, it would be only a simple documentation reproduction, i.e., it will be a Right Hand Side Activity on the PERA Diagram in the Customer Product and Service
Equipment Architecture of the operating #2 Enterprise Entity (see Figure 7.7).
FIGURE 7.7 THE PURDUE GUIDE AS A PRODUCT OF #2 ENTERPRISE ENTITY
DEVELOPMENT OF PROJECT MGMT SYSTEM FOR #3 ENTITY

DEVELOPMENT OF PROJECT MGMT SYSTEM FOR #2 ENTITY

DEVELOPMENT OF #3 ENTITY

DEVELOPMENT OF #2 ENTITY

DEVELOPMENT OF #1 ENTITY

DEVELOPING MISSION, VISION AND VALUES OF #3 ENTITY AND #2 ENTITY

DEVELOPMENT DUTIES OF #1 AND #2 ENTERPRISE ENTITIES
7.3 Discussions of the Genealogical Tree of PERA

Reference [103] points out that enterprise integration is a strategy as well as a technology. Likewise, PERA’s achievement as an Enterprise Reference Architecture can be shown to have the two aspects, i.e., both strategic and technological which are reflected in the enterprise genealogy of PERA as shown above. The findings of the formalism study conducted earlier in this report may also be reviewed here based on the needs for formalism as shown by the enterprise genealogy.

7.3.1 PERA’s Contribution to System Thinking and Systems Engineering

The study of enterprise genealogy can prove to be very helpful in determining the proper strategic view of system development and an important tool to aid this development. In [19], the Indian parable of the Four Blind Fakirs and the Elephant is used as an analogy (see Figure 7.9 [19]) to explain the importance of recognizing a complete picture of the CIM problem as a whole. If an enterprise is now explained by the analogy of the elephant, PERA via the enterprise genealogy is able to show the whole family of these “elephants” and the developing mechanism for the whole “elephant” race. As can be readily seen, this “elephant” happens to be any typical engineering system as discussed in the field of systems engineering.

In [95], systems are considered to be listed as follows in systems engineering:
• Large systems, e.g., in a large number of different parts, in a multiple repetition of identical parts, or of the numbers of functions to be performed, etc.;

• Complex systems, e.g., a mathematical model of the system will be complicated, or many interfaces exist with high traffic across them, etc.;

• Non-homogeneous systems; in particular, those which contain many of the four distinguishing components, i.e., natural, man-made, informational and organizational. Even the man-made components can themselves be non-homogeneous in the sense that they are realized by very different technologies like mechanics, electronics, software, computer hardware, etc.

As the approach taken by PERA to study an enterprise development program covers every aspect of the system as examined in systems engineering, PERA has thus provided a universally applicable lifecycle model of system development for any systems engineering study.

When system problems at the strategic level are attacked, a systematic point of view is a must for everyone involved in order for them to understand the system and to recognize the problems encountered.

Many difficult questions related to system development can be asked concerning any enterprise development program. For example, what is the true nature of the problem to be attacked? What is the right approach to take to start the investigation? How does one know that the right answer has been obtained as the outcome from that study? And so on.
If the systematic investigation leads to a wrong conclusion, any following actions will be hopeless from the beginning.
FIGURE 7.9 WHAT IS CIM?
Technically speaking, in system thinking, you have to keep a certain level of abstraction. Otherwise, too much detail will result in an endless mire of data. Likewise, too little detail can only lead to an inadequate result. Neither answer will be correct. Current theories are apparently of little help. A theoretical basis for the major part of systems engineering is largely missing [95]. Although Prof. Suh's axiomatic approach helps identify the nature of a good engineering design, a more generic theory must apply to the overall engineering domain if it is to render any particular instructions to system problems (see Figure 7.10). B. Thomé comments on such a situation as follows in the conclusion of his recent book on systems engineering [95]:

"The practice of engineering Computer Based Systems has been widespread for quite some time. In spite of this, Computer Based Systems Engineering is still a relatively undeveloped discipline. Currently, we do not have a practical systematic body of knowledge available to solve the problems encountered in the engineering of Computer Based Systems. At the present time the fact that many of those problems follow patterns is being more widely recognized. Thus systems engineering in the field of Computer Based Systems has begun to evolve as a discipline in its own right."

By studying the path of PERA's development from a reference model of Computer Integrated Manufacturing to the Enterprise Reference Architecture, the authors believe that PERA presents a universal model of system development in systems engineering (see Figure 7.10). Although a high
Figure 7.10 Applicability of Prof. Suh's Axioms and Pera's Lifecycle
level system study could be difficult to carry out, PERA has developed a unique strategy for attacking a system life history as completely as possible and for seizing the essence of the growth pattern of system development. The development of PERA has illustrated that any system study should recognize the current system and also its complete life history. Not only should this be the life history as a whole but also of any of the independent streams appearing throughout this history. Not only the individual system's life history but also any intersections with the life histories of other different systems must be considered.

The framework provided by PERA is also a helpful tool to explore different levels of abstraction in a system study. It is powerful enough to capture all aspects of the system lifecycle; simple enough to illustrate all kinds of ideas concerning system development with understandability; flexible enough to study system problems at all levels from interfaces, to subsystems of a system, and all the way through to whole families of systems. Equipped with the "birds-eyes-view" provided by this framework, you do not have to worry about missing a particular point or being overwhelmed by a mass of systems activities.

7.3.2 The Architectural Formalism of PERA

As seen in Figure 7.5, the major activities carried out by #1 and #2 Enterprise Entities are developing products for or providing services for a #3 Entity’s development program. These products or services may either be information or, as at the Construction Phase of the #3 Entity Development Program, be a physical product (i.e. the #3 Entity itself) cre-
ated according to instructions from one of the information products (i.e. the detailed design of the #3 Entity). The genealogical tree as a conceptual model stemming from PERA presents a picture of fostering the history of #3 Entity's development all the way through to its final operation. This is the central issue which dominates all undertakings of the Enterprise Entities involved, including interactions between them.

It is now well understood why this information developing process can involve so much magnitude and complexity. Actually, the problem of handling complexity is the fundamental reason of developing Enterprise Reference Architectures as the model of enterprise development [19, 21, 103]. However, in order to achieve the simplification of the overall complexity needed by this model, "we must find the right representation" [104, p. 228]. This is the type of formalism which has been pursued in this report. Here, the degree of the complexity to be presented will still be important.

In his book on the science of management decision [105], H. A. Simon has categorized two polar types of decision making according to their degree of difficulty or complexity. He calls these "programmed decisions" and "non-programmed decisions" (see Figure 7.11 [105]). He further explains these two types as follows:

"Having christened them, I hasten to add that they are not really distinct types, but a whole continuum, with highly programmed decisions at one end of that continuum and highly nonprogrammed decisions at the other end. We can find decisions of all shades of gray along the continuum, and I use the
terms programmed and nonprogrammed simply as labels for the black and the white of the range.

"Decisions are programmed to the extent that they are repetitive and routine, to the extent that a definite procedure has been worked out for handling them so that they don't have to be treated de novo each time they occur....

"Decisions are nonprogrammed to the extent that they are novel, unstructured and unusually consequential. There is no cut-and-dried method for handling the problem because it hasn't arisen before, or because its precise nature and structure are elusive or complex, or because it is so important that it deserves a customer-tailored treatment...." [105, pp. 45-46]

Later in the same book, Simon also asserts that computers can in no way change "the special place that man has in God's scheme or nature's." He predicts that the decision making process in future organizations will not change the way in which workers and executives view their own organizations from what they see today. His prediction states:

"1. Organizations will still be constructed in three layers: first, an underlying system of physical production and distribution processes, second, a layer of programmed (and probably largely automated) decision processes for governing the routine day-to-day operation of the physical system, and third, a layer of non-programmed decision processes (carried out in a man-machine system) for monitoring the first-level processes, redesigning them, and changing parameter values,
2. Organizations will still be hierarchic in form. The organization will be divided into major
### Types of Decisions

<table>
<thead>
<tr>
<th>Programmed:</th>
<th>1. Habit</th>
<th>1. Operations Research:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine, repetitive decisions</td>
<td>2. Clerical routine: Standard operating procedures</td>
<td>Mathematical analysis Models</td>
</tr>
<tr>
<td>Organization develops specific processes for handling them</td>
<td>3. Organization structures: Common expectations A system of subgoals Well-defined informational channels</td>
<td>Computer simulation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nonprogrammed:</th>
<th>1. Judgment, intuition, and creativity</th>
<th>Heuristic problem-solving techniques applied to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-shot, ill-structured novel, policy decisions</td>
<td>2. Rules of thumb</td>
<td>(a) training human decision makers</td>
</tr>
<tr>
<td>Handled by general problem-solving processes</td>
<td>3. Selection and training of executives</td>
<td>(b) constructing heuristic computer programs</td>
</tr>
</tbody>
</table>

### Decision-Making Techniques

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Modern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Habit</td>
<td>1. Operations Research:</td>
</tr>
<tr>
<td>2. Clerical routine: Standard operating procedures</td>
<td>Mathematical analysis Models</td>
</tr>
<tr>
<td>3. Organization structures: Common expectations A system of subgoals Well-defined informational channels</td>
<td>Computer simulation</td>
</tr>
</tbody>
</table>

#### FIGURE 7.11 TRADITIONAL AND MODERN TECHNIQUES OF DECISION MAKING
subparts, each of these into parts and so on, in familiar forms of departmentalization. The exact bases for drawing departmental lines may change somewhat. Product divisions may become even more important than they are today, while the sharp lines of demarcation among purchasing, manufacturing, engineering, and sales are likely to fade.” [105, p. 134]

If we look into the tasks carried out by the #1 Enterprise Entity in Figure 7.5, all of them will be classified into the nonprogrammed decisions noted in Figure 7.11. This is because the information development required to proceed from management’s mission, vision and values to the sets of policies needed for all related enterprise programs are all typical policy decisions which can only be taken by humans (see Figure 5.1 and 5.2 also). Similarly, in the field of engineering endeavors carried out by the #2 Enterprise Entities, human-centered and creative activities continue to be the characteristic needs there [95]. Even in an ideal situation where all technical design jobs would be fully automated, there would still be the necessity to directly follow any management special mandates, etc., in engineering design (see Figure 7.12 [19]). Human judgment and decision making would still be necessary to merge these special management requirements into the design. In other words, all the decisive tasks with which we are most concerned in the development are typically nonprogramable in their nature.

Then, the architectural formalism used for PERA must consider its major audience: the human being involved in the enterprise development program, i.e., the management, the engineers, etc. PERA and its methodology must present the crucial knowledge of enterprise development to these
FIGURE 7.12 HOW MANAGEMENT MANDATES, REQUIREMENTS, ETC., INTERFACE FUNCTIONAL AND DETAILED DESIGN PRACTICES AND RESULTS
implementors by using the proper mixture of prose and graphics. This is not to say that a more formalized presentation is impossible during the execution of an enterprise development program. The most rigidly defined formality will be required when the program comes to its programmable tasks. The program implementor at that point will most probably not be a human but a computer (see Figure 5.19) if the necessary automated tools are available.

It is impossible for any presentation to discuss each and every detail of a real world problem. The principle of considering only the "minimally complex", which is introduced by M. Flavin [106] for information modeling, is followed in this study to help solve the problem. In the context of this report, this "all-and-only" [106] principle can be understood as a condition that guarantees that the presentation contains all the relevant subject topics needed to bound the domain of the Enterprise Reference Architecture, but only the subject topics needed to describe the domain.

An immediate follow-up to the principle of "minimally complex" is the concern of completeness. P. Bernus defines the completeness of enterprise models as:

"An enterprise model is complete relative to the processes using the model if the processes can create (and behave according to) the intended interpretation of the model." [107]

He further points out that the above definition considers completeness "as a pragmatic property of enterprise models". In contrast, "an absolute measure of completeness is the extent to which enterprise models are theories (in the sense of model theory) ..." [107]. The fact is that even if an enterprise model was complete in the absolute sense, its
completeness in the relativistic sense in reality would still depend on the way in which it was interpreted by the user who had to create the process described by the model.

For example, if we take a look at all engineering processes related to enterprise development, it can be seen that engineering activities actually need to draw on both science and experience as two prime sources in practice [95]. Even in one of the most mature engineering professions, mechanical engineering, an engineering drawing, the most commonly used presentation "makes full sense only to someone who is familiar with local practices. It leaves a great deal to the reader to interpret and to supply default values." [108]

When commenting on a system based methodology for problem solving in the real world, P. B. Checkland has given probably the most pessimistic estimation from a theorist:

"The methodology has been found to be helpful in a wide range of problems of many different types; but given the nature of it even that is an assertion which cannot be proved." [109, p. 313]

However, the real world may not be that desperate. In the engineering world, people can still use their observations to determine what they believe to be universal in their applications. Prof. Suh points out that Euclid, Newton and Einstein each actually developed their own studied system based on sets of fundamental postulates or axioms which they found to be in agreement with their observations of nature [84]:

"These axioms could not be proven, but were assumed to be valid as long as there were no exceptions or counterexamples." [84, p. 15]
In the case of this study, things are even better. The Axioms of Design developed by Prof. Suh have helped the authors launch an axiomatic approach to show that the Purdue Methodology and PERA are good engineering designs. On the other hand, however, the theoretical studies also remind us that the success possible from an application of a systems engineering technique will largely depend upon the understanding, judgement and decision making made possible by that particular technique. Methodologies also will surely help but never more than to the extent that an engineering drawing helps its users.

As an investigation on the formalism of PERA, the authors believe, this study has covered all important issues related to executing an enterprise development program as follows (see further discussion of the formalism method in Section 8.2):

- Presentation of PERA and of the Purdue Methodology from simple application up to the genealogical tree of enterprise development with the theoretical support from Suh's Axioms for proving its design.
- Interface identification, including intraarchitectural, interarchitectural, and program interfaces with all the related formalisms needed in order to promote interfaces standardization whose importance is commented in [110] as:
  
  "Integration provides the requirements for many interfaces while standards provide a powerful form of specification for interfaces."

- Integration of all available tools under the Enterprise Reference Architecture necessary for executing
the tasks of an enterprise development program. Formalism of these tools will be to different de-
grees dependent on the degree of automation of the subject tool. The formalism of a particular tool
would be better if it could be integrated with whatever interface(s) are connected to it.

Based upon the above important issues, the authors believe, it should not be difficult for an experienced system planner to understand the messages delivered by PERA. Actually, the authors were amazed to observe how quickly the members of the Industry-Purdue Architecture Users Group perceived the essence of Suh’s Axioms and the concept of program interfaces promoted by this study and further developed these concepts when they applied PERA and the Purdue Methodology to improve their engineering system including the information technologies adopted [111, 112, 113].

7.4 Program Iteration Shown by the PERA Lifecycle Model

In today’s manufacturing world, every one is constantly looking for better, faster and cheaper manufactured products. This trend has imposed an inevitable fact that the only constant factor is change [110]. In order to cope with the changing world, the following fundamental approaches will help enterprise development programs progress faster and better:

- Develop industry-specific, step-by-step guides based on PERA and the Purdue Methodology for different Enterprise Entities in different industries along the enterprise genealogical tree (Figure 7.5). These guides will serve as detailed handbooks for manage-
ment and engineering professionals to develop their own particular programs or projects.

- Develop information systems to support the information dissemination through the lifecycle of the subject enterprise from its Concept Phase, Definition Phase, Specification and Detailed Design Phases through its Construction and Operation Phase. These information systems must meet different requirements at different Phases and also facilitate communications through all Program Interfaces.

- Develop automatic tools for programmable tasks through all Phases of an enterprise development program. Both the information system support mentioned above and the automatic tools must be integrated with the framework of the program development to send the right information to the right place at the right time to perform the right operation as automatically as possible.

- Promote standardization in developing all the subjects noted above.

In addition, project management of enterprise development programs may adopt a more concurrent strategy, other than a purely sequential process mode, an approach which is a left-over topic from the first section of this chapter. This topic was put forward many times during meetings of both the IFAC/IFIP Task Force on Architectures and the Industry-Purdue Architecture Users Group. Discussions between members in those meetings have been of great benefit for this report and have prompted the authors to introduce an iterative mode into the lifecycle model of PERA as follows.
7.4.1 Implications of Program Interfaces

Although the phased approach of PERA has indicated a sequential execution of enterprise programs, it is still possible to combine an iterative mode explicitly within its lifecycle. In order to fit this mode herein, the starting and ending points of the program iteration have to be arranged properly. At the same time, based on experiences in engineering design provided by the members of the Industry-Purdue Architecture Users Group [111, 112, 113] and an understanding of the spiral model of software engineering processes (Figure 7.2), prototyping as part of the iterative mode should also be introduced.

By examining the evolving process of an enterprise program as described by PERA (Figure 3.18), it can be seen that the program interfaces represent milestones along its lifecycle. Each of them can be the right place where people will see the currently important complete picture of the future enterprise but with different degrees of details. Even the two sets of policies completed at the end of the Concept Phase ought to be enough to fully demonstrate the Management Mission, Vision and Values and to lay "a foundation for building an enterprise integration plan that ensures it's viability" [36, p. II-2-34]. The Functional Networks at the end of the Definition Phase, the Functional Design at the end of the Specification Phase and the Detailed Physical Design at the end of the Detailed Design Phase (Figure 3.18) are all expected to be able to present global pictures of the future enterprise with an increasing degree of detail.
along with the progression of the development program. Therefore, each of the program interfaces will be

- The right place to check coordination between the multiple information streams such as connections between the two Functional Networks or between the three Implementation Architectures.
- The right place to have a complete systematic prototyping finished and then to restart it if necessary.
- The right place to check the program with the upper management if necessary.

The more complicated and more unknown the subject enterprise is, the more important the managerial functions of the program interfaces are. Management and/or customers may desire to have more participation during program development. The program interfaces show them a way of having a "minimum" interaction with that development.

7.4.2 Graphic Model of Program Iteration by PERA

Figure 7.13 presents a possible program iterative mode as shown in PERA by taking three phases as an example, where a solid line on the relevant small diagrams represents the portion of the program already carried out to the point where the need of rework is evident in the information already obtained. The dotted lines in Figure 7.13 represent the remaining work but also the regions for possible estimation of the remaining program of the subject enterprise as used to show the degree of correctness of that part already completed. The more solid the line, the more complete will be the tasks represented. The initial trials such as
prototyping are only considered as a certain abstraction or model of the future program. However, failure to find support for current findings from the initial trials signals the need for a further iteration and revision of the earlier work.
FIGURE 7.13 REPRESENTATION OF POSSIBLE PROGRAM ITERATION AS SHOWN IN PERA
Each mini "windchime" in Figure 7.13 is confined within one set of program interfaces. This indicates that the rework to be carried out does not have to start from scratch since quite a large amount of information has already been obtained from the previous work. If, however, the program really had to be restarted from the very beginning of the big "windchime", a higher cost and time input would be inevitable. This, of course, would be undesirable.

The mini "windchimes" also indicate that it would be better if the rework could still follow the sequence of the original lifecycle. One would think at first look that this sequence might not be necessary. However, according to the observation of a member of the Industry-Purdue Architecture Users Group [111, 112, 113], the program iteration may lead to a unstable execution if the sequence is not properly maintained.

7.4.3 Program Management and Program Stability

According to one of the corollaries of Suh's Axioms, Decoupling of Coupled Design [84], and his long-term experience in engineering design, Gary Rathwell [112] of Fluor Daniel points out that although dependencies exist between the tasks of program execution, the phased approach described by the PERA lifecycle can be considered as "solving a set of simultaneous equations by putting them into a 'triangular form'" as shown in the following equation [112]:
Where

\[ O_i \quad \text{output } i \text{ of an project;} \]

\[ I_j \quad \text{input } j \text{ of an project; and} \]

\[ A_{ij} \quad \text{project activity which needs input } I_j \text{ to produce output } O_i. \]

As noted before, all efforts in the earlier phases of a program, as emphasized by PERA, will carry ever greater weight throughout the later development. Their increased influence can be explained by the lower part of the triangle in Equation 7.1, except the diagonal elements \( A_{ii} \) there. If the sequence of “Phased Engineering” [112] is followed, it will be like solving Equation 7.1 row by row starting from its first row. Then just exactly one input each time needs to be considered in calculation. Each phase of the project can be treated like a single subproject independent from others. Failure to “phase” the engineering project properly will result in unnecessary interdependency, including phase interdependency, between outputs and much more control efforts on inputs will also be required. In the worst case, if all possible interdependencies between input and output are invoked, the simplicity shown in Equation 7.1 will deteriorate into that complexity represented by the following equation [112]:

\[
\begin{bmatrix}
O_1 \\
O_2 \\
O_3 \\
O_4 \\
O_5
\end{bmatrix} =
\begin{bmatrix}
A_{11} & 0 & 0 & 0 & 0 \\
A_{21} & A_{22} & 0 & 0 & 0 \\
A_{31} & A_{32} & A_{33} & 0 & 0 \\
A_{41} & A_{42} & A_{43} & A_{44} & 0 \\
A_{51} & A_{52} & A_{53} & A_{54} & A_{55}
\end{bmatrix}
\begin{bmatrix}
I_1 \\
I_2 \\
I_3 \\
I_4 \\
I_5
\end{bmatrix}
\]  

[7.1]
Rathwell also noted that the sequential mode as an effective approach will also achieve independence between multidiscipline groups working for engineering design within the same phase [112].

Figure 7.14 [111] presents a possible impact of "rework loops" [113] on the stability of a design process through the inherent sequential steps started by a local change. This is because each step on the rework loop takes "both time delay and gain" to make the relevant changes. Thus, if the changing pace along the loop does not follow the rule demonstrated in Equation 7.1, these rework loops may "not only waste effort and time, but can also become unstable" [113] by throwing in more and more costly changes through later design phases.

Therefore, according to Rathwell, both version control within each phase and control of change orders between phases must be "fine tuned for optimum response and stability" [111] for the rework process. Otherwise, a faster program iteration will not necessarily be a better choice, especially for a multidisciplined program. An information technology support with tracability embedded in the engineering design system has proved to be such a powerful tuning aid.
FIGURE 7.14  AN UNSTABLE REWORK LOOP WILL INVOKE MORE AND MORE CHANGES
8. COMPARISONS BETWEEN TYPE TWO ENTERPRISE REFERENCE ARCHITECTURES

Comparisons between major architectures will be presented in this chapter. With respect to the work done on this subject by the IFAC/IFIP Task Force [1, 103], the major comparison methods, especially the graphical mapping methods, developed by the Task Force will be introduced first. Then, based on that result, further comparisons between the lifecycle models of the Type Two Architectures, including the matrix model recently proposed by P. Bernus and L. Nemes [101], will be further carried out.

8.1 Comparison Done by the IFAC/IFIP Task Force on Architectures for Integrating Manufacturing Activities and Enterprises

It was set forth at the 11th Triennial Congress of the International Federation of Automatic Control (IFAC) in 1990 that a new Working Group in IFAC would be formed to work on the subject of Control Architectures for Integrating Manufacturing Activities and Enterprises. Later, this Working Group evolved into a joint Task Force between the IFAC Manufacturing Technology and Computers Committees and the IFIP Technical Committee for Computer Applications in Technology (TC-5) with membership containing representatives from the
groups developing each major candidate architectures all over the world [103].

It was suggested that this Task Force interface to ISO TC 184/SC5/WG1, which has been chartered to

"Study existing work and to create a multi-dimensional, open-ended reference model which will provide a basis for long-range planning for standardization through the identification of interfaces and their characteristics (e.g., electrical, mechanical, man-machine, information, procedural, language, etc.) between system automation elements."

[114]

and interface to the IEEE Control Society and IFAC TC on Theory project "Facing the Challenge of Computer Science in the Industrial Area of Control" [103].

The Task Force first decided that only three from among the many available candidate architectures would fall into the category of the Type Two Reference Architectures (see Section 2.4 and reference [1]). Methods were then designed by the Task Force to evaluate "how well the three candidate architectures (CIMOSA, GRAI-GIM, and Purdue) fulfilled the needs for 'integrating manufacturing activities and enterprises'". Major conclusions drawn from the work of the Task Force on each Type Two Architecture have been presented in Section 1.1, Section 4.1 and Section 4.3 of this report. In order to further illustrate some perspectives of different architectures, the comparison methods used by the Task Force are presented in this section.

8.1.1 Evaluation Methods developed by the Task Force
The following three methods were developed by the Task Force [1, 103]:

1. A comprehensive questionnaire and its answers which were developed and completed to survey major concerns and to collect extensive data on the subject of the Type Two Reference Architectures;

2. A graphical mapping method based on a co-representation of the lifecycle diagrams of each Type Two Reference Architecture to compare in each pair between their characteristics and capabilities;

3. Another graphical mapping method based on a matrix which would represent general research issues from each Type Two Architecture, that is, compare them by mapping them all to one common matrix-like functional representation of reference architectures.

The questionnaire collected a great amount of information for evaluating the three Type Two Architectures. In order to develop the questionnaire, three versions plus a shorter questionnaire were generated and reviewed during a two-year period before the final version. It was even more difficult, however, to develop graphical representations for architecture mappings, because a proper representation for this purpose has to be simple enough for its understandability and also be expressive enough to convey all useful information for architecture development. Besides, the mapping presentations have to be neutral between the candidates so that nobody could gain extra advantages by any biased means.

Ing Dick Zoetekouw of AT&T Netherlands and Dr. Peter Bernus of Griffith University in Australia, both members of the Task Force, made the major contributions to the develop-
ment of these three evaluation methods. They have been accepted by all members of the Task Force. Because this chapter is using the two graphical mapping methods for reference in further discussion, a brief description of the two mapping methods is given below.

8.1.2 The Co-Representation Mapping Method [1, 103]

The purpose of this mapping is to compare the coverage of each candidate architecture with each of the others by using their lifecycle diagrams. The major problem here was CIMOSA did not provide such a specific lifecycle diagram like that of PERA (see Figure 3.1) or that of GRAI-GIM (see Figure 4.15).

D. Zoetekouw of AT&T Network Systems, Nederland BV, who was representing CIMOSA, developed this desired mapping which was adopted by the Task Force. His mapping starts by taking a right-side view from one of the vertical layers of the CIMOSA Cube (Figure 4.1) as shown in Figure 8.1. In order to present a complete lifecycle model, three more layers are added to the side view as in Figure 8.2. They are named as (1) the Identification and Concept Block, (2) Build, and (3) Operation. These additional layers were adapted from PERA. The first four layers (Figure 8.2) together become a larger block named as Analysis and Development. Figure 8.2 provides the needed diagram against which each architecture including CIMOSA itself is then mapped.

Figure 8.3 explains the mapping procedure by showing how the CIMOSA diagram (Figure 8.2) presents the CIMOSA Architecture itself. This self-mapping diagram concludes that the CIMOSA Architecture does not provide what Purdue calls
Identification and Concept (Figure 3.1), or what GRAI-GIM calls User Requirements and Initialization (Figure 4.15). In CIMOSA, this earlier development in an enterprise program
FIGURE 8.1 A RIGHT VIEW OF CIMOSA BUILDING BLOCKS
FIGURE 8.2 DEVELOPMENT OF THE CIMOSA BASIC SKELETON DIAGRAM
PROCESS and Tools for software Development are Available But Not Modelled. Processes and Tools for the Rest of the Enterprise are not Available. (Outside Scope of CIMOSA).

* Only Systems Change Within the Terms of Software Change Development is Within the Scope of CIMOSA.

FIGURE 8.3 CIMOSA ARCHITECTURE MAPPING AGAINST CIMOSA BASIC SKELETON DIAGRAM
is assumed to be previously available before the CIMOSA Architecture development work gets started. Similarly, the Build and the Operate Blocks are not fully considered in the CIMOSA scope.

8.1.3 The Matrix Form Mapping Method [1]

In order to develop an evaluation method for discussing all candidate Type Two Architectures on the basis of a set of general requirements, Dr. P. Bernus proposed a framework in a matrix form by combining the major differentiating aspects of each of these Architectures as different dimensions into the matrix. These dimensions include:

1. The Generic, Partial and Particular separation of CIMOSA in developing architectural models as three horizontal divisions.
2. The Function, Information, Decisional/Organization and Structural Views of GRAI-GIM as subdivisions of each horizontal division.
3. The Information and Customer Service and the Human and Machine separation categories from PERA as vertical divisions in each phase.
4. The Identification, Concept, Requirements, Design, Implementation, Build, and Operate Phases as a composite of the lifecycle presentation of each of the three architectures.
5. The Hardware versus Software categorization developed by SATT as an additional dimension.

*SATT is a methodology developed by P. Bernus, L. Nemes and others at the Computer and Automation Institute of the Hungarian Academy of Sciences the late 1970’s and early 1980’s. It was published in full only as proprietary material of the Institute and only in Hungarian. This technique was not evaluated by the Task Force in its studies. [1]*
It was noticed during development of the matrix form that not every combination of the above dimensions offers a substantive meaning. For example, in terms of program life-cycle, genericity can only apply to requirements and functional design. In other words, a detailed design must always consider the specific requirement of a particular enterprise if it is going to be meaningful. Therefore, in terms of generic and partial modeling, the Detailed Design of Purdue, and the corresponding Implementation Levels of GRAI-GIM and CIMOSA are considered meaningless. Further, the Construction and Operations Phase of Purdue and the corresponding Build and Operate Layers of GRAI-GIM and CIMOSA must be particularly specific in each case. Likewise, the division into hardware and software is not meaningful under the functional aspect of the Requirements Layer where the types of implementation are yet to be decided.

Thus, the evaluation matrix is modified as shown in Figure 8.4. Only those areas in a reference architecture which are considered potentially meaningful are present in the matrix form of Figure 8.4. Likewise for each dimension of the matrix. The Generic Build Block and Operation Block, for instance, are not presented because all modules and needed techniques and methods there are considered specific for each and every particular development program. On the other hand, at the Design layer, designs of the modules which may be applied to any Control/Information System may be supported by available Generic or Partial (e.g. industry-specific) design tools.

In order to verify the evaluation based on the proposed Matrix form, Dr. P. Bernus discussed his results with the developers of each candidate architecture, including Dr.
Francois Vernadat representing CIMOSA, and Mr. David Chen representing GRAI-GIM. Figure 8.5 presents the symbology used in the Matrix for evaluation. Figures 8.6, 8.7 and 8.8 present the resulting evaluation forms for CIMOSA, GRAI-GIM and Purdue by mapping each of them, including their Reference Architectures and Methodologies if available, onto the Matrix.

Since the shaded areas in Figures 8.6, 8.7 and 8.8 represent those parts where each architecture provides extensive guidelines, etc. (see Figure 8.5), it can be seen that Purdue is the most complete one among the three candidates according to this evaluation. On the other hand, the major areas studied by CIMOSA are the software used for control/information systems where it has provided the most formal results among the three candidates. Note that in Figure 8.6 the areas occupied by both solid triangles and solid circles represent the formally defined CIMOSA IIS. Similarly the solid triangles of Purdue's Matrix in Figure 8.8 represent the Purdue Reference Model. A thorough explanation of this matrix form evaluation can be found in the Task Force Report [1, pp. IV-4-1-IV-4-48].

8.2 A Discussion of the Task Force's Evaluation

As pointed out in [1], the generation of a generic evaluation form to characterize a candidate architecture is very crucial for the proposed evaluation. Since each of the major developers of the three candidate architectures was heavily involved or consulted during the Task Force evaluation, the results from the evaluation have laid down a le-
gitimate foundation for further studies on the three candidates.
### FIGURE 8.4 THE EVALUATION MATRIX

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Generic

<table>
<thead>
<tr>
<th>Function</th>
<th>Information</th>
<th>Decision/Org'n</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Partial

<table>
<thead>
<tr>
<th>Function</th>
<th>Information</th>
<th>Decision/Org'n</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Particular

<table>
<thead>
<tr>
<th>Function</th>
<th>Information</th>
<th>Decision/Org'n</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### REQUIREMENTS

<table>
<thead>
<tr>
<th>Customer Control Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
</tr>
</tbody>
</table>

### IMPLEMENTATION

<table>
<thead>
<tr>
<th>Customer Control Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
</tr>
</tbody>
</table>

### DESIGN

<table>
<thead>
<tr>
<th>Customer Control Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
</tr>
</tbody>
</table>

### BUILD

<table>
<thead>
<tr>
<th>Customer Control Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
</tr>
</tbody>
</table>

### OPERATIONS

<table>
<thead>
<tr>
<th>Customer Control Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Human</td>
</tr>
<tr>
<td>Machine</td>
</tr>
</tbody>
</table>
Separately defined description category, but methods not defined. An area surrounded by a solid line corresponds to a type of description which is separately defined in the architecture.

Separately defined description category with extensive methods and design principles available. If the area is shaded, then the architecture also provides -- in addition to defining the category of description -- an extensive set of guidelines which can be used to facilitate and speed up the generation of the necessary descriptions (models).

The architecture comprises a definitive generic model of this aspect. A solid triangle within a shaded area means that the architecture incorporates a generic model (description) which can be utilized as a blueprint by the user of the architecture, or could be the basis of standardization. An unfilled triangle means a model which is specified but not provided (yet) in detail in the architecture.

A hollow circle indicates formal modeling language(s) is not proposed but some would be adequate. A solid circle indicates a formalized modeling language prescribed to be used for all modeling activities. A circle within an area marked with solid line or with shading means that the architecture proposes or prescribes a modeling method which can be used as a means of representation for the given description category. The circle is filled if the modeling method is formal, unfilled if it is only partial formal.

A hollow rectangle indicates system component (module) identified, and a solid one indicates detailed module specifications available. A rectangle within a marked area represents a module description (thus is meaningful only in the resource aspect). An unfilled rectangle means that a software or hardware (or aggregate) module is present in the architecture definition as a separately identifiable component of the implementation. A solid filled rectangle means that a definitive detailed module description is part of the architecture (such as the Information Integrating Infrastructure of CIMOSA).

Category "not applicable".

**FIGURE 8.5 SYMBOLOGY USED FOR COMPLETING THE MATRIX FORM EVALUATION**
FIGURE 8.6  THE MATRIX EVALUATION FORM FOR CIMOSA
FIGURE 8.7  THE MATRIX EVALUATION FORM FOR GRAI–GIM
FIGURE 8.8 THE MATRIX EVALUATION FORM FOR PURDUE
A common ground in carrying out the evaluation by the Task Force is that all parties involved have realized the importance of a full lifecycle model as the basis for an Enterprise Reference Model. If we compare the GRAI-GIM lifecycle (Figure 4.15), the CIMOSA lifecycle (Figure 8.2), and the Evaluation Matrix (Figure 8.4) with the PERA lifecycle (Figure 3.1), it can be seen that the concept of a full lifecycle for enterprise program development as first proposed by PERA has been endorsed by this research group.

Another agreement reached by all parties in the Task Force Evaluation is that an enterprise should consist of three and only three Domains (GRAI-GIM) or Implementation Architectures (Purdue) at its Design Phase (Technical — GRAI-GIM; Detailed — Purdue) or its Implementation Phase (the Matrix) and Operations Phase/Layer. These can be described as Informational, Organizational and Manufacturing if we follow the terminology of GRAI-GIM (see Figure 4.15, 8.2, 3.1 and the dimensions on the horizontal direction of the Evaluation Matrix of Figure 8.4). There are still some differences between GRAI-GIM and Purdue in the wordings of the definitions of these three architectural parts and in the way of establishing a development program. However, it has been realized by the members of the Task Force that an enterprise in its final form as a product delivered to its end-users may undertake many different missions as assigned by these users. But, in every case, its underlying technical structures will fall into the above categorization under which the required design or development process should be worked out.

As also noted above, every Type Two Reference Architecture has shown its own type of multiple stream development
in its lifecycle model which finally leads to the three similar technical structures in implementation. Nevertheless, there is still considerable difference in developing the patterns of these multiple streams between Purdue and the other two, CIMOSA and GRAI-GIM as a group. In the Task Force Evaluation, this difference has been kept on the same matrix. However, in the next section, it will be pointed out that these two different patterns (PERA versus the others) are not directly substitutable. The four Views of CIMOSA and GRAI-GIM can become a useful supplement to the Purdue approach to enterprise modeling if desired.

As quoted in Section 1.1, the major criticism of PERA from the Task Force Evaluation is the needs for "formality in order to achieve eventual executability of the architectural constructs" [1, p. V-2-6]. Some suggestions for PERA from the evaluation included employment of the methodologies already used by CIMOSA and GRAI-GIM for their enterprise modeling, such as "formal modelling language (with a mathematically defined semantics)" or "a semi-formal one (with some formalism but no mathematically defined semantics)" [1, pp. IV-4-20-IV-4-21]. The emphasis in this Evaluation therefore was on the formality of those programmable tasks which might be carried out by computerized tools.

If we consider the different degrees of complexity involved in an enterprise development program as discussed in Section 7.3.2, the formalism of PERA and its Methodology can not be limited only to those forms or languages which are able to be recognized by computerized tools. In order to present the full-range of representations through all the levels of complexity involved, this study proposes a possible hierarchical structure for the formalism of PERA and its
Methodology as shown in Figure 8.9 with four levels: architectural, interface, tools and semantics. Establishment of a semantic formalism via the glossary provides a potential standardized definition of all important concepts involved in enterprise integration. This standardized semantics makes sure that all parties involved have a common understanding for the problems studied.

A formalism of the tools adopted in developing an enterprise program will facilitate the execution of the program by automating the tasks it carries out if the tool and its formalism can be integrated into the program structure. That is to say, the development of the tool and its formalism must meet the requirements raised by the program architecture itself.

The formalism of the interfaces will require that the individual tool developers recognize all kinds of communication needs if the information product of their automatic tool will be part of the information going through the communication interface. The tool will then be able to provide such a presentation form that both the computer which implements the tool and the human who needs the information from its product can understand its information content. In this case, this tool formalism must be very rigid because of involvement of the computer.

The design of this interface is in turn decided by the program architecture. The latter, in the case of Purdue, has its own architectural formalism supported by Suh's Axioms. The principles of the Axioms can then provide the theoretical foundation for the establishment of all interfaces.
FIGURE 8.9 A POSSIBLE HIERARCHICAL STRUCTURE OF THE ARCHITECTURAL FORMALISM PROPOSED IN THIS STUDY

**ARCHITECTURAL FORMALISM**
- Support the execution of the reference architecture and its methodology as a whole;
- Provide the guidance to the formalism of interfaces (both type 1 and type 2 interfaces) including their standardization;
- Human-oriented formalism for human program implementors;
- Standardization will enhance the reuseability of the architecture.

SATISFIED BY SUH’S AXIOMS

**INTERFACE FORMALISM**
- Support the transformation through the interface related;
- Provided the requirements for the formalism of the tools which connect to the interface;
- Either human- or machine-oriented formalism depending upon the technology available.
- Standardization of interface formalism will simplify the execution of the development program.

SATISFIED BY INTERFACE STANDARDIZATION

**TOOL FORMALISM**
- Support the automation of tasks during the execution of the architecture and methodology;
- Provide phase products with proper formalism for further processing or customer needs;
- Either machine- or human-oriented formalism depending upon the requirements involved.
- Standardization of tools is promoted to integrate them into the program architecture.

SATISFIED BY TOOL STANDARDIZATION (COORDINATION WITH INTERFACES)

**SEMANTIC FORMALISM**
- Support a common understanding for all important concepts involved in enterprise integration;
- Provide a standardized glossary to achieve this.

SATISFIED BY STANDARDIZED GLOSSARY
In terms of the promotion of standardization, Suh's theory is important because standardization can be considered to "hide" the complexity in information development process, and also according to one of Suh's corollaries, Use of Standardization (see Section 6.1), the standardization at the lower levels of the hierarchical structure (Figure 8.9) must obey the rules of design at the upper levels.

Since the completion of the Task Force Evaluation in March 1993, more and more people representing CIMOSA have realized the value of PERA. Vernadat [115] has made the following discussion of a presentation of PERA by Williams [116]:

"I would like to take this opportunity to stress again the complementarity between the Purdue Architecture and CIMOSA. Basically the Purdue Architecture provides the method and CIMOSA tries to provide the tools. We totally agree, at least we have reached a consensus in the sense that in terms of the activity description we tend to be totally in line with what you have said. Thus we can have different types of activities to fit your framework.

"The other thing is that we try to model only the information for control of the enterprise. That is, we try to have in the enterprise model all of the information needed for integration and control of the processes which need to be controlled. And I appreciate very much, of course, what you said about automatability and humanizability, because it opens the door to the aspects which we always forget in these kind of technical conferences, I mean manage-
ment science, and something called ergonomy, psychology, and so forth."

In addition to responding to the formality concern first put forward by CIMOSA, during the Task Force Evaluation, Purdue has also adopted the concepts of Generic, Partial and Particular Models originally developed by CIMOSA. This shows the degree of commonality with PERA in presenting an enterprise development program (see Figure 8.10 [1, pp. IV-5-11-IV-5-12]).

As shown here, it can be verified that there are generic regions in the earliest stages of enterprise development which are common to any and all industries, especially with regard to those involving management activities and the development of information (control) functions. Note that the Identification, Detailed Design, Manifestation and Operations Phases do not have Generic or Partial components. This is because these concepts can only be discussed in terms of specific (i.e. Particular) systems (see more interpretation of Figure 8.10 in [1, pp. IV-5-2-IV-5-13]).

8.3 Multiple Streams of Development in Type Two Reference Architectures

As noted in the last section, the biggest differences in the lifecycle models between Purdue and the other two Type Two Reference Architectures, CIMOSA and GRAI-GIM is in their different multiple streams of development. On the one hand, the developers of Enterprise Reference Architectures all realize that a model devoted to a single stream pattern of development as used for architectures devoted to an information system lifecycle alone is not capable of presenting the complexity needed in the larger domain of a complete enter-
prise system (see Section 7.1). On the other hand, as in
the case of an engineering drawing, we may have many differ-
et types of views from different angles for the same sub-
ject, but not all of them will provide an equally clear
overall picture of the subject. A few of them like main
views, i.e., those called front views, top views, right side
views, etc., will do the job, but others such as sectional
drawings will provide only needed local details. It is dif-
ficult to generate the whole picture from those sectional
views alone because without the main views a great deal of
additional information about the relationships between the
various sectional drawings would be needed before the whole
subject could be understood. Thus it can be seen that for
an application of a lifecycle model of enterprise develop-
ment, if the views used in the development pattern are not
actually the "main" ones, that much more effort will have to
be made to establish the whole picture. According to Suh's
Axioms, such a design is not a good one.

8.3.1 Independency in PERA's Multistream Development

In order to compare the different usage of the multiple
streams in the different reference architectures with each
other, let us first show the simplified model of PERA in
Figure 8.11.

As already discussed before in this report, the two-
stream development in Figure 8.11 starts with two sets of
Policies and then evolves into two Functional Networks. The
interdependency between the two streams is minimal because
each stream has its own independent functional requirements
and related data and analysis operations. The only needed
Essentially no generic definition since these is no total generic enterprise possible.

Some generic definition possible because of common practices and beliefs throughout all industry

Large amount of generic detail possible due to common industry practices.

Considerable amount of generic definition possible because of common factors involved.

Great deal of generic definition of system possible particularly for same industry systems.

Complete definition possible.

**FIGURE 8.10 HOW THE PURDUE METHODOLOGY HANDLES THE CONCEPTS OF GENERIC, PARTIAL AND PARTICULAR MODELS AS ESPoused BY CIMOSA**
FIGURE 8.11 MULTIPLE STREAM DEVELOPMENT OF PROGRAM PRODUCTS IN PERA
connectivity between the two streams is the "virtual" data exchange between the two network for the purpose of control. As long as this connectivity is preserved thereafter during the later decomposition of the two streams into the three Implementation Architectures, the connectivity between the three Implementation Architectures in their Design and Construction phases will be kept to a minimum for the same reason.

8.3.2 Dependency in CIMOSA and GRAI-GIM within their Four View Development

In contrast to PERA, CIMOSA and GRAI-GIM show major interdependencies between their Four Views as the lifecycle develops. Although a direct conversion from the Four Views to the three Technical Domains is missing in CIMOSA (see Figure 8.3), there is a direct mapping between the Four Views of CIMOSA and the Four Views of GRAI-GIM which was established in the Task Force Evaluation (see Figure 8.12 [1, pp. IV-3-7–IV-3-8]). Thus we can use the pattern of GRAI-GIM as an example to show this dependency.

According to the modeling sequence of the GIM method provided in [21, p. 33], Figure 8.13 shows a sketch of the working streams needed for converting the four Views developed in the GIM User Oriented Design into the three Technical Domains in the GIM Technical Oriented Design (see Figure 4.13 also). There each Technical Domain needs an input from each of three different Views. No simpler connection is possible because each View can only provide one aspect of the modeling subject in the Domains. In order to keep consistency between the four Views, GRAI-GIM has arranged for consistency checking at its Analysis Phase (Figure 4.15) and
for coherence checking at its User Oriented Design Phase (see also Figure 8.15) [16, p. 33].

CIMOSA authors have also noted that the CIMOSA Views are not independent from each other [13].

8.3.3 Phase Dependencies and the Four View Approach

In mapping the three Type Two Reference Architecture against each other in the Task Force Evaluation, CIMOSA uses its Requirements Level to represent Purdue's Definition Phase (see Figure 8.2) while GRAI-GIM use its Conceptual Model likewise (see Figure 4.15, 4.19 and reference [1, p. IV-3-50]). That is to say, all four Views including the Resource View (CIMOSA) or the Physical View (GRAI-GIM) are presented at that phase even though only functional aspects can be given consideration there. Thus, the CIMOSA and GRAI-GIM Resource and Physical Views at that point can only consider restrictions or constraints not total requirements.

In the case of Purdue, if full resource requirements were developed in the Definition Phase, this would mean that the major tasks presently assigned in the Specification Phase would be shifted ahead into this Definition Phase (see Section 3.1.1). Thus, before the Functional Networks were finished, the Functions involved would have already been assigned to their physical implementation means. In a large project, a phenomenon called "unstable" program execution by Mr. Rathwell, which is caused by making the earlier phases depend on the later phases in program development, would most probably result in major difficulties in program management (see Section 7.4.3).
FIGURE 8.12 COMPARISON OF THE SEMANTICS FOR CIMOSA AND GRAI-GIM
FIGURE 8.13  GRAI-GIM: WORKING STREAMS IN FOUR VIEWS TO THREE TECHNICAL DOMAINS
8.3.4 An Explanation for the Different Development Patterns

One possible explanation for the different multistream development patterns discussed above may be given by the different basic patterns in model processing used. In Purdue a task module (see Section 3.1.3) is considered as the basic modeling unit to be processed. The multiple streams of development of Purdue are generated by the assignment of the resulting task modules. Whereas, in CIMOSA and GRAIGIM, the counterparts to the multiple streams, the four Views, are generated by decomposing the task modules of Purdue into their components, i.e., their constituent parts, "which are the Inputs, Outputs and Transfer Functions" of the system [12].

By adding the development of the basic task modules to Figure 8.11, Figure 8.14 shows that after these Task Modules are generated in the Functional View of PERA, these basic units are preserved intact when they are organized into the two Functional Networks and then assigned their task implementations in the Implementation View. Further discussions can be stated as follows:

1. The generation of the two sets Policies decides that the task modules will be classified into two streams according to the types of Tasks or Transformations which they are carrying out, not anything else.
2. The "virtual" connections between the two streams correspond to the data flows needed for the purpose of Control in real world.
3. The generation of the three Implementation Architectures from the two streams is to further classify the task modules in each stream according to the
types of task implementations assigned to each of them, not anything else.

4. The connections between the task modules of the same type but which have been assigned different task implementors are for coordination between those tasks or transformations which have the same functional requirement, i.e., these connections are interfaces.

5. Arrangements of these interface connections between the assignments of task implementations will not change the "virtual" connections established earlier between the two Functional Networks.

6. The multiple streams of development of PERA give a description of the "natural" process along which an enterprise development program is engineered. This may be another reason why the structure of PERA is considered by many plant workers as the one which best approximates their factories and operations (see Section 1.1)

The pattern of the four-views development in GRAI-GIM (see Figure 4.15 and 8.15 [16, p. 33] for the case of GRAI-GIM, and Figure 8.16 for more general cases) shows little possibility to offer a similar growth pattern as simple as PERA's. This indicates that anyone who adopts one of the multi-view development schemes may well need much more work to handle all kinds of necessary consistency checking, cross-mapping, etc., required during the program development (see Figure 8.16).

It should be noted that if we apply a four-view scheme only to an information/control system, we may find a good match between each view and some combination of the system components. For example, an information view may match with
FIGURE 8.14 TASK MODULES AS BASIC UNIT IN PERA'S MULTIPLE STREAM DEVELOPMENT
FIGURE 8.15  MODELING SEQUENCE OF THE GIM METHOD
The illustration can be used in any combination of system components.
the database system; a function view may match with application software; an organization view with the structure of the human decision-making system; and a resource view with the hardware system which may also include the human operators, etc. However, when we are dealing with development problems in the field of overall enterprise integration, we are usually subjected to a more complicated system and the adoption of the four-view development may make everything even more complicated by unnecessarily breaking down the basic functional modules and bringing up the connections between the components of the modules.

8.3.5 "Divide et impera (Divide and rule)"

When considering every Type Two Enterprise Reference Architecture, it can be seen that no matter whether it is CIMOSA, GRAI-GIM, or Purdue, there is a common character among them in terms of the way in which they are coping with complexity. That is, "Divide et impera (Divide and rule)" — "the technique of mastering complexity has been known since ancient times" [117, p. 3].

The major question is whether differences between the implementation strategies of the three Architectures and the use of "Divide et impera" matter. In this case, it would seem that Prof. Suh's Axioms can be seen as two important principles for evaluating the overall strategy of complexity management. We may read these two Axioms in the following way:

- When you divide, keep the numbers of connections between the modules at a minimum;
The divisions you make should be such that the task implementations in each module are the most easily accomplished.

Not every candidate architecture can withstand the test of Suh’s Axioms. Only the unique phased-wise division used by Purdue seems to have passed this test to date (see also Chapter 6).
9. CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

The topic selected for this study has been that of establishing a formal base for the Purdue Enterprise Reference Architecture (PERA) [19, 20] and its associated methodology for analyzing and developing enterprise integration projects [27]. This work has also had the mission of providing additional methods for the description of PERA and for the application of PERA to integration studies along with a comparison of its relative capabilities in relation to other available enterprise integration architectures.

The work of this report has made the following contributions to PERA and its methodology:

- The establishment of a formal theoretical basis for PERA as an enterprise integration architecture through the application of the Design Axioms of Prof. Suh [84]. Through this particular study, it has been shown that PERA satisfies the requirements for independence of its functional requirements and of the minimum needed information for their application. Use of these Axioms has shown that the structure of PERA as originally derived purely from logical reasoning is the correct one from a theoretical design aspect as well.
A companion study of the interfaces between the resulting phases of the enterprise development program as defined by the Design Axioms shows that those program interfaces used are the minimum necessary to carry out such a project. Formal definition of these interfaces through standardization methods provides the best way of applying formality to an integration project and helps greatly in its conduct. It is noted that one of Prof. Suh's Corollaries to his Axioms also recommends standardization as a minimization technique.

The phases originally developed by PERA and confirmed by the Axiom study form self-contained groups of related tasks which in many cases can be carried out by computer-based tools, either separately or in conjunction with human workers. The work carried out here has shown how the appropriate tools can be associated with each of the development phases to present a matrix of tools versus applicable needs at each location on the PERA diagram. This becomes a simple and readily understood method of specifying such tools. Standardization of these tools again becomes the best method of establishing formality in the enterprise integration design process.

A major Glossary of the field of enterprise integration has been developed in this study by revising and updating that originally produced by the Industry-Purdue University Consortium for CIM [27] in their earlier study of this field.
The above four methods of applying formality to the Purdue Enterprise Reference Architecture form a hierarchy of formalism methods as follows:

- Architectural formalism, defined in Chapter 6 and 7;
- Interface formalism, defined in Chapter 5;
- Design tool formalism as defined in Chapter 5 and 7; and
- Semantic formalism through the Glossary of the Appendix.

The study of Enterprise Genealogy in Chapter 7 points out the potential recursiveness of the PERA diagram in representing enterprise entities. This recursiveness of the PERA diagram shows how the members of one class of entities become the developers of succeeding classes of entities in turn. The success of this action establishes by this further example the applicability of PERA to define any enterprise or systems entity regardless of its origin or field of application. PERA is the only architecture able to show this recursiveness.

The comparative study carried out in Chapter 8 between PERA and the other major enterprise reference architectures serves to amplify the Conclusions above that PERA is currently the overall simplest, yet most complete, representation possible for such an architecture. Each of the others are shown to be both incomplete and also to have major interdependencies and interactions among their major components or phases.
It is interesting to note that the phased approach proposed by the PERA architecture and methodology epitomized the ancient paradigm of "Divide et impera (Divide and rule)" - the technique of mastering complexity which has been known since ancient times [117, p. 3]. The problem which has to be solved in to make the division is the best possible way with the minimum interaction and requiring the minimum information. Suh's Axioms and comparison with competitive methods shows that PERA may well be this simplest and therefore the best architecture and methodology for enterprise integration.

9.2 Recommendations for Further Research

The following recommendation can be made for further studies in the field of enterprise integration architectures and their associated application methodologies:

- A major need in any field is for proof of the concepts involved through experimentation via actual physical systems or by a corresponding computer-based simulation. Because of the nature of this study and the time involved, it has not been possible to carry out such an experimental program along with the work already done here. Thus a major recommendation for needed further work in this field is to establish a series of case studies, either computer-based or from actual industrial examples which would study the actual applicability of the PERA architecture and its associated methodology to such systems. These could and should be from a wide variety of industries and types of enterprises, since
PERA claims applicability in all areas of enterprise endeavor.

It should be noted that there currently is a rapidly expanding application of PERA and its associated methodology in industry today. Success of these endeavors may make the case studies noted above moot in terms of their necessity. It is thus further recommended that the status of industrial applications should be investigated before the extensive simulation or experimental study noted above is initiated.

- Industry's constant need is for the simplest possible methods for carrying out their required tasks and operations, even including those as inherently complex as enterprise integration. While PERA has been shown to be much simpler and more intuitive than all other presently available competitive schemes, here is obviously always room for research to develop ever simpler and more applicable representations and methodologies. The potential for enterprise integration on furthering industrial competitiveness today makes further research in this field very valuable.
LIST OF REFERENCES


[34] Trivedi, A. V., "World Class Manufacturing and the Justification of CIM", in *Proc. the RIA/ISIR Conference, Section 17*, pp. 1–10, Detroit, MI., USA, Oct. 1991. RIA, Ann Arbor, MI.


nical Report 111, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, IN, Mar. 1985.


[81] "The Place of the Human Worker in the Manufacturing Plant of the Future", in Williams [37], Chapter 10.


[83] Li, Hong and Williams, T. J., "The Importance of Interfaces in Enterprise Integration and Their Treatment", The Eighth Workshop Meeting, IFAC/IFIP Task Force on Architectures for Enterprise Integration, Vienna, Austria, June 11–12, 1994.


[100] Petrie, Charles, "Introduction", in Petrie [22], pp. 1–14.


A partial translation by this author of the quotation from The Book of Change as given on the right is as follows:

"The Change began with the Great Creation from which the Yin and Yang were born. From the Yin and Yang, the basic components of the world were created. This can be expressed as The Heavens, The Human Realm and The Earth itself. The Change uses a special set of symbols known as the hexagram which can be used to explore and explain the development mechanism of the universe."

Explanation by this author:

The I Ching (known as The Book of Change in English), which is numbered in China as the first one among Chinese Classical Works, is the primary expression of ancient Chinese philosophy. It is well known throughout the world as a philosophy and as a methodology for making prophesy. The author feels that when the pat-
tern of development of the world is discussed, there is a special relationship between the I Ching and the Purdue Enterprise Reference Architecture as follows:

From this "ONE bears TWO and TWO bears THREE" [Laotse, Daohder-Jing], the relationship to the basic diagram of PERA appears evident:

In The Book of Change, the Yin and Yang can be used to explain characteristics of subjects studied. Usually, the Yang may be considered to represent the features of intelligence, control, leadership, and dynamics, while the Yin may be considered to represent the features of productivity and statics, response to the control exerted by the Yang. Therefore, if we follow the theory of The Book of Change, the information functionality as described in PERA belongs to the Yang category, and the manufacturing functionality under modern enterprise systems can be considered to belong to the Yin category.
The manufacturing machines of the Manufacturing Equipment Architecture will carry out part of the manufacturing functions to produce customer products and services, i.e., Yin functions. The information machines of the Information System Architecture will carry out part of the control functions to guide the manufacturing processes, i.e., Yang functions. The humans of the Human And Organizational Architecture will take care of the remaining part of the manufacturing and the control functions to carry out those functions which are not assigned to the machines. As noted in the *Book of Change*, humans have both Yang and Yin characteristics.

If we further introduce PERA's concept of Enterprise Mission Support and Mission Fulfillment into the above discussion, the Yin and Yang categories will still fit very well. Any Mission Support System will provide the information and control needed by the operations of a modern enterprise. Likewise, the Mission Fulfillment System of the enterprise, by following the instructions from the Mission Support System, will meet its customers' demands for whatever products and/or services are needed or desired. It should be noted that generally speaking, the changing rate of information flow involved in the information functions is faster than the changing rate of the production flow involved in the manufacturing functions.

Thus PERA has provided a framework for the further application of the basic principles stated in the ancient Chinese Book to the modern industrialized world. In return, the system of *The Book of Change* is able to help validate this framework as a universally applicable system architec-
ture to describe and guide development of any man-made modern systems.
This Glossary was initially prepared by a Subcommittee of the Industry-Purdue University Consortium on An Implementation Procedures Manual for Developing Master Plans for Computer Integrated Manufacturing (CIM) [27] under the Chairmanship of Mr. Jim Brosvic of Honeywell, Inc. The Glossary was derived mainly from the following sources plus considerable input from the Consortium:

1. The Honeywell Computer Automated Manufacturing (HCAM) Glossary, developed from Honeywell's participation in the Consortium for Advanced Manufacturing International (CAMi). This Glossary was thoroughly revised by the Purdue Consortium Committee.

2. The Information Flow Model of Generic Production Facility contributed by the Foxboro Company [118] to the work of the CIM Reference Model Committee of the International Purdue Workshop for Industrial Computer Systems [37].

In view of the need for a Generic Enterprise Reference Architecture and Methodology (GERAM) based upon PERA, GRAI-GIM, and CIMOSA as proposed by Drs. Peter Bernus and Laszlo Nemes [101, 103] to the IFAC/IFIP Task Force on Architectures for Integrating Manufacturing Activities and Enter-
prises, a revision and expansion of this Glossary is necessary to clarify the differences in terminology as used by the major reference architectures. Concepts derived from recent development in PERA are also presented, such as those involved in interfaces, and in the tools provided by information technologies. In order to keep this Glossary to a reasonable length, some common words previously included have been removed.

The following Glossary is organized alphabetically. Entries comprise a term in bold followed by one or more definitions of its principal meanings in the context of the Purdue Enterprise Reference Architecture and its associated Methodology. The part of speech of a term is specified in parentheses if it is not obvious from its everyday usage. The subject area of a definition is indicated in slanted style. The sources of a definition is noted in square brackets if the definition is an excerpt from sources other than the Purdue Enterprise Reference Architecture and its associated Methodology. Abbreviations used to indicate parts of speech and subject are as listed below.

The following abbreviations are used for parts of speech:

adj    adjective
n      noun
v      verb

The following abbreviations are used for subject areas:

Gen.    General
Pl.     Plural
Syn.    Synonym

The following abbreviations are used for sources:

APIC    American Production and Inventory Con-
trol Society

CAMI  The Consortium for Advanced Manufacturing International

CIMOSA  CIMOSA publications [2-15] and [22]

CMFG  Cost Management Systems Guide

DEC  Digital Equipment Corporation

SAMA  Scientific Apparatus Manufacturers Association


WEB  Webster's New World Dictionary, 3rd College Edition
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbreviation</td>
<td>(n) The act of product of abbreviating; a shortened form of a work or phase.</td>
</tr>
<tr>
<td>Absolute</td>
<td>Perfect in quality or nature; complete - not mixed; pure - not limited by restrictions, qualifications, or exceptions - positive, certain, absolute proof. Physics: Pertaining to measurements or units of measurement derived from fundamental relationships of space, mass, and time. Pertaining to a temperature scale where zero point is absolute zero.</td>
</tr>
<tr>
<td>Abstraction</td>
<td>(n) The act or process of removing or separating; an abstract idea.</td>
</tr>
<tr>
<td>Academy</td>
<td>A society of scholars or artists.</td>
</tr>
<tr>
<td>Accept</td>
<td>To answer affirmatively; accept an invitation; to consent to pay, as by a signed agreement.</td>
</tr>
<tr>
<td>Acceptance</td>
<td>1. The process by which the customer's representative formally agrees to ownership of a completed system. Syn.: Consent, Acknowledgment. 2. Updating of active order to indicate acceptance of order.</td>
</tr>
<tr>
<td>Acceptance Criteria</td>
<td>An acceptance data package describe the form, fit, manufacturing requirements, characteristics, and performance history of a system. The package includes specifications, standards, drawings, plans, manufacturing orders, inspection data, test procedures, test data and other data required</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Accomplished</td>
<td>To succeed in completing; achieve. Syn.: Accomplish, Attain, Reach, Realize, Score.</td>
</tr>
<tr>
<td>Account</td>
<td>1. (n) A description of events; narrative, a set of reasons; explanation, a record of statement, especially of business dealings or money received or spent, a business arrangement. 2. (v) To consider; regard, account for; to give the reason for; explain, on account of.</td>
</tr>
<tr>
<td>Accountant</td>
<td>An expert in accounting, accountancy.</td>
</tr>
<tr>
<td>Accounting</td>
<td>The bookkeeping methods involved in recording business transactions and preparing the financial statements of a business. A description of event; narrative, a set of reasons; explanation, a record or statement, especially of business dealings or money received or spent, a business arrangement.</td>
</tr>
<tr>
<td>Accounting Information</td>
<td>Information that accounting has gathered and accumulated, such as the cost of labor, material, equipment, tools, and facilities and transformed into the cost of the product, overhead, operating costs, inventory, etc. Syn.: Ledger.</td>
</tr>
<tr>
<td>Accredited Standard Committee</td>
<td>A standards committee accredited to ANSI.</td>
</tr>
<tr>
<td>ACSE</td>
<td>Association Control Service Element. ACSE is one of the application protocols specified by MAP (Manufacturing Automation Protocol).</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active Orders</td>
<td>Details of all entered orders (customer ID, product type, quantity, delivery date, shipping requirements ...).</td>
</tr>
<tr>
<td>Activity</td>
<td>The condition of being active; any specific action or pursuit, a planned or organized thing to do.</td>
</tr>
<tr>
<td>Activity-Based Costing</td>
<td>Alternative accounting method that costs products by the production activities performed during actual manufacture.</td>
</tr>
<tr>
<td>Activity-Based System</td>
<td>An alternative accounting method that costs products by the production activities performed during actual manufacture.</td>
</tr>
<tr>
<td>Actual Cost</td>
<td>An acceptable approximation of the true cost of producing a part, product, or group of parts or products, including all labor and material costs and a reasonable allocation of overhead charges. Actual costs are those labor and material costs that are charged against a job as moves through the production process [APIC].</td>
</tr>
<tr>
<td>Actual Cost System</td>
<td>A cost system which collects direct costs as they are incurred during production, and allocates indirect costs based upon their specific costs and achieved volume.</td>
</tr>
<tr>
<td>Actuator Settings</td>
<td>Output to process equipment, valve position, motor status.</td>
</tr>
</tbody>
</table>
| Adaptive Control                        | A control strategy that automatically changes the type or influence of control parameters to improve overall control systems per-
Ad Hoc Development Method - A development methodology that is, in essence, no methodology. However, it is very popular.

Administration - The act of administering, management; often administration, the executive body of a government, its term of office.

Advertisements - Information directed at the customer with the objective of increasing sales or reaching new markets. Syn.: Ad, Announcement.

Advertising Results - The results of advertising, such as the targeted market, customer surveys, market response, etc.

Advice Support - Assistance from engineering specialists.

AIX - Advanced Interactive Executive. The UNIX operating system delivered by IBM for its mainframe, workstation and PC hardware.

Alarm - (n) An audible or visible signal indicating abnormal or out-of-limit conditions in a plant or a control system [CMSG]. Syn.: Bell, Light, Siren.

Algorithm - A prescribed set of well defined rules or processes for the solution of a problem in a finite number of steps, e.g., a full statement of an arithmetic procedure for evaluating sin x to a stated precision. An explicit, finite set of instructions that is guaranteed to find a solution to a particular problem, although not necessarily by the best or fastest
route.

**Alloy** - A macroscopically homogeneous mixture of solid solution, usually of two or more metals, something added that lowers value or purity.

**Amend** - To correct; rectify, to alter (a law) formally by adding, deleting, or rephrasing, to improve.

**Amendment** - Improvement, a correction, a formal alteration.

**AMIG** - Australian MAP Interest Group (see World Federation).

**AMT** - Advanced Manufacturing Technology.

**Analogue** - Something that is analogous.

**Analysis** - The separation of a whole into constituents with a view to its examination and interpretation, a statement of the results of such a study.

**Anneal** - To heat (glass or metal) and slowly cool to toughen it and reduce brittleness, to temper.

**ANSI** - American National Standards Institute (see Standards Organizations).

**API** - Application Programming Interface. An interface that enables programs written by users or third parties to communicate with certain IBM program products. The facility enables users and third parties to add functions to IBM-supplied software.

**APICS** - American Production and Inventory Control Society.
Appendix  - Supplementary material at the end of a book. Pl.: Appendixes or appendices.

Application  - A user or machine oriented function supported by automation technology. Software packages which execute the functions defined in the model of the enterprise. [CI-MOSA]

Application Process  - An element within a system that performs the information/data processing for a particular application.

Application Software  - Software specifically produced for the functional use of a particular application on a computer system; for example, software of a payroll system, or general accounting ledger.

Architectural Resources  - The integrating elements used to build an enterprise system. Resources can be categorized as interfaces, protocols or handlers and management tools, etc.

Architecture  - 1. The set of principles, rules and standards and other supporting data, classified and presented in an orderly form to illustrate the basic arrangement and connectivity of parts of a system. 2. The formation or construction of an object, device or system whether this is the result of the conscious act of, or the gross or random, disposition of the constituent parts.

Area  - A flat, open surface or space, a specific region, the range or scope of anything, the measure of
a planar region or of the surface of a solid.

ART - Automated Reasoning Tool which is an expert system software development environment from Inference-Corporation. ART provides knowledge engineers with a comprehensive set of knowledge representation and storage techniques and graphics capabilities for building expert systems. [DEC]

Artificial Intelligence - A growing set of computer problem-solving techniques being developed to imitate human thought or decision making processes, or to produce the same results as those processes. [DEC]

ASC - Accredited Standard Committee. A standards committee accredited to ANSI.

As-Is Plant - A description of the subject entity of an enterprise program study at the current time or prior to the program implementation.

ASN.1 - Abstract Syntax Notation One. An ISO standard (DIS 8824 and DIS 8825) that specifies a canonical method of data encoding. This standard is an extension of CCITT Standard X.409.

Aspect - An appearance; air, an element; facet, a position or side facing a given direction.

Assembly - The act of assembling or the state of being assembled; the putting together of parts to make a completed product, a set of parts so assembled: the steering assembly of a truck, the signal calling
troops to assemble.

**Assembly Specification** - The assembly specification establishes detailed assembly procedures for any combination of parts, subassemblies, etc., that perform a specific function [SAMA].

**Assess** - To evaluate.

**Assign** - To specify; designate; to select for a duty: appoint, to give out as a task; allot, to ascribe; attribute. *Law*: to transfer (i.e., property) to another.

**Association** - The act of associating, an organized body of people; society.

**Attitude** - 1. A position of the body or manner of carrying oneself, indicative of a mood or condition; a state of mind or feeling with regard to a person or thing. 2. The orientation of and aircraft's axes relative to a reference line or plane, as the horizon.

**Attribute** - A data fact about an entity or relationship.

**Audit** - 1. A management review of a hardware/software project for the purpose of assessing compliance with requirements, specifications, baselines, standards, procedures, instructions, codes, and contractual and licensing requirements. 2. In an enterprise development program, an activity to determine through investigation the adequacy of, and adherence to, established procedures, instructions, codes, and standards or other applicable
contractual and licensing requirements, as well as the effectiveness of implementation.

**AUTOFACT** - A yearly trade show sponsored by the Society of Manufacturing Engineers. It specializes in the technology of CAD/CAM.

**Automate** - To convert to automation, to operate by automation.

**Automated Assembly** - Assembly by means of operations performed automatically by machines. A computer system may monitor the production and quality levels of the assembly operations.

**Automatic** - 1. (adj) Capable of operating with little or no external control or influence, involuntary; reflex, capable of firing continuously until out of ammunition. 2. (n) A device that is automatic.

**Automation** - 1. The implementation of processes by automatic means; the theory, art or technique of making a process more automatic; the investigation, design, development and application of methods for rendering processes automatic, self-moving or self-controlling; the conversion of a procedure, a process or equipment to automatic operation. 2. (n) Gen. The automatic operation or control of a process, machine, equipment or system, the mechanical and electronic techniques and equipment used to achieve automatic operation or control, the condition of being automatically controlled or operated.

**Available Product** = Inventory + planned production -
accepted orders.

**Available To Promise** - The uncommitted portion of a company's inventory or planned production. This figure is frequently calculated from the Master Production Schedule and is maintained as a tool for order promising.

**Average Inventory** - In an inventory system, this is the sum of one-half the lot sizes plus the reserve stock in formula calculations.

**Axiom** - 1. An established rule, or principle or law of a science, art, etc. 2. A statement or proposition that needs no proof because its truth is obvious, or one that is accepted as true without proof [WEB].

**B**

**Back-up** - A reserve, as of provisions, a person standing by and ready to serve as a substitute, support or backing.

**Backbone** - The trunk media of a multimedia LAN separated into sections by bridges, routers, or gateways.

**Background** - The area, space, or surface against which objects are seen or represented, conditions or events forming a setting, a place or state of relative obscurity, one's total experience, education, and knowledge.

**Backtracking** - An element of a search process that involves returning the database or conditions in a system to
a previous state in order to try all alternative solution paths.

**Backward Chaining**
- A type of system activity that attempts to solve a problem by stating a goal and looking into the database for the conditions that would cause that goal to come about, then reiterating this process, using those conditions as the goals while searching for their preconditions [DEC].

**Bandwidth**
- The number of user data bytes (i.e. exclusive of communications overhead) that can be sent across the network per second.

**Bar Code**
- 1. Array of rectangular marks and spaces in a predetermined pattern depicting object identification, machine performance or other required data; can be numeric, alphanumeric or combinations thereof. 2. Also used to identify finished or semifinished products.

**Base**
- The lowest or bottom part, the fundamental principle or underlying concept of a system or theory, a chief constituent, the fact, observation or premise from which a measurement or reasoning process is begun.

**Baseband**
- A single channel signaling technique in which the digital signal is encoded and impressed on the physical medium.

**Baseline**
- 1. A hardware/software work product that has been formally reviewed and agreed upon, which then serves as the basis for further development, and that can be changed only through formal change
control procedures. Each baseline must specify items that form the baseline (for example, software requirements, design documentation, and deliverable source code), the review and approval mechanisms, the acceptance criteria associated with the baseline, and the customer and project organization who participated in establishing the baseline. 2. A hardware/software configuration identification document or set of documents formally reviewed and agreed on at a specific time during the system lifecycle, which completely describes the functional and/or physical characteristics of a hardware/software configuration item. Baseline, plus approved changes to those baselines, constitute the current hardware/software configuration identification of a product. An example (U.S. Department of Defense (DoD) usage) of system baselines is Functional baseline – The initial configuration established at the end of the requirements definition phase.

Basis - A foundation; a supporting element, the main part, principle; criterion.

Batch - An amount prepared or produced at one time.

Batch Process - An industrial manufacturing method in which one of several units are produced at a time, in contrast to Continuous Process (q.v.).

Baud - Unit of signaling speed. Baud is the same as bits per second only
when every signal event represents exactly one bit.

**Behavior**
- Deportment, demeanor, action, reaction, or function under specified circumstances.

**Benchmark**
- A standard by which something can be judged.

**Benefit**
- An advantage, an aid; help, a payment or series of payments to one in need, a fund-raising public entertainment.

**BER**
- Bit Error Rate. The ratio of bits received in error to total bits received.

**Bias**
- (n) A line cutting diagonally across the grain of fabric, preference or inclination that inhibits impartiality; prejudice. (v) To cause to have a bias; prejudice.

**Bill of Lading**
- A contract or receipt for goods that a carrier agrees to transport from one place to another and to deliver to a designated person or that it assigns for compensation upon the conditions stated therein [APIC].

**Bill of Material**
- A listing of all the subassemblies, parts and raw materials that go into the parent assembly. It shows the quantity of each raw material required to make the assembly. There are a variety of display formats for BOMs, including single level, indented, modular/planning, transient, matrix and costed BOMs [APIC, CMSG].
Bill of Resources - A statement of the key resources required to manufacture one unit of a selected item. It is often used to predict the impact of an item in the master production schedule on the supply of resources [APIC].

Bit - 1. An abbreviation of Binary Digit. 2. A single character is a binary number. 3. A single pulse in a group of pulses. 4. A smallest code element which may possess information in either of two states. 5. An acronym for Binary Digit; the smallest unit of information in the binary numbering system, represented by the digits 0 and 1. 6. The smallest division of PC word.

Bitbus - A serial low speed network developed by Intel Corporation in 1983 for sensor and factory level controlling devices.

Blackboard - A structured workspace on which a system can post information about the internal states of objects or system registers, for consultation and appropriate action in the system by operators [DEC].

Blending - The process of physically mixing two or more lots of material to produce a homogeneous lot. Blends normally receive new identification and require retesting.

Block Diagram - In engineering, a diagram of a system, computer, or a device in which the principal parts are represented by suitable annotated geometrical figures showing both the basic functions of the parts and their functional relation-
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOM Costed</td>
<td>A form of the Bill of Material that extends the quantity of every component by the cost of the components [APIC].</td>
</tr>
<tr>
<td>Bondroom Location</td>
<td>A controlled storeroom for configuration items such as material and parts [SAMA].</td>
</tr>
<tr>
<td>Bottleneck</td>
<td>A facility, function, department, etc., that impedes production.</td>
</tr>
<tr>
<td>Bottom-up Design</td>
<td>In systems engineering, the process of designing a system by identifying low-level components, designing each component separately, then designing structure to integrate the low-level components into larger and larger subsystems until the design is finished.</td>
</tr>
<tr>
<td>Boundary</td>
<td>Something that indicates a border or limit.</td>
</tr>
<tr>
<td>Breakthrough</td>
<td>An act or the place of breaking through an obstacle or restriction, a sudden achievement or success that promotes further progress.</td>
</tr>
<tr>
<td>Bridge</td>
<td>A network device that interconnect two local area networks that use the same LLC (Logical Link Control) but may use different MACs (Media Access Control). A bridge requires only OSI (Open System Interconnect) Level 1 and 2 protocols (Also see Gateway and Router).</td>
</tr>
<tr>
<td>British Thermal Unit (BTU)</td>
<td>The quantity of heat required to raise the temperature of one pound of water by one degree Fahrenheit.</td>
</tr>
</tbody>
</table>
Broadband - A medium based on CATV (Community Antenna Television) technology where multiple simultaneous signals may be frequency division multiplexed.

Broadcast - A message addressed to all stations connected to a LAN (Local Area Network).

Bubble - A representation of a process which, when drawn on a sheet of paper, looks like a circle, thus the term bubble.

Budget - (n) An itemized list of probable expenditures and income for a given period, the sum of money allocated for a particular purpose or time period. (v) To make a budget, to enter or plan for in a budget.

Build - (v) 1. To erect; construct, to fashion; create, to add to; develop, to establish a basis for; build up, to construct in stages or by degrees. 2. To connect separate software modules together to form a complete software program [CIMOSA] [TF].

Building - A structure; edifice, the act, process, art, or occupation of constructing.

Building Blocks - Software modules developed to form a CIMOSA enterprise modelling program. Syn.: Constructs. [CIMOSA] [TF]

Bus - A broadcast topology where all data stations are connected in parallel to the medium (see Topology).
Business - The occupation in which a person is engaged, commercial, industrial, or professional dealings, a commercial establishment, volume of commercial trade, commercial policy or practice, one's concern or interest, an affair or matter.

Business Plan - A statement of income projections, costs and profits usually accompanied by the budgets and a projected balance sheet as well as a cash flow (source and application of funds) statement. It is usually stated in terms of dollars only. The business plan and the production plan, although frequently stated in different terms, should be in agreement with each other. (cf. manufacturing resource planning). Its components may include Profit Objectives, Production Strategy, Inventory Level, Employment Level, Budgets [APIC].

Business Unit - The lowest level of the company which contains the set of functions that carry a product through its life span from concept through manufacture, distribution, sales and service.

Business Unit Scope - This is a description of the boundaries and content of a Business Unit.

Buy-in - The process of acceptance of and active cooperation in the development, promulgation and use of a project, technology or new equipment by those individuals concerned.

By-pass - (n) A road or highway that passes around or to one side of an obstructed or congested area, elec-
tricity, a shunt, a surgically created alternative passage between two blood vessels, an operation to create a by-pass.

By-product – Something produced in the making of something else, a side effect.

Byte – A small unit of data bits that are treated as a single unit. The number of bits in a byte is hardware specific, but is most commonly eight (see Octet).

C

CAD – Computer Aided Design. The part of Computer Integrated Manufacturing restricted to the design of a device or a system with manufacturing functions.

Calendar – (n) A system of reckoning time divisions and days, a table showing such divisions, usually for a year, a chronological list or schedule. (v) To enter on a calendar.

Calibration Data – Data collected and/or used to carry out a calibration scheme and/or to record the results of said calibration.

Calibration Schedule – The objective of a calibration schedule is to detect the deterioration of performance of equipment and tools before the thresholds of performance are exceeded. Calibration schedules are established by classes of equipment and are varied to reflect precision, nature and extent of use. Schedules can be based on elapsed calendar time,
actual amount of usage or actual operating hours [JURA].

Calibration Status - The status of the calibration, such as adherence or deviation from a calibration schedule, etc.

CAM - Computer Aided Manufacturing. The part of Computer Integrated Manufacturing restricted to the operation and control of manufacturing functions.

Capacity - The highest reasonable output rate that can be achieved with the current product specifications, product mix, work force, plan and equipment (cf., Efficiency) [APIC].

Capacity Plan - The capacity requirements plan determines how much labor and equipment are needed to accomplish the tasks of production. The capacity requirements plan translates the production orders into hours of work at a workcenter by time period [APIC].

Capacity Plan Rough - The process of converting the master schedule into capacity needs for critical resources: manpower, equipment, warehouse space, vendor capabilities and money. Often the Bill of Resources is used to accomplish this. The purpose of the Capacity Plan is to evaluate the master schedule prior to implementing it [APIC]. Syn.: Capacity Resource Plan.

Capital - (n) A town or city that is the official seat of government in a political entity, wealth in the form of money or property, the net worth of a business, the funds in-
vested in a business by the owners or stockholders, capitalists considered as a class, an asset or advantage, a capital letter.

(adj) First and foremost; chief; principal, pertaining to a political capital.

Carrier Band - A single channel signaling technique in which the digital signal is modulated on a carrier and transmitted (also see Baseband).

Carrying Cost - The cost of carrying inventory is usually defined as a percent of the dollar values of inventory per unit of time. Carrying Cost depends on cost of capital invested as well as the costs of maintaining inventory, such as taxes and insurance, obsolescence, spoilage and storage. Costs vary from 10 to 35 percent annually. Carrying Cost is an opportunity cost due to alternative uses found for funds tied up in inventory [APIC].

CASA/SME - The Computer and Automated Systems Association of the Society of Manufacturing Engineers. CASA/SME is a professional engineering association dedicated to the advancement of engineering technology. CASA/SME sponsors both the MAP and TOP (Technical and Office Protocol) Users Groups.

CASE - 1. Common Applications Service Elements. CASE is one of the applications protocols specified by MAP, largely superseded by ACSE (op cit). 2. Computer Aided Software Engineering. A computer based tool to assist software engineers in analyzing, designing, coding, testing, and documenting a
software system and managing a software project.

Catalyst - A substance that modifies and especially increases the rate of a chemical reaction without being consumed in the process (catalysis).

Category - A specific division in a system of classification; class.

CATV - Community Antenna Television (see Broadband).

CBEMA - Computer and Business Equipment Manufacturers Association (see Standards Organization).

CCITT - International Consulting Committee on Telephone and Telegraph (see Standards Organizations).

Cell Model - A graphic representation of a human- or machine-directed function, which has elements of input, activity and output.

Centralization - 1. The process of consolidating authority and decision making within a single office or person. 2. The act of bringing together physically or geographically operations or organizational units related by nature of function to form a central grouping.

Certification - The act of certifying, the state of being certified, a certified statement.

Chairman - One who presides over an assembly, meeting, committee, or board - chairmanship - chairperson -
chairwoman.

Champion - (n) One that holds first place or wins first prize in a contest, one who fights for or defends a cause or another person. (v) To fight as champion of; defend.

Changeover Time - The time required to modify or replace an existing facility or workplace, usually including both tear down time for the existing condition and setup for the new condition.

Checklist - Specialized lists of necessary activities based on previous successful development. A checklist can be used effectively with verification and validation methods. Checklists are excellent for catching omissions such as missing items, functions, and products. These are valuable aids in addressing some of the lifecycle feasibility considerations, such as human and organizational design, maintainability, reliability, availability, affordability, security, privacy, and lifecycle efficiency.

CI List - A tabulation of engineering drawings, specifications and other reference documents needed to fabricate and assemble a configuration item [SAMA].

CIM - Computer Integrated Manufacturing (see Computer Integrated Manufacturing).

CIM Architecture - (see Computer Integrated Manufacturing Architecture)

CIMOSA - Computer Integrated Manufacturing
Open System Architecture. An enterprise reference architecture defined and developed under an ESPRIT (the European Strategic Programme for Research and Development in Information Technology) project by AMICE (the European Computer Integrated Manufacturing Architecture [reverse acronym] Consortium).

CIMOSA IIS - CIMOSA Integrating Infrastructure. A software system which may be considered as interface-related services (see CIMOSA Service-Functional Entities) built on top of OSI-based communications facilities to provide a more general computer operating support for integrated manufacturing systems. [CIMOSA]

CIMOSA Service Functional Entities - There are four Service-Functional Entities provided by CIMOSA Integrating Infrastructure (IIS) as follows: [CIMOSA]

- Business Related Services (B) — for control of Business Processes and Enterprise Activities and management of resources, such as dispatching the execution of Enterprise Activities and scheduling the provision of resources, which may be considered technically as interfaces with computerized tools, or management of application programs and procedures;

- Information Related Services (I) — for management of information, such as information administration required by application processes, which may be considered technically as interfaces
with data or management of database;

- Front End Related Services (F) — for management and control of the Implemented Functional Operations, i.e., representing various types of manufacturing resources to the Business Related Services, which may be considered as interfaces with end-users and/or manufacturing equipment or, in other words, management of user interfaces; and

- Communication Related Services — for system wide information exchange between distributed instances of the IIS services B, I and F, which may be considered technically as a network interface manager or management of a network.

The CIMOSA Service-Functional Entities are counterparts of the foundation functional entities of PERA (see Functional Entity).

CIM System - (see Computer Integrated Manufacturing System)

CI Parts List - This list may be either an indented or an alphanumeric listing of all parts making up a configuration item, (CI). The list illustrates the indentured relationship, application, quantities and engineering revision status for all parts and assemblies within the CI [SAMA].

CI Specification - The Configuration Item (CI) Specification establishes the functional, performance and design
criteria for the design, development, testing and production of any combination of parts, subassemblies, units or groups that perform a specific function and is essential to the completeness of a system or subsystem [SAMA].

Closed Loop System - Refers to a feedback control system involving one or more feedback control loops, which combine functions of controlled signals and of commands, in order to keep relationships between the two stable.

CMIG - Canadian MAP Interest Group (see World Federation).


Code ID - A unique number assigned to each company that builds or develops items for the government [SAMA].

Cohesion - Requires that each module is designed to perform a single-well-defined function, and the function is completely contained in the module.

Commit - To do, perform or perpetrate; commit perjury; commit suicide, to consign; entrust, to place in confinement or custody, to pledge (oneself) to a position on some issue.

Common LISP - An implementation of the LISP programming languages that incorporates features that are common to several implementations of LISP [DEC].

Communicate - To make known; impart, to transmit.
Communication - The transfer of information and understanding from one point or person to another person. The basic elements in the process of communication are an information source, encoding, transmission, reception, and decoding.

Company - A group of people, people assembled for a social purpose, companionship; fellowship, a business enterprise; firm.

Compatibility - Using an instruction, program, or component on more than one computer with the same result [DEC].

Competition - The act of competing, a contest - competitiveness.

Completion-Date - The completion date is the date that a task is actually completed.

Completions - Completions represent the quantity of components or a production order that has been completed.

Component - A term used to identify a raw material, ingredient, part or subassembly that goes into a higher level assembly, compound or other item. It may also include packaging materials for finished goods [APIC].

Computer - A functional programmable unit that consists of one or more associated processing units and peripheral equipment that is controlled by internally stored programs and that can perform substantial computation, including numerous arithmetic operations or logic operations, without human intervention [ANSI/IEEE Standard
Computer Aided Software Engineering – Computer based tools to aid programming tasks, CASE tools.

Computer Graphics – A human-oriented system which uses the capabilities of a computer to create, transform, and display pictorial and symbolic data.

Computer Integrated Manufacturing (CIM) – The computer integrated enterprise of manufacturing including the management of required resources; people, organization, material, energy, data, computer technology and automation equipment.

1. Computer Integrated Manufacturing (CIM) is manufacturing supported by information and automation intended to create an overall system, (a) which is responsive to the human and economic environment interpreted on all levels and, (b) which improves the management of the industrial facility.

2. Computer Integrated Manufacturing is the use of computers to streamline the flow of materials and information within a manufacturing organization. The goal of CIM is to increase productivity, product quality and manufacturing flexibility while decreasing cost and time-to-market. It is important to keep in mind that CIM itself is not the goal, but instead a strategy to ensure the long-term survivability of the manufacturing organization.

3. CIM is the strategy by which manufacturers organize the vari-
ous hardware and software components, such as robotics, machine vision, CAD (Computer-Aided Design), CAM (Computer-Aided Manufacturing) and Manufacturing Resource Planning (MRP-II), into a unified system working toward the same goals. There is, however, no hard and fast scientific formula for CIM.

4. Computer Integrated Manufacturing involves the development and implementation of a computer-based information management and automation system for the enterprise which allows the establishment of a business process to:

(1) Automate the information flow of the plant.

(2) Deploy appropriate automation and information technologies wherever they are needed in the plant.

(3) Make optimal use of the capabilities of plan personnel.

(4) Maximize information access at all levels of the system.

(5) Provide timely, accurate and complete information on plant operations wherever and whenever needed with the object of obtaining a competitive advantage for the company.

Computer Integrated Manufacturing (CIM) Architecture - A set of principles and rules for selecting and developing products and standards that can participate
in a CIM system.

Computer Integrated Manufacturing (CIM) System - Refers to an implementation of the CIM architecture to integrate an enterprise. Proper selection of CIM products and standards as part of the system will require characteristics of the particular enterprise and attributes of its data requirements to be defined.

Concept - A general idea or understanding, especially one derived from specific instances.

Conceptual Design - The result of the conceptual phase in defining the product. It begins with the broad product objectives and decomposes these objectives into concept formulation, general design approaches, feasibility evaluations, block diagrams and high-level layouts of the product design [SAMA]. Syn.: Functional Baseline.

Conceptual Model - An abstract representation of an object or phenomenon that provides a common understanding.

Conceptual Schema - Comprising the central description of the various information contents that may be in a database. [CIMOSA]

Conceptual Skills - Degree to which AMT jobs require knowledge of the subject matter as well as the ability to interpret, abstract, and infer.

Configuration - 1. The complete technical description required to build, test, accept, operate, maintain and logistically support a piece of equipment. It includes the physical
and functional characteristics of the equipment [SAMA]. 2. The arrangement of a system or network as defined by the nature, number, and chief characteristics of its functional units. 3. The functional and/or physical characteristics of hardware/software as set forth in technical documentation and achieved in a product [DoD Standard 480B 1988].

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>As-Designed</td>
<td>When a drawing list on the As-Planned list is formally released, the drawing constitutes the As-Designed Configuration requirement. The purpose of this list is to provide a reference point for comparing the As-Built Configuration to the As-Designed Configuration. [SAMA]</td>
</tr>
<tr>
<td>As-Modified</td>
<td>The list which identifies the changes made to a product in the field or at a test site. This report is an updated record of the product as it proceeds through a series of field modifications and changes. It includes the CI and serial numbers, modification drawing number, ECP numbers, date of incorporation, product location, persons making the modification and the results of the modification [SAMA].</td>
</tr>
<tr>
<td>As-Planned</td>
<td>Lists which are used to enable advanced planning by various departments to record interface definitions and systems requirements. These lists are based on the specification tree and design requirements defined in the product specification. The drawings are initially identified in an indented assembly drawing list, and detail drawings are added to the list as the task progresses. The</td>
</tr>
</tbody>
</table>
list is released by engineering [SAMA].

Configuration Change Request - Used to request a change to the approved product configuration [SAMA].

Configuration Control Documents - Documents to control changes in configuration. Configuration Control establishes procedures to process these changes from initiation of the document through analysis and finally to the approval or disapproval and affectively of the document [SAMA].

Configuration Identifier - An alphanumeric designator used to identify configuration elements [SAMA].

Configuration Item (CI) - In configuration management, an aggregation of hardware/software, or any of its discrete portions, that satisfies and end-use function and is designated by the customer for configuration management. CIs may vary widely in complexity, size, and type, from an aircraft, electronic system, or ship system to a test meter, circuit board, or teddy bear. During development and initial production, CIs are only those specification items that are referenced directly in a contract (or equivalent in-house agreement). During the operation and maintenance period, any repairable item designated for separate procurement is a CI. [DoD directive 5010.19]

Configuration Log - A logbook kept with each end item after final assembly. The logbook is used for recording data on all periods of operation, for example, operation history, tests, fail-
ures, repair and replacements, diagnosis and rework [SAMA].

Configuration Management (CM) - In systems engineering, the discipline of identifying the configuration of a hardware/software system at discrete points in time with the purpose of systematically controlling changes to the configuration and maintaining the integrity and traceability of the configuration throughout the system lifecycle.

Consensus - Collective opinion; general agreement. Usage note: The phrase consensus of opinion is considered to be redundant because consensus by itself denotes a general opinion.

Consolidate - To form into a compact mass; solidify, to unite into one system or body; combine.

Consortium - A conglomerate or partnership.

Constraint - Something that restricts, restraint.

Constraints - Actual operating limits of process equipment.

Constraint Rule - A type of rule that applies limits to a search by specifying that its associated pattern is never allowed to occur in a valid solution. These rules help reduce the number of unproductive searches [DEC].

Constructs - Software modules used in CIMOSA modelling approach (see Building Blocks).

Consult - To seek advice or information of,
to exchange views; confer.

Continuous Process - Processes which run at some relatively constant rate for long periods of time. Contrasts with Batch Process (q.v.).

Contractor Planning - Schedules of requests for outside contractor assistance.

Contractual Obligations - Conditions/agreements/contracts with customers, and/or suppliers. By definition these are written and must be honored. Examples are blanket orders, terms and conditions for purchase orders or service requests, etc.

Control - Measurement of performance or action and comparison with established plan or standards in order to maintain performance and action within permissible limits of variance from the standard. It may involve taking corrective action to bring performance into line with the plan or standard.

Control Action - Is the institution of the necessary activity to cause a process, device or system to carry out the tasks assigned to that particular process, device or system.

Controlled Variables - Validated measurements for direct control of pressure, temperature, flow, etc.

Controls IDEF - Inputs to the top surface of the IDEF₀ block representing those items which exercise controls or constraints over the transformation represented by the block.

Conversion - The act of converting or the condition of being converted, a
change in which a person adopts a new religion, the unlawful appropriation of another’s property, football; a score on a try for a point or points after a touchdown.

| Coordination Needs | - Extent to which operators must work with others to obtain the information necessary to perform their jobs. |
| COS | - Corporation for Open Systems. An organization of vendors formed in 1985 to coordinate member company efforts in the selection of standards and protocols, conformance testing, and the establishment of certification. |
| Cost | - (n) An amount paid or required in payment for a purchase, a loss, sacrifice, or penalty; (v) To require a specified payment, expenditure, effort, or loss. |
| Cost-Benefit Analysis | - In engineering, the comparison of alternative course of action, or alternative technical solutions, for the purpose of determining which alternative would realize the greatest cost benefit. |
| Cost Effective | - 1. In management, the cost of an item or services is less than the expected return (in dollars) through the use of that item or service. 2. An item or service is more cost effective, when the benefits or payoff, per unit of cost, is highest among alternatives. |
| Cost Objectives | - Cost goals determined by product cost accounting. |
| Costs Policies | - Marketing costs + profit margin. |
Cost Specifications - Details to allow cost estimating, gross layout, preliminary equipment list.

Council - An assembly called together for consultation, deliberation, etc., an administrative, legislative, or advisory body; Usage note: council and counsel are not interchangeable, although they are related terms. A council is a deliberative assembly, such as a city council. Counsel pertains chiefly to advice and guidance in general.

Counsel - (n) An exchange of opinions and ideas; discussion, advice or guidance, a deliberate resolution; plan, a private opinion or purpose; Pl.: Counsel. (v) To give counsel (to); advise, to urge the adoption of; recommend (Latin consilium).

Counselor, also Counsellor - An adviser, an attorney, especially a trial lawyer - counselorship.

Coupling - Refers to the number of informational and control linkages between two modules. It is desirable to minimize these linkages and make them explicit.

Credible - (adj) Believable; plausible, trustworthy; reliable.

Credits and Limits - Information on credibility of customer, financial situation.

Crisis - A crucial or decisive point or situation; turning point.

Criterion - A standard on which a judgment can
be based; Pl.: Criteria; Usage note: Criteria is a plural form and should not be substituted for the singular criterion.

Critical Success Factors (CSFs)
- The few key areas of activity in which favorable results are absolutely necessary for a particular business entity to reach its goals. "Things must go right."

CSMA/CD
- Carrier Sense Multiple Access with Collision Detection. A method of assigning mastership in a communications network.

Culture
- Development of the intellect through education and training, intellectual and artistic taste and refinement, the arts, beliefs, customs, institutions, and all other products of human work and thought created by a people or group at a particular time.

Customer
- Anyone who makes a demand on the manufacturing entity that is external to the entity, whether they be research and development, corporation management, government authority or product customers, that requires a quality response.

Customer Details
- Customer information (name, address, shipping address, credibility, special needs ...).

Customer Profile
- Information relating to a customer used to facilitate manufacturing activities. This includes such information as customer contact, specific product, product information requirements, shipping requirements and special instructions and anything else related to
customer relations.

Customer Responses - Replies to customer in response to requests for order status, purchase order acknowledgement, technical information, shipping dates, quality information, etc.

Customer Returns, Requests and Feedback - Any contact with a customer of the manufacturing entity. It may be requests for information, requests for manufacturing capacity, feedback on manufacturing performance or product performance, or any output sent to a customer that was returned for failure to meet expectations. They are converted into any further output that is sent to the customer in response.

DAE - Distributed Automation Edition. An IBM software solution to provide tools and techniques that will enable factory floor applications programs to be written that are independent of the distributed data, local area network protocol terminal.

Data - A representative of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or automatic means. The raw material for producing information.

Data and Information - 1. The values stored in a database are referred to as data. They become information only when associated with some definition of their meaning, purpose and relationship with other values. Normally, it
is the user who synthesizes data to derive information that is meaningful. This synthesis can also be performed through the use of artificial intelligence.

2. Data is associated with relevance and semantics, whereas information refers to data to which the relevance and the semantics are already associated.

**Database** - A comprehensive collection of complex, interrelated files or libraries of data which may be used by more than one application.

**Database Management** - 1. IBM's relational database. It runs on the MVS operating system. 2. A set of rules about file organization and processing, generally contained in complex software, which controls the definition and access of complex, interrelated files which are shared by numerous application systems.

**Dataflow Diagram (DFD)** - A graphic representation of an information system, showing data sources, data sinks, storage, and processes performed on data as nodes, and logical flow of data as links between the nodes. It can also be used to represent hardware/software systems.

**Data Flows** - These are like pipelines that carry data between processes, data stores or external agents.

**Data Knowhow** - Technical information on state-of-the-art of technology, RD.

**Data Stores** - Serve as repositories of data produced by processes. These data are then made available to other
specified processes.

DB2 - DATABASE-2. 1. IBM’s relational database. It runs on the MVS operating system. 2. IBM’s “strategic” product for general-purpose information storage, including database management. It is the product in the MVS environment that implements SQL, the SAA database programming interface. As such, DB2 is a building block, the linchpin of IBM’s entire software strategy.

DBMS - Database Management System. A software package that enables end users or application programmers to share data. DBMSs are generally also responsible for data integrity, data access control and automated rollback/restart/recovery.

DCS - Distributed Control System.

DEC - Digital Equipment Corporation.

Decentralization - The extent to which the organizational structure allows lower-level workers to make decisions about their jobs, work flow, and work activities.

Decentralize - To distribute the functions of (a central authority) among local authorities.

Decision - A final conclusion or choice; judgment, firmness of character or action; determination.

Decision-Making - The response to a need or stimulus by means of acquiring and organizing information, processing this information to yield alternative
courses of action, and selecting one course of action from among the alternatives.

DECNET - DEC's proprietary version of ETHERNET.

DECOMNI - DEC Open Manufacturing Network Interface. DEC's version of the OSI MMS protocol to be used for control-level device communications.

Decompose - To break into its constituent parts, to separate into component parts or basic elements, to rot or cause to rot.

Deduction - The process of reaching a conclusion by logical means [DEC].

Delivery Schedule - The required or agreed upon time or rate of delivery of goods or services purchased for a future period.

Demand Management - The function of recognizing the managing all of the demands for products to ensure that the master scheduler is aware of them. It encompasses the activities of forecasting, order entry, order promising, branch warehouse requirements, interplant orders, and service parts requirements.

Demography - The statistical study of human populations.

Department - A distinct usually specialized division of an organization, government, or business.

Dependent Demand - Demand is considered dependent when it is directly related to or derived from the demand for other items or end products. Such de-
mands are, therefore, calculated and need not, and should not, be forecast. A given inventory item may have both dependent and independent demand at any given time.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptive Model</td>
<td>A model that describes facts and relationships but does not contain a basis for evaluation of the system.</td>
</tr>
<tr>
<td>Design Basis</td>
<td>Document that contains the basis of a design to allow further detailed engineering.</td>
</tr>
<tr>
<td>Design Practices</td>
<td>Engineering methods, standards, practices.</td>
</tr>
<tr>
<td>Details</td>
<td>Technical details of new equipment.</td>
</tr>
<tr>
<td>Diagnosis Report</td>
<td>Technical report on malfunction reasons.</td>
</tr>
<tr>
<td>Differentiate</td>
<td>To constitute or perceive a distinction, to make or become different, distinct, or specialized.</td>
</tr>
<tr>
<td>Digitizer</td>
<td>A purely graphical input device (i.e., a CAD/CAM CAE system digitizer for convening locations into storable electronic impulses) with a surface on which a location or a point is selected and then automatically converted into a digital x y coordinate suitable for transmission to a computer.</td>
</tr>
<tr>
<td>Direct Digital Control (DDC)</td>
<td>The use of a digital computer to establish commands to the final control elements of multiple regulatory loops.</td>
</tr>
<tr>
<td>Directory Service</td>
<td>The network management function that provides all addressing in-</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DIS</td>
<td>Draft International Standard. The second stage of an ISO Standard (see IS).</td>
</tr>
<tr>
<td>Discipline</td>
<td>(n) Training intended to produce a specified character or pattern of behavior, controlled behavior resulting from such training a state of order based on submission to rules and authority, punishment intended to correct or train, a set of rules or methods, a branch of knowledge or teaching. (v) To train by instruction and control, to punish, disciplined, disciplining, disciplines.</td>
</tr>
<tr>
<td>Discontinuous</td>
<td>Marked by breaks or interruptions.</td>
</tr>
<tr>
<td>Discrepancy</td>
<td>Lack of agreement; difference; inconsistency.</td>
</tr>
<tr>
<td>Discretion</td>
<td>How and to what extent operators exercise autonomy.</td>
</tr>
<tr>
<td>Discretionary Skills</td>
<td>Degree to which AMT jobs demand autonomous problem-solving ability.</td>
</tr>
<tr>
<td>Dispatching</td>
<td>The selecting and sequencing of available jobs to be run at individual workstations and the assignment of these jobs to workers.</td>
</tr>
<tr>
<td>Distributed Computing</td>
<td>Computing performed within a network of distributed computing facilities. The processors for this type of system usually function with control distributed in time and space throughout the network. Associated with the distributed process are distributed storage facilities.</td>
</tr>
</tbody>
</table>
Distributed Control System  - A series of computer-based devices that operate in conjunction with each other on variety of applications.

Distributed Processing  - A data processing organizational concept under which computer resources of a company are installed at more than one location with appropriate communication links. Processing is performed at the user's location generally on a minicomputer, and under the user's control and scheduling, as opposed to processing, for all users is done on a large, centralized computer system.

Distribution Constraints  - Information which influences the choice of distribution channels including distribution channel priorities, alternates, availabilities, customer requests, procedures, and lead times.

Documentation  - 1. A collection of documents on a given subject. 2. the process of generating a document. 3. Any written or pictorial information describing, defining, specifying, reporting, or results. [ANSI N45.2.10-1973] [ANSI/IEEE Standard 729-1983]

Domain  - 1. An arbitrary boundary defined to limit the scope of an entity/relationship or other model. 2. In a CIMOSA modelling program, a domain is a software module which describes a subset of the subject enterprise which needs to be analyzed as an entity. It is made up of a limited set of domain processes. The first major subdivision of an enterprise to be mod-
eled. The functions described is equivalent to the functions of an application functional entity discussed in PERA.

| Domain Expert | - A domain expert contains the raw knowledge that a knowledge engineer uses to structure the knowledge fuse and formulate the rule base [DEC]. |
| Domain Knowledge Representation | - The knowledge representation that is specific to the application. These may be compound knowledge representations. The advantage of domain specific representations is that they represent knowledge in the most natural form for the knowledge engineer and domain expert. |
| DP | - Draft Proposal. The first stage of an ISO Standard (see IS). |
| DS | - Distributed Systems. Refers to computer systems in multiple locations throughout an organization working in a cooperative fashion, with the system at each location primarily serving the needs of that location but also able to receive and supply information from other systems within the network. |
| Dynamic | - (n) The study of the relationship between motion and the forces affecting motion, the physical forces that produce motion, activity, and change in a field or system, variation in force or intensity. (adj) Marked by energy and vigor; forceful, of or pertaining to energy, force, or motion in relation to force. |
ECHO – The immediate update and display of additions, deletions or data entry for review (i.e., to ECHO – repeat video text on a printer automatically and/or simultaneously).

ECSA – Exchange Carriers Standard Association (see Standards Organizations).

EDI – Electronic Data Interchange. Technology that permits the transfer of electronic information between different parts of an organization or different organizations. EDI uses a variety of protocols and standards depending upon the application and the industry.

Efficiency – A comparative measure of production with cost in terms of energy, time, money, etc.

EIA – Electrical Industries Association (see Standards Organizations).

Electronic Data Processing (EDP) – 1. Data processing largely performed by electronic devices. 2. Pertaining to data processing equipment that is predominantly electronic, such as an electronic digital computer.

Employee – One who works for another.

Employer – A company, enterprise or individual which has other persons working for them.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empower</td>
<td>To assign the privilege of making decisions.</td>
</tr>
<tr>
<td>EMUG</td>
<td>European MAP Users Group (see World Federation).</td>
</tr>
<tr>
<td>Engineering</td>
<td>The science and technology concerned with putting scientific knowledge to practical uses.</td>
</tr>
<tr>
<td>Engineering Change</td>
<td>A revision to a parts list, Bill of Materials drawings, authorized by the engineering department. Changes are usually identified by a control number and are made for &quot;safety&quot;, &quot;cost reduction&quot;, or &quot;functionality&quot; reasons. To effectively implement engineering changes, all affected functions, such as materials, quality assurance and assembly engineering, etc. should review and agree to the changes.</td>
</tr>
<tr>
<td>Engineering Details</td>
<td>Documents and information to vendors or contractors.</td>
</tr>
<tr>
<td>Engineering Equipment Costs</td>
<td>Total cost of engineering and purchased equipment.</td>
</tr>
<tr>
<td>Enterprise</td>
<td>A set of functions that carry out a product through its entire life span from concept through manufacturing, distribution, sales and service; an undertaking; venture; initiative; a business organization.</td>
</tr>
<tr>
<td>Enterprise Development Program Proposal</td>
<td>A logical series of projects to achieve key business requirements.</td>
</tr>
<tr>
<td>Enterprise Integration (EI)</td>
<td>A global approach to coordinate in an integrated way among all the aspects of an enterprise develop-</td>
</tr>
</tbody>
</table>
ment program through its whole life cycle to improve its overall performance. CIM is a subset of Enterprise Integration.

**Entity**
- 1. Is a thing (e.g., a person or a device), a concept, an organization, or an event of interest about which the business cares enough to keep data, usually expressed as nouns in English. 2. An active element within an OSI layer (e.g. Token Bus MAC is an entity in OSI Layer 2).

**Entity/Relationship (E/R) Model**
- Conceptual model of business subject matter consisting of a diagram (showing the structural organization of entities, relationships, and attributes) and formal policy-based definitions of each, usually stored in a dictionary.

**Entity Type**
- A classification of entities satisfying certain criteria.

**Entrepreneur**
- One who organizes, operates, and especially assumes the risk of a business venture.

**Entry**
- Order details (customer ID $+$$ product type and quantity + required delivery data $+$$ special requirements, etc. ...).

**Environment**
- 1. Surroundings. 2. In software engineering, the conditions under which something is expected to be built or operate. Sometimes synonymous with software engineering environment.

**EPA**
- Enhanced Performance Architecture. An extension to MAP that provides for low delay communication between nodes on a single segment.
(see MAP/EPA and MINI-MAP).

**Equipment** - The things with which one is equipped, the act of equipping or the condition of being equipped.

**Equipment Information** - Drawings, instructions, data on installed equipment.

**Equipment Order Request** - Purchase order request for new equipment.

**Equipment Performance** - Actual operating performance of process equipment, power, temperatures, overall condition.

**Equipment Variables** - Validated measurements related to equipment performance, vibration, displacement, pressures, temperatures, corrosion analysis, etc.

**ERP** - Enterprise Resource Planning. A concept developed by Gartner Group describing the next-generation of manufacturing business systems. The next-generation of RPII technology that will facilitate the running of business applications in a much more distributed fashion under the client/server model.

**Established Manufacturing Policy** - The set of rules (i.e., previously established) for operating the example manufacturing plant to achieve the goals of management. It can be articulated and delegated in a general way (e.g., a set of algorithms rather than required human innovation, etc.). The term policy is understood to extend to individual measurements and tolerances prescribed to implement production.

**ETHERNET** - A baseband LAN developed by Xerox
Corporation and supported by Intel, DEC, Hewlett-Packard and others. It uses a bus topology with CSMA/CD access control.

**Excellence**
- The quality or condition of being excellent; superiority, something in which one excels.

**Exception Policies**
- Rules and guidelines from marketing to handle waiver and special requirements when accepting an order.

**Executability**
- 1. In software engineering, a factor pertaining to requirement specifications. It describes the extent to which functional simulation and components can be constructed from the requirements specifications before starting the design or implementation. 2. A factor pertaining to reference architectures (model) that describes the extent to which the relevant architecture (model) can be translated by engineers and/or computers into a workable program to pursue the subject in practice.

**Executive**
- (n) A person or group having administrative or managerial authority in an organization. (adj) Pertaining to or capable of carrying out plans, duties, etc.

**Expert**
- A person with a high degree of knowledge or skill in a particular field.

**Expert System**
- A computer system that embodies the specialized knowledge of one or more human experts and uses that knowledge to solve problems [DEC].
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation Facility</td>
<td>A feature of many expert systems that tells what steps were involved in the process by which the system arrived at a solution. These facilities can be simple traces of steps, or they can be more complex, supplying encoded reasons why the solution uses one alternative rather than another. [DEC]</td>
</tr>
<tr>
<td>Exploratory Programming</td>
<td>A set of techniques developed to deal with problems for which the design of a solution cannot be known in advance. Supporting techniques include interactive editing and debugging integrated programming environments, and graphics oriented user interfaces. [DEC]</td>
</tr>
<tr>
<td>External Agents</td>
<td>Components outside the model boundaries that send data to and receive data from the functional entities (process).</td>
</tr>
<tr>
<td>External Influence</td>
<td>A functional entity (external entity) that is separate from the production plant and does not take part in its internal on-going operations but whose actions can have an effect upon the future operation of the plant. They may be part of the company in question or may be units of a separate company working with the functional entities of the production plant.</td>
</tr>
<tr>
<td>External Schema</td>
<td>Specifies how information extracted from the conceptual schema is presented to users in the outside world. [CIMOSA]</td>
</tr>
</tbody>
</table>
Facilitate - To make easier; assist.

Facility - Ease in doing, resulting from skill or aptitude, often facilities. The means to facilitate an action or process.

Factor - One who acts for another; agent, one that actively contributes to a result or process; mathematics; one of two or more quantities having a designated product.

Failure Report - A report describing the failure of a piece of equipment or item, its cause, and the corrective action taken to prevent the recurrence of the failure [SAMA].

Feedback - The determination of the degree or manner of accomplishment of the control action and the use of the information to assure that the control action is accomplished; degree and type of feedback needed by the equipment about a variety of different factors, such as tolerance, temperature, density, and so on.

Feedback Control - A type of system control obtained when a portion of the output signal is operated upon and fed back to the input in order to obtain a desired effect.

Feedforward Control - A control strategy that converts any control systems upset or disturbance into corrective action so that deviations to the control variable can be minimized.
Fiber - (see Fiber Optics)

Fiber Optics - A medium that uses light conducted through glass or plastic fibers for data transmission.

Field Bus - A standard under development in ISA SP50 for a bus to interconnect process control sensors, actuators, and control devices.

Finish Specification - The finish specification establishes the method and requirements for protective treatments and finishes for materials, parts, etc. [SAMA].

Finished Goods - A product sold as a completed item or repair part or any item subject to a customer order or sales forecast. All manufacturing operations, including final test, have been completed. These may be finished parts, renewal parts or finished products [APIC]. Syn.: Finished Product, End-Item.

Finite Loading - Conceptually, the term means putting no more work into a factory than the factory can be expected to execute. The specific term usually refers to a computer technique that involves automatic shop priority revision to level load operation by operation.

FIPS - Federal Information Processing Standards (see NBS).

Fixed Costs - An expenditure that does not vary with production volume, such as rent, property tax, administrative salaries, etc. [APIC].

Flow Rate - 1. Running rate: the inverse of
cycle time, for example, 360 units per shift [APIC]. 2. Continuous/liquid/fluid/gaseous flow in terms of units/second-minute-hour.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMS</td>
<td>Flexible Manufacturing Systems.</td>
</tr>
<tr>
<td>Forecast</td>
<td>An estimate of the future demand.</td>
</tr>
<tr>
<td>Forecasted Orders</td>
<td>Expected orders to deliver within a period of time.</td>
</tr>
<tr>
<td>Formal</td>
<td>1. In engineering, according to accepted definitions, rules, standards or conventions to achieve a disciplined approach and improve the program/product quality. 2. In hardware/software development, according to rigorous mathematical formalism or techniques if available.</td>
</tr>
<tr>
<td>Forward Chaining</td>
<td>A type of system activity that applies operators to a current state in order to produce a new state until the solution is reached. In an expert system, a forward-chaining rule detects certain facts in the database and takes action because of them. [DEC]</td>
</tr>
<tr>
<td>Frame</td>
<td>A knowledge representation technique based on the idea of a frame of reference. A frame carries</td>
</tr>
</tbody>
</table>
with it a set of slots which can represent objects that are normally associated with the frame's subject, allowing frame-based systems to support inferences.

**Framework System** - A type of artificial intelligence systems-building tool designed to reduce the amount of time required to develop an expert system. A knowledge engineer customizes a framework system for a specific application by building a knowledge base for the problem domain of interest.

**FTAM** - File Transfer Access and Management Protocol (ISO DP 8571). FTAM is one of the application protocols specified by MAP and TOP. (DP - Draft Proposal)

**Function** - A group of tasks that can be classified as having a common objective within a company.

**Functional Entity** - That cohesive collection of elements (humans, machines, computers, control devices, computer programs (any or all)) required to carry out one or more closely related tasks or transformations which comprise a recognized function of the manufacturing plant in fulfilling the established manufacturing policy of the company, e.g., production units or staff departments, etc.

A functional entity may contain other functional entities.

An application functional entity is involved in carrying out the primary mission of the manufacturing plant in question as outlined
by the established manufacturing policy of the company. It is directly concerned with the handling and control of raw materials, intermediates and products of the company. The principles of autonomy and locality apply to these entities.

Application functional entities serve as sources and/or sinks of process operational data in the problem domain. They are made up of manufacturing specific functional entities and the physical means of production or plant production media.

A foundation functional entity is a cohesive collection of elements (possibly shared) that carry out a generic supporting function. It does not necessarily obey the principles of autonomy and locality in its operations. Examples of foundation functional entities are:

1. Communications
2. Control Library
3. Databases
4. Graphics Packages
5. Hardware
6. Man-Machine Interfaces
7. Operating Systems
8. Sensor Management
9. Statistical Quality Control Systems
10. Etc.

Manufacturing specific functional entities are commonly elements of larger applications functional entities but may be listed as separate entities in their own right. They form the parts of the application functional entities which are included in the plant's integrated information and automation system in contrast to the plant production media which carry out the physical production steps and material handling functions of the plant. Manufacturing specific functional entities will commonly include foundation functional entities within their make-up. Examples of manufacturing specific functional entities are:

1. Computer System Configurations
2. Cost Accounting
3. Inventory Management
4. Maintenance Planning
5. Order Entry
6. Product and Process Planning
7. Product Shipping Administration
8. Purchasing (Raw Material and Spares)
9. Quality Control
10. Resource Management Scheduling
11. Etc.

Plant production media functional entities comprise those physical production machines, equipments and devices including material handling, which move, position, and transform raw materials into the desired products of the manufacturing enterprise.

**Functional Requirement**

- A specification constraining the way in which a given task is to be performed, the results to be obtained (speed, accuracy, etc.) as well as the elements of the functional entities involved (initiator, source, receptor, etc.).

**Function View**

- In CIMOSA, description of the functional structure required to satisfy the objectives of the enterprise and related control structure, i.e., the rules which define the structures and the principles of the underlying business processes, control sequences or flow of action within the enterprise. [CIMOSA]

Note: Although PERA has its Functional View with purpose of describing the functional structure of an enterprise, the overall functional partitioning is different between the Function View of CIMOSA and the Functional View of PERA. The latter divides the functional structure of an enterprise into Information Functional Network and Customer Product and Service Functional Network according to different transformations performed by different task modules, whereas, the former develops
a multiple-view modeling approach to describe the different input/output of the modules.

**Fundamental**
- Elemental; basic, central; key, generative; primary.

**G**

**Gateway**
- A network device that interconnects two networks that may have different protocols (see Bridge and Router).

**Gearbox**
- An industrialized personal computer developed by IBM that is configured much like a programmable logic controller.

**General**
- Applicable to or involving the whole or every member of a group, widespread; prevalent, not restricted or specialized, true or applicable in most, but not all cases, not precise or detailed, diversified, highest or superior in rank.

**Generic**
- Of, including or indicating an entire group, not protected by a trademark or trade name.

**Generic Level**
- 1. A reference catalog of basic constructs used for the widest application in CIM. [CIMOSA] 2. A comprehensive collection of principles, rules, standards, and tools for any enterprise development programs.

**Glossary**
- A list of difficult or specialized words with their definitions.

**GM C4**
- The General Motors program that
look to set a variety of standards for all levels of manufacturing applications. The two parts of the program that are most developed include standards for communications technology, as well as CAD/CAM software and hardware.

**Goals**

- Specific targets for a period of time.

**Gross Margin**

= Selling price $-$ product cost.

**Gross Requirements**

- The total of independent and dependent demands for a part or an assembly prior to netting on-hand and scheduled receipts [APIC].

**Grouping Departments**

- An organizational structure where the main departments of the organization are grouped by distinctly different production processes—such as technology, flow, or parallel processes. [A. Majchrzak]

**Grouping Departments by Process**

- An organizational structure where the main departments of the organization are grouped by distinctly different production processes—such as technology, flow, or parallel processes.

**Grouping Departments by Products**

- An organizational structure where the main departments of the organization are grouped by major product lines.

**Group Technology Classification by Process**

- A means of coding parts or processes based on the similarities of the parts or grouping parts into product families or grouping production equipment together to produce a family of parts [CMSG]. It provides for rapid retrieval of
existing designs and anticipates a cellular-type equipment layout [APIC]. In this context it is the code or classification for a process.

Group Technology Classification by Product

- A means of coding parts or processes based on the similarities of the parts or grouping parts into product families or grouping production equipment together to produce a family of parts [CMOG]. It provides for rapid retrieval of existing designs and anticipates a cellular-type equipment layout [APIC]. In this context, it is the code or classification for a process.

Group Technology Data

- A means of coding parts or processes based on the similarities of the parts or grouping parts into product families or grouping production equipment together to produce a family of parts [CMOG]. It provides for rapid retrieval of existing designs and anticipates a cellular-type equipment layout [APIC].

GUI

- Graphical User Interface. A generic user interface. IBM has made it part of SAA and supports it on the PS/2. Other examples of a GUI include UII's OpenLook, OSFs Motif and Apple's Macintosh interface. A comprehensive GUI environment includes four components: a graphics library, a user interface tool kit, a user interface style guide and consistent applications. A graphics library provides a high-level graphics programming interface. The user-interface tool kit, built on top of the graphics library, provides
application programs with mechanisms for creating and managing the dialogue elements of the WIMPS interface. The user interface style guide specifies how applications should employ the dialogue elements to present a consistent, easy-to-use environment (i.e., "feel") to the user. Application program conformance with a single user interface style is the primary determinant of ease of learning and use and, thus, of application effectiveness and user productivity.

Guidelines
- Provided to ensure a consistent application of a Reference Architecture and/or Reference Shells when creating particular models. [CIM-OSA]

H

Hardware
- Physical equipment, as opposed to the computer program or method of use; e.g., mechanical, magnetic, electrical or electronic devices; contrast with software.

Hardware Configuration Item (HCI)
- In system engineering, a hardware entity which has been established as a configuration item. The hardware configuration item exists where functional allocations have been made which clearly delineate the separation between equipment functions and software functions and the hardware has been established as a configuration item.

Hardware Interface
- Interfaces by hardware systems and equipment to the external world even though, for example, the ultimate interface in a communica-
tion system may be another piece of software in another computer.

Heterogeneous - Consisting of dissimilar elements or parts; not homogeneous.

Heuristic - A process that may help in the solution of a problem, but that does not guarantee the best solution, or any solution. [DEC]

Hierarchy - 1. Gen. A body of clergy, organized or classified according to rank or authority, an arrangement of persons or things in a graded series. 2. A data structure consisting of sets and subsets such that every subset of a set is a lower rank than the data of the set. 3. Any structure consisting of units and subunits where the subunits are of lower rank than the units involved.

Human Factors - The field of effort and body of knowledge devoted to the adaptation and design of equipment for efficient and advantageous use by people considering physiological, psychological and training factors.

Human Interface - A tool able to intercept, interpret and guide the interaction of the end user with the system.

Human-Machine Redundancy - Extent to which operators should be involved in tasks performed by machines as a check on machine operation.

Human Relations Skills - The ability to communicate, engage in group interaction, and coordinate with others to solve problems.
Hurdle Rate - The expected rate of return on capital which a proposed project must meet in order to be accepted by management.

ICON - A symbol to which a computer can point an interface in order to select a function such as "move window" [DEC].

IDEF - ICC DEFinition Language. A systems modelling technique using a specific graphical structure.

IEEE - Institute of Electrical and Electronic Engineers (see Standards Organizations).

IEEE 802 - One of the standards committees working on LAN standards. IEEE 802 has produced standards for CSMA/CD, Token Bus, Token Ring, and Logical Link Control (LLC). Activity continues in all of the above areas and in the area of Metropolitan Area Networks. IEEE 802 is composed following WGs (working groups) and TAGs (technical assistance groups):

IEEE 802.0 - Executive Committee

IEEE 802.1 - Higher Layer Interface

IEEE 802.2 - Logical Link Control

IEEE 802.3 - CSMA/CD

IEEE 802.4 - Token Bus

IEEE 802.5 - Token Ring
IEEE 802.6 – Metropolitan Area Network

IEEE 802.7 – Broadband TAG

IEEE 802.8 – Fiber Optics TAG

IEEE P1118 – A standards committee working on the development of a “Microcontroller Serial Control Bus”. This standard is to be technology-based, not application-based and is intended to be suitable for many different application types, including (but not limited to) instrumentation, process control, and RS232-type peripherals.

Image Processing – The examination by a computer of digitized data about a scene and the features in it in order to extract information [DEC].

Implement – (n) A tool or utensil. (v) (ment) to put into practice, to supply with implements.

Implementation – An actual, tangible system and the result of the conversion of the “what” to “how”.

Implode – Compression of detailed data into a summary level record or input.

Incoming Confirmation – Updating of incoming material status to release payment.

Indirect Cost – Costs that are not incurred by a particular job or operation, such as utilities, management salaries, material handling, data processing, etc. [APIC]

Industrial Computer – A personal or process control computer that is designed to with-
stand the rigors of the factory floor. Typically, the device is encased in a metal or EMI (Electro Magnetic Interference) proof casing. In addition, some industrial computers are configured so that maintenance and cold start up are relatively simple. Such devices are used for applications such as data collection, monitoring and programming.

Inference - A conclusion based on a premise. [DEC]

Inference Engine - The part of a rule-based system that selects and executes rules. The conclusion that an inference engine will draw from a given set of facts that is not known in advance. [DEC]

Information - 1. Gen. The act of informing or condition of being informed; communication or reception of knowledge, knowledge derived from study, experience, or instruction; facts, a non-accidental signal used as an input to a computer or communications system. 2. The knowledge of facts, measurements and requirements necessary for accomplishing useful work. Information has a semantic context in contrast to data.

Information Needs - Type, source, and amount of information that the operator must have to operate equipment effectively.

Information View - Describes the information required by each function. Three kinds of schemata are used to model the information, namely; conceptual, internal and external. [CIMOSA] [ANSI/X3/SPARC and ISO]
Initiative – The power or ability to begin or follow through with a plan or task; enterprise, a first step or action, the procedure by which citizens can propose a law by petition and ensure its submission to the electorate.

In-Process Inventory – Product in various stages of completion throughout the factory, including raw material that has been released for initial processing and completely processed material awaiting final inspection and acceptance as finished product or shipment to a customer.

Insight – The capacity to discern the true nature of a situation; penetration.

Inspection Procedure – A tailor-made plan for a specific component or product type. It lists the characteristics to be checked, the method of check (visual, gage, etc.) and the instruments to be used. In addition, it may include the seriousness classification of characteristics: tolerances and other criteria, a list of applicable standards, sequence of inspection operations, sample plans and other lot criteria. [JURA]

Installation Details – Documentation, drawing information and instruction for construction.

Installation Updates – Documentation, drawing information and instruction for new installed equipment.

Intangible – Not capable of being touched; lacking physical substance, not capable of being perceived; vague.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrate</td>
<td>To make into a whole; unify, to join with something else; unite.</td>
</tr>
<tr>
<td>Integrated System</td>
<td>A system in which separate programs perform separate functions with communication and data-passing between functional programs performing standardized I/O routines and a common data-base. Such systems allow flexibility in addition/revision/deletion of various processing functions without disrupting the entire system.</td>
</tr>
<tr>
<td>Integration</td>
<td>The extent to which components of the production process are inextricably linked. A software design concept that allows users to move easily between applications. [DEC]</td>
</tr>
<tr>
<td>Interface</td>
<td>A shared boundary; e.g., a hardware component to link two devices, a portion of storage or registers accessed by two or more programs.</td>
</tr>
<tr>
<td>Interlisp</td>
<td>A general purpose environment for building and using artificial intelligence applications based on the LISP programming language. [DEC]</td>
</tr>
<tr>
<td>Internal Schema</td>
<td>Describing how the information present in the conceptual schema is structured and stored inside the system. [CIMOSA]</td>
</tr>
<tr>
<td>Interpersonal</td>
<td>Of, relating to, involving, or being relations between persons.</td>
</tr>
<tr>
<td>Interrelate</td>
<td>To place in or come into mutual relationship.</td>
</tr>
<tr>
<td>Inventory</td>
<td>Parts and material on hand.</td>
</tr>
</tbody>
</table>
Inventory is Items that are in a stocking location or work in process and that serve to decouple successive operations in the process of manufacturing a product and distributing it to the consumer. Inventories may consist of finished goods ready for sale; they may be parts or intermediate items; they may be work in progress; or they may be raw materials [APIC].

**Inventory Management** - Management of the inventories, with the primary objectives of determining: (1) Items that should be ordered, and in what quantity. (2) The timing of order release and order due dates. (3) Changes in the quantity called for and the rescheduling of orders already planned. Its two broad areas are inventory accounting, which is the administrative aspect, and inventory planning and control, which consists of planning procedures and techniques that lead to inventory order action.

**Inventory Valuation** - The value of the inventory at either its cost or its market value. Because inventory value can change with time, some recognition is taken of the age distribution of the inventory. Therefore, the cost value of inventory is usually computed on a first-in, first-out, or last-in, last-out basis or a standard cost basis to establish the cost of goods sold [APIC].

**Invoice** - Invoice of performed transport, of RM or other supplies.

**IS** - International Standard. The third (and highest) stage of an ISO
Standard. Prospective ISO standards are balloted three times. The first stage is as a Draft Proposal (DP). After a Draft Proposal has been in use a period of time (typically 6 months to a year) the standard, frequently with corrections and changes, is re-balloted as a Draft International Standard. After the Draft International Standard (DIS) has been in use for a period of time (typically 1 to 2 years) it is re-balloted as an International Standard (IS).

ISA – Instrument Society of America (see Standards Organizations).

ISA SP50 – A standards committee working on a standard of a communications bus for interconnecting control device to sensors and actuators (Field Bus).

ISA SP72 – A standards committee working on a standards for use in process control. These standards include PROWAY, Process Control Architecture, and Process Messaging.

ISDN – Integrated Systems Digital Network. ISDN is a suite of protocols being defined by CCITT to provide voice and data services over wide area networks (WANs).

ISO – International Standards Organization (see Standards Organizations).

Issue – (n) An act or instance of flowing, passing, or giving out, something produced, published, or offered, the result of an action. (v) To go or come out.
Iterate

- 1. To say or perform again, repeat.
- 2. Repeat certain procedures until preset goals or conditions are satisfied.

ITI

- Industrial Technology Institute. A nonprofit organization founded by the University of Michigan and sponsored by the State of Michigan dedicated to computer integrated manufacturing. ITI offers MAP conformance testing and certification.
JMUG – Japanese MAP Users Group (see World Federation).

Journal – A daily record of occurrences or observations, an official record of daily proceedings or transactions, as of a legislative body, a newspaper, a specialized periodical, the part of a shaft or axle supported by a bearing.

Justify – To demonstrate to be just, right, or valid, to provide sound reasons for.

Just-In-Time – A method of controlling and reducing direct and work-in-process inventory by having suppliers deliver material “just-in-time” to manufacturing.

KEE – An artificial intelligence language that emphasize objects as the basic component of an expert system. [DEC]

Kit – Components of an assembly that have been pulled from stock and readied for movement to the assembly area [APIC]. A kit is a collection of carefully identified and controlled items used to build a module, subassembly or assembly. Kit items are usually kept in a box or bag and labeled [SAMA].

Kit List – A tabulation of all parts and materials that go into a kit. The list includes part and serial numbers, lot numbers or date codes
for each item listed [SAMA].

**Kitting** - The process of removing components of an assembly from the stock room and sending them to the assembly floor as a kit of parts. This action may take place automatically whenever a full set of parts is available and/or it may be done only upon authorization by a designated person.

**Knowledge** - The state or fact of knowing, familiarity, awareness, or understanding gained through experience or study, the sum or range of what has been perceived, discovered, or inferred, learning; erudition.

**Knowledge Acquisition** - The process of extracting knowledge. It can be derived from many sources: an expert, documents, books, manuals, forms, etc.

**Knowledge Base** - The part of an artificial intelligence system that contains structured, codified knowledge and heuristics used to solve problems. [DEC]

**Knowledge Craft** - A complete expert development system which provides an integrated set of programming tools. [DEC]

**Knowledge Engineer** - A person who implements an expert system. A knowledge engineer interviews experts to utilize the raw knowledge from which to structure the knowledge base and formulate the rule base. [DEC]

**Knowledge Representation** - A structure in which knowledge can be stored in a way that allows the system to understand the relationships among pieces of knowledge and to manipulate those relation-
Labor Costs - Dollar amounts of added value due to labor performed during manufacturing [APIC].

Labor Force Mobility - Availability of alternative jobs outside the organization to which members of the work force can turn if dissatisfied with plant management.

Labor Grade - A classification of labor into separate groupings of those whose capability makes them unique in terms of their particular skill level or craft [APIC].

LAN - Local Area Network. Local area networks are a communications mechanism by which computers and peripherals in a limited geographical area can be connected. They provide a physical channel of moderate to high data rate (1-20 Mbit) which has a consistently low error rate (typically 10^-9).

Language - A systematic means of communicating ideas by the use of conventionalized signs, sounds, gestures, or marks as well as rules for the formation of admissible expressions. [ANSI/IEEE Standard 729-1983]

Layer - Subdivision of the OSI architecture (See OSI Reference Model).

Lead - To guide, conduct, escort, or direct, to influence; induce, to be ahead or at the head of, to pursue; live; to tend toward a cer-
tain goal or result; to make the initial play.

**Lead Time** - A span of time required to perform an activity. In a logistics context, it is the time between recognition of the need for an order and the receipt of goods. Individual components of lead time can include order preparation time, queue time, move or transportation time, and receiving and inspection time [APIC].

**Level of Skill** - Average skill level and experience of the work force at the plant.

**Life Cycle** - The series of changes in form undergone by an organism in development from its earliest stage to the next recurrence of the same stage in the next generation [WEB].

**Line Driver** - A circuit specifically designed to transmit digital information over long lines, that is, extended distances.

**LISP** - A programming language (LIST Processing) designed specifically to manipulate symbols rather than numeric data. A LISP data element is a list of symbols that may represent any object, including its own list processing functions. [DEC]

**LISP Interpreter** - A part of any LISP-based software tools that allows specific list-processing operations such as match, join, and substitute, to execute on a general purpose computer rather than a special purpose LISP machine. [DEC]

**LISP Machine** - A single-user workstation with a
dedicated LISP programming architecture. [DEC]

List - In a list processing software language, a list is an ordered sequence of elements. [DEC]

LLC - Logical Link Control. The upper sublayer of the data link layer (Layer 2) used by all types of IEEE 802 LANs. LLC provides a common set of services and interfaces to higher layer protocols. Three types of services are specified:

Type 1: Connectionless. A set of services that permit peer entities to transmit data to each other without the establishment of connections. Type 1 service is used by both MAP and TOP.

Type 2: Connection oriented. A set of services that permit peer entities to establish, use, and terminate connections with each other in order to transmit data.

Type 3: Acknowledged connectionless. A set of services that permit a peer entity to send messages requiring immediate response to another peer entity. This class of services can also be used for polled (master-slave) operation.

Logical Cells - Consist of a set of specified components and support a set of Enterprise Activities. [CIMOSA]

Logistics Documents - Documents to support field operations, such as instruction manu-
als, training information, spares, modification and maintenance information [SAMA].

Lot - A quantity of items produced together and sharing the same production costs and resultant specifications [APIC].

Lot ID - A unique identification number or code assigned to a homogeneous quantity of material (e.g., a batch number or mix number) [APIC].

Lot Traceability - The ability to identify the lot or batch numbers of consumption and/or composition for manufactured, purchased and shipped items. This is a federal requirement in certain regulated industries [APIC].

LPC - Local Procedure Call.

LSAP - Link Service Access Point (see SAP).

M

MAC - Media Access Control. The lower sublayer of the Data Link Layer (Layer 2) unique to each type of IEEE 802 Local Area Networks. MAC provides a mechanism by which users access (share) the network. The MACs defined by IEEE 802 are IEEE 802.3 CSMA/CD, IEEE 802.4 Token Bus, IEEE 802.5 Token Ring, and IEEE 802.6 Metropolitan Area Network (still under study).

Macro Function - 1. A LISP function which serves as a template for translating a LISP form (language structure) [DEC].
2. In PERA, a macro function is a composition of sets of function modules or task modules with the same objectives.

Maintenance - Any activity intended to eliminate faults or to keep hardware or programs in satisfactory working condition, including tests, measurements, replacements, adjustments and repairs.

Maintenance Costs - Total calculated maintenance cost report by work order, time period ...

Maintenance History - Technical details on performed work, diagnosis, used parts, etc.

Maintenance, Preventive - Includes preventive maintenance information or services.

Maintenance Request - Request for repair of equipment, identification of systems, reason ...

Make-to-Order Product - The end item is finished after receipt of a customer order. Frequently, long lead time components are planned prior to the order arriving to reduce the delivery time to the customer. Where options or other subassemblies are stocked prior to customer orders arriving, the term "assemble to order" is frequently used.

Make-to-Stock Product - The end item is shipped from finished goods "off the shelf", and therefore, is finished prior to a customer order arriving.

Management - 1. The process of utilizing material and human resources to accomplish designated objectives. It involves the activities of plan-
ning, organizing, directing, coordinating and controlling. 2. That group of people who perform the functions described above.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management Decision-Making Style</td>
<td>Extent to which management prefers that decisions in the organization be made using a consultative, consensual, delegative, or authoritarian style.</td>
</tr>
<tr>
<td>Management Human Resources Priorities</td>
<td>Top management's attitudes about the importance of such human resource (HR) issues as training and group process skills for workers.</td>
</tr>
<tr>
<td>Management Information System (MIS)</td>
<td>The use of computers and computer software specifically to aid in the management of personnel; an automated system designed to provide management with the information required to make basic decisions. [IEEE Computer Applications Technology Glossary 1986]</td>
</tr>
<tr>
<td>Management Model of Worker Motivation</td>
<td>Extent to which managers feel workers are motivated by the challenge of the job, as opposed to money.</td>
</tr>
<tr>
<td>Manual Dexterity Skills</td>
<td>Degree to which a particular task requires the motor abilities of material handling, picking, and placing, etc.</td>
</tr>
<tr>
<td>Manufacture Goods (Activity)</td>
<td>The activity involving all steps necessary to manufacture, warehouse, and ship product, including maintenance of manufacturing facilities and engineering and construction of plant upgrades.</td>
</tr>
<tr>
<td>Manufacturing Lead-Time</td>
<td>The total time required to manufacture an item. Included here are order preparation time, queue</td>
</tr>
</tbody>
</table>
Manufacturing Method - The method used to manufacture an item. There may be a choice between different manufacturing methods, such as machining a part versus bending it out of sheet metal.

Manufacturing Planning - The function of setting the limits or levels of manufacturing operations in the future, consideration being given to sales forecasts and the requirements and availability of personnel, machines, materials and finances. The manufacturing plan is usually in fairly broad terms and does not specify in detail each of the individual products to be made but usually specifies the amount of capacity that will be required.

Manufacturing Policy - The set of methods and procedures for operating the manufacturing plant to achieve the goals of management. The term policy is understood to extend to individual measurements and tolerances prescribed to implement production.

Manufacturing Requirements/Policies/Plans - These are quality, product, and/or environmental requirements and/or policies. These can be internal (to the company) or governmentally generated. Also included here are manufacturing budgets and plans.

Manufacturing Resource Planning - A method for the effective planning of all the resources of a manufacturing company. Ideally it addresses operational planning in units, financial planning in dollars, and has a simulation capa-
bility to answer "what if" questions. It is made up of a variety of functions, each linked together: Business Planning, Production Planning, Master Production Scheduling, Material Requirements Planning, Capacity Requirements Planning and the execution systems for capacity and priority. Outputs from these systems would be integrated with financial reports such as the business plan, purchase commitment report, shipping budget, inventory projections in dollars, etc. Manufacturing resource planning is a direct outgrowth and extension of MRP. Often referred to as MRP II. (cf. closed-loop MRP).

Manufacturing Technology
- The development of new technologies for processes and manufacturing methods.

MAP
- Manufacturing Automation Protocol. A specification for a suite of communications standards for use in manufacturing automation developed under the auspices of the General Motors Corporation. The development of this specification is being taken over by the MAP/TOP Users Group under the auspices of CASA/SME (The Computer and Automated Systems Association of the Society of Manufacturing Engineers).

MAP/EPA
- Part of the EPA architecture, a MAP/EPA node contains both the MAP protocols and the protocols required for communication to Mini-MAP. It can communicate with both Mini-MAP nodes on the same segment and full MAP nodes anywhere in the network.
MAP/TOP Users Group - The United States and Canada’s MAP/TOP Users Group (see CASA/SME and World Federation).

MAPICS - Manufacturing, Accounting and Production Information and Control System. An IBM MRP system that runs on the AS/400 minicomputer and includes 18 modules of applications.

MAP Users Group - An organization comprising manufacturing users seeking a manufacturing-specific implementation of OSI networking technology.

Market Demand - The total need for a product or line of product. Conditions and events in the marketplace which define the potential product volume and product mix. This includes perceived customer need for product features and product volumes, and marketplace changes due to political, social, and technology changes.

Market Expectations - Assessment of the market situation of the products.

Market Growth - Predictability of market swings that affect long-term expectations and decision making.

Market Plan - A high-level plan of tasks and schedules to meet marketing and sales objectives.

Market Research - Study of market conditions and future behavior based on market trends, marketing data and customer preferences [CAMI].

Master Plan - A document presenting a critical look at current plant capabilities.
and practice (AS-IS); at the desired future state of the plant under the proposed enterprise system (TO-BE); and a proposal to manage the transition between them.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Schedule</td>
<td>For selected items, a statement of what the company expects to manufacture. It is the anticipated build schedule for those selected items assigned to the master scheduler. The master scheduler maintains this schedule which, in turn, drives MRP. It is what the company plans to produce expressed in specific configurations, quantities and dates [APIC].</td>
</tr>
<tr>
<td>Master Slave</td>
<td>A mode of operation where one data station (the master) controls the network access of one or more data stations (the slaves).</td>
</tr>
<tr>
<td>Material</td>
<td>Any commodity used directly or indirectly in producing a product, e.g., raw materials, component parts, subassemblies, and supplies. Materials are purchased items or raw materials that are converted via the manufacturing process into components and/or products [APIC]. This can also be consumable material used to support production operations.</td>
</tr>
<tr>
<td>Material Control</td>
<td>The function of maintaining a constantly available supply of raw materials, purchased parts and supplies that are required for the production of products.</td>
</tr>
<tr>
<td>Material Flow</td>
<td>The progressive movement of material, parts or products toward the completion of a production process between work stations, storage ar-</td>
</tr>
</tbody>
</table>
eas, machines, departments and the like.

Materials Management - A term to describe the grouping of management functions related to the complete cycle of material flow, from the purchase and internal control of production materials to the planning and control of work-in-process to the warehousing, shipping and distribution of the finished product. Differs from materials control in that the latter term, traditionally, is limited to the internal control of production materials.

Materials Planning - The planning of requirements for components based upon requirements for higher level assemblies. The production schedule is exploded or extended through the use of the bills of materials and the results are netted against inventory.

Material Requisition - An authorization that identifies the type and quantity of materials to be withdrawn from inventory [CMSG].

Material Specification - The establishment of the properties and the detailed requirements for raw or fabricated material [SAMA].

Mathematical Model - A mathematical representation of a process, device, or concept.

Mbit - Million Bits Per Second.

Measures - Calibrations of performance.

Media - The physical interconnection between devices attached to the LAN. Typical LAN media are Twisted Pair, Baseband Coax, Broadband
Message - A collection of one or more sentences and/or command statements to be used as an information exchange between applications or users.

Method - A procedure or technique for performing some significant portion of the operational or software life cycle of the system.

Methodology - A collection of methods based on a common philosophy that fit together in a framework called the system development life cycle.

Milestone - In project management, a scheduled event that is used to measure progress.

MINI-MAP - A subset of MAP protocols extended to provide higher performance for applications whose communications are limited to a single LAN. A Mini-MAP node contains only the lower two layers (Physical and Link) of the MAP protocols. It can only communicate directly with MAP/EPA or MINI-MAP nodes on the same segment.

MIPS - Million Instructions Per Second. An approximate figure to denote a computer's raw processing power. It is often misleading, since it does not necessarily provide a good throughput figure of merit.

MMFS - Manufacturing Messaging Format Standard. The application protocol specified by older versions of MAP to do manufacturing messaging. This protocol has been replaced by MMS.
MMS - Manufacturing Messaging Specification. MMS is one of the application protocols specified by MAP.

Model - A synthetic abstract representation of reality.

Modelling Methods - The formal description of model building techniques. [CIMOSA]

Modelling Systems - Computer languages or software packages which support the formulation of models and their execution (simulation) on computer systems. [CIMOSA]

Modem - Modulator-Demodulator. A device that provides both combining (modulation) and separation (demodulation) of data and carrier, and a physical medium interface, typically used to connect a node to a broadband network.

Modification Requests, Approvals - Requests, approval, basic information for modification of equipment or design of new facilities.

Modular Bill (of Material) - A type of planning bill which is arranged in product modules or options, often used in companies where the product has many optional features, e.g., automobiles.

Modular System - A system design methodology that recognizes that different levels of experience exist in organizations and, thereby, develops the system in such a way so as to provide for segments or modules to be installed at a rate compatible with the users’ ability to implement the system.
Monitoring - Follow up of engineering work to ensure adherence to standards and correct interpretation of design basis.

Move Order - The authorization to move a particular item from one location to another [APIC].

Move Ticket - A document used in dispatching to authorize and/or record the movement of a job from one work center to another. It may also be used to report other information, such as the active quantity or the material storage location [APIC].


MRPII - A method for effective planning of all the resources of a manufacturing company. Ideally, it addresses operational planning in units, financial planning in dollars and has a simulation capability to answer "what if" questions. It is made up of a variety of functions, each linked together: Business Planning, Production Planning, Master Production Scheduling, Material Requirements Planning (MRP), Capacity Requirements Planning and the execution systems for capacity and priority. Outputs from these systems would be integrated with financial reports such as the business plan, purchase commitment report, shipping budget, inventory projections in dollars, MRP is a direct outgrowth and extension of MRP.

Multiplexing - The time-shared scanning of a number of data lines into a single channel. Only one data line is enabled at any instant.
MVS – Multiple Virtual Storage. IBM’s flagship operating system. Essentially all device support, software functions, time-sharing aids and reliability improvements ever produced by IBM are available with MVS/XA.

N

Natural Language – A person’s native tongue. Natural language systems attempt to make computers capable of processing language the way people normally speak. [DEC]

NBS – National Bureau of Standards, now NIST (National Institute for Standards and Technology) (see Standards Organizations).

NCMRP – Net Change MRP. An approach via which the material requirements plan is continually retained in the computer. Whenever there is change in requirements, open order to inventory status or engineering usage, a partial explosion is made only for those parts affected by the change. NC systems may be continually and totally transaction-oriented, or done in a periodic (often daily) batch.

Net Requirements – In MRP, the net requirements for a part or an assembly are derived by netting the gross requirements against inventory on hand and the scheduled receipts (released orders). Net requirements, lot sized and offset for lead time, become planned orders [APIC].

Network Management – The facility by which network com-
munication and devices are monitored and controlled.

Networking - Consists of software/hardware in combination systematically linking a number of devices (computers, workstations, printers) into a network (system) for the purpose of sharing resources.

New Product Information - Information about new products or concepts about new products.

Nonconformance Report - An inspection report on an item that describes the item's failure to meet its specification, drawing or quality requirements [SAMA].

Object - The expert systems equivalent of a record. In a case frame-knowledge representation it is a case frame. In a semantic net-knowledge representation it is a node.

Object-Oriented Programming (OOP) - A method of programming characterized by the identification of classes of objects closely linked with the methods (functions) with which they are associated, thus leading to information hiding. It also includes ideas of inheritance of attributes and methods.

Objective - 1. A desired end result, condition or goal which forms a basis for managerial decision-making. 2. General directional statements.

Objectives - Throughput, yield, rates, quality.

Objectives Constraints - Basic information and limits of new projects, project adjustments,
cost control adjustments.

Occurrence - A single instance of an entity type, relationship type, or attribute type.

OCR - Optical Character Recognition. A method of collecting data involving the optical scanning of hand printed or special character fonts. If handwritten, the information must adhere to re-defined rules of size, format and location on the form.

Octet - A group of eight bits treated as a unit (see Byte).

ODA - Office Document Architecture. An international standard for the interchange of documents that may contain text, graphics, image and data material.

ODBMS - Object-Oriented DBMS. A Database Management System that takes the concepts of object-oriented programming and applies them to the management of persistent objects on behalf of multiple users, with capabilities for security, integrity, recovery and contention management, while also providing acceptable performance.

ODIF - Office Document Interchange Format. The format of the data stream used to interchange documents in accordance with ODA.

OOP - (see Object-Oriented Programming)

Open Loop System - A control system which has no means of comparing the output with the input; i.e., there is no feedback.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open System</td>
<td>A system that obeys public standards in its communication with other systems and/or between layers.</td>
</tr>
<tr>
<td>Operating Capacity</td>
<td>The capacity or throughput of a particular operation.</td>
</tr>
<tr>
<td>Operating Conditions</td>
<td>Calculated optimum process operating conditions and targets.</td>
</tr>
<tr>
<td>Operating Costs</td>
<td>Total operating costs = maintenance + engineering + operation + RM + energy + looser costs.</td>
</tr>
<tr>
<td>Operating System</td>
<td>1. Software which controls the execution of computer programs and which may provide scheduling, debugging, input/output control, accounting compilation, storage assignment, data management and related services. 2. The master control program of a computer which controls all hardware activity.</td>
</tr>
<tr>
<td>Operator</td>
<td>1. In the description of a process, that which indicates the action to be performed on operands. 2. A person who operates a machine.</td>
</tr>
<tr>
<td>Opportunity Cost</td>
<td>The return on capital that could have resulted had the capital been used for some purpose other than its present use. Sometimes refers to the best alternative use of the capital; at other times to the average return from a feasible alternative.</td>
</tr>
<tr>
<td>Optimization</td>
<td>A method by which a process is continually adjusted to the best obtainable set of operating conditions.</td>
</tr>
</tbody>
</table>
Order Information - Information on accepted order (confirmation, due date, changes ...).

Order Inquiries - Request for information on product or formal purchase order.

Order Traceability - A production order that travels with the job and may include a route, blueprint, material requisition, move tickets, time tickets, etc. [APIC]. This is also used to collect data.

Organization - 1. The classification or groupings of the activities of an enterprise for the purpose of administrating them. Division of work to be done into defined tasks along with the assignment of these tasks to individuals or groups of individuals qualified for their efficient accomplishment. 2. Determining the necessary activities and positions within an enterprise, department or group, arranging them into the best functional relationships, clearly defining the authority, responsibilities and duties of each and assigning them to individuals so that the available effort can be effectively and systematically applied and coordinated.

Originator - An employee who initiates or originates a form or action.

OSI - Open System Interconnect. A logical structure for network operations using seven layers as defined by the ISO. It defines network protocol standards to enable any OS, compatible computer or device to communicate with any other
OSI compliant computer or device for information exchange.

OSI Reference Model - A seven layered model of communications networks defined by ISO. The seven layers are:

Layer 7 - Application: provides the interface for the application to access the OSI environment.

Layer 6 - Presentation: provides for data conversion to preserve the meaning of the data.

Layer 5 - Session: provides user-to-user connections.

Layer 4 - Transport: provides end-to-end reliability.

Layer 3 - Network: provides routing of data through the network.

Layer 2 - Data Link: provides link access control and reliability.

Layer 1 - Physical: provides an interface to the physical medium.

Overhead Costs - Costs incurred in the operation of a business which cannot be directly related to individual products or services. These costs, such as light, heat, supervision and maintenance, are grouped in several pools (department overhead, factory overhead, general overhead) and are distributed to units of product or service by some standard method, such as direct labor hours, direct labor dollars, direct materials dollars [APIC].
Packaging Specification - A specification establishing the method and detailed requirements for packaging, handling, preservation, storage, or shipment of a part, material, product or system [SAMA].

Parameter - 1. A variable that is given a constant value for a specified application. 2. A variable that controls the effect and usage of a command. 3. Alterable values that control the effect and usage of a graphics command. 4. A constant whose values determine the operation or characteristics of a system. In $y = ax_1 - bx_2 + c$; $a$, $b$, and $c$ are the parameters of a family of parabolas. 5. A variable, $t$, such that each variable of a related system of variables may be expressed as a function of $t$.

Part - A material item that has been purchased or fabricated from raw material and is normally not part of an assembly [CMSG].

Particular Architecture - The structure and contents of a specific enterprise and the guidelines of its development program.

Parts for Rework - Nonconforming material that is dispositioned to be reworked.

Parts Order Request - Purchase order request for spare parts replenishment.

Parts Replenishments - Supply of reordered parts to replenish inventory.

Parts Request - Request to spare parts warehouse
for parts by work order.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts Specification</td>
<td>A specification which establishes the design and detailed requirements for components that are joined together and cannot be taken apart without destroying the function of the resulting assembly.</td>
</tr>
<tr>
<td>Part Supplies</td>
<td>Delivery of parts to maintenance crews.</td>
</tr>
<tr>
<td>Pattern</td>
<td>The description of something for which a system should search either in a knowledge base or rule base. [DEC]</td>
</tr>
<tr>
<td>Pattern Matching</td>
<td>A process performed by an expert system during a search through its knowledge base. [DEC]</td>
</tr>
<tr>
<td>Payment Authorization</td>
<td>Payment on authorization for payment for all purchased goods, services, and capital equipment. It also includes salaries, wages, and benefits to the employees, and taxes, social security payments to the government, and all other employee withholding to the appropriate organization.</td>
</tr>
<tr>
<td>PCA</td>
<td>Process Communications Architecture. An architecture for a three layer (Physical, Data Link, and Application) open communications system being developed by ISA SP72. It can provide communications functions that are needed in control and automation applications. PCA uses OSI protocols and provides a transparent application interface to 7-layer MAP networks.</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit. Each of the seven OSI layers accepts data SDUs</td>
</tr>
</tbody>
</table>
(SubData Unit) from the layer above, adds its own header PCI (Protocol Control Information) and passes the data to the layer below as a PDU. Conversely, each of the layers also accepts data from the layer below, strips off its header, and passes it up to the layer above.

Peoples’ Time and Effort - The time spent by people (employee or contractor) for which compensation is required. It is converted to requests for payments based on the contractual obligation and compliance.

Perceptual Skills - Degree to which AMT jobs demand vigilance, concentration, attention, and judgment.

Performance and Costs - Operating performance and cost reporting, rates, utilization, yield, quality ...

Performance and Reporting Data - Plant operating and production data as developed to satisfy historical records, customer requests, plant test operations, costing needs, plant performance results, etc.

Performance Feedback - Feedback on plant equipment and process performance.

Performance Gaps - Situations in which the organization’s recent performance fails to meet expectations.

Performance Specification - A specification that describes what is to be accomplished by the equipment, but does not describe how the system is to be designed. It specifies the development goal
Performed Work - Time reports per work order.


Pick List - A document that lists the material to be picked for manufacturing or shipping orders. Picking is the process of withdrawing from stock the components to make products or the finished goods to be shipped to the customer. [APIC]

PID - Proportional-Integral-Derivative Control. A popular control strategy that produces output control action proportional to the sum of the input, plus the integral of the input, plus the rate of change of the input.

PIEEE 1003 - The PIEEE 1003 activity represents different committees that are crafting sets of specifications for Posix. The IEEE 1003.4 group is focusing on real-time issues that all facilitate the use of Posix-compliant hardware on the factory floor.

Planned Order - A suggested order quantity and due date created by MRP processing when it encounters net requirements. Planned orders only exist within the computer and can be changed or deleted before they are released. Planned orders at one level are exploded into gross requirements for the next level [APIC].

Planning - The procedure for determining a course of action intended to ac-
complish a desired result.

<table>
<thead>
<tr>
<th>Planning Horizon</th>
<th>In an MRP system, the planning horizon is the span of time from the current to some future date for which material lans are generated. This must cover at least the cumulative purchasing and manufacturing lead time, and usually is quite a bit longer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant Agreements With Suppliers</td>
<td>Written agreements in place that specify quality and delivery time of materials provided by suppliers.</td>
</tr>
<tr>
<td>Plant Works</td>
<td>A series of packages that enable users to create applications for the production environment. Its object-oriented programming and 4GL techniques let non-programmers use it.</td>
</tr>
<tr>
<td>Plant's Degree of Autonomy</td>
<td>Degree to which plant management (as opposed to corporate management) controls decisions involving in-house resources such as personnel, capital expenditures, R &amp; D, and so on.</td>
</tr>
<tr>
<td>PMS</td>
<td>Process Messaging Service (see ISA SP72).</td>
</tr>
<tr>
<td>Policy</td>
<td>A definite course or method of action selected from among alternatives and in light of given conditions to guide and determine present and future decisions.</td>
</tr>
<tr>
<td>Policy Implementors</td>
<td>1. Entities which execute the established manufacturing policy. They may be humans, computer systems or other devices depending upon the capabilities needed. 2. Policy implementors comprise the information management and automa-</td>
</tr>
</tbody>
</table>
tion system configuration. 3. Policy implementors comprise those (agents whose decisions are effectively computable).

Policy Makers - Are external influences that formulate the established manufacturing policy. Because of the innovation necessary, they will be human beings for the foreseeable future.

POMS - Process Operations Management System. An IBM- and customer-sponsored software initiative to link MRPII applications running on AS/400 minicomputers and shopfloor data collection systems running on PS/2 computers. It is not an IBM product, though IBM funded its development.

POSIX - Portable Operating System for Computer Environments. A UNIX-based standard under development by the IEEE. In 1988, the committee officially released the first installment of the standard (1003.1), which presents a core set of system cells. POSIX has been adopted as a Federation Information Processing Standard and is included in OSFs basic specifications.

Predicate - A function that returns a truth value. Predicates are used to select among conditional alternatives.

Preferred Job Challenge - Extent to which some or all members of the work force want challenge in their jobs.

Predic-

tive/Preventive - Maintenance specifically intended to prevent faults from occurring
Maintenance during subsequent operation.

Price - Product cost and marketing cost.

Principle of Autonomy - Leads to a modular system with high cohesion and low coupling. Autonomy means that individual units are as independent in action as overall integration can permit.

Principle of Locality - Leads to distributed processing and time-phased decomposition. Locality means that units in the same geographical region tend to work closely together. (They were placed in the same locality during design for this purpose.)

Priority - In a general sense, priority refers to the relative importance of jobs or tasks and determines in which sequence the competing jobs for one resource will be performed [APIC].

Problems - Tasks resulting from unsatisfactory performance or the environment.

Procedure Policy - These are definitions of approved methods of how to accomplish tasks [APIC].

Procedure Rules - Represent the Enterprise operation as they dynamically control sets of Enterprise activities.

Process - Functions conducted by the enterprise to accomplish its purpose. Processes satisfy the principle of locality and the principle of autonomy (Purdue model). A process can be decomposed into infinite levels of subprocesses.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Analysis</td>
<td>Analytical examination of a process for the purpose of documenting and understanding the phenomena which occur in/within the process.</td>
</tr>
<tr>
<td>Process Control</td>
<td>The regulation of variables that influence and/or control the conduct of a process so that a specified quality and quantity of product is obtained. Pertaining to systems whose purpose is to provide automation of continuous operations. This is contrasted with numerical control, which provides automation of discrete operations.</td>
</tr>
<tr>
<td>Process Data</td>
<td>Data collected from a process.</td>
</tr>
<tr>
<td>Process Data Requirements</td>
<td>A specification of the data that must be collected to support the information needed for other functions, such as process planning.</td>
</tr>
<tr>
<td>Process Plan</td>
<td>A detailed plan for the production of a piece part or assembly. It includes a sequence of steps to be executed according to the instruction in each step and consistent with the controls indicated in the instructions [CAMI].</td>
</tr>
<tr>
<td>Process Specification</td>
<td>A specification which establishes material properties and detailed process control requirements for materials or items that require specific process operations [SAMA].</td>
</tr>
<tr>
<td>Process Status</td>
<td>Information concerning the current well being of the process; in control, alarms (out of control process), etc.</td>
</tr>
<tr>
<td>Producibility Requirements</td>
<td>An evaluation of the product's design with respect to how the de-</td>
</tr>
</tbody>
</table>
sign can be changed to improve the producibility of the product.

**Product Allocation** - Identifies the planned systems and/or the geographical location of each configuration item and the quantity requirements at each location or in each system [SAMA].

**Product Cost** - Total manufacturing cost of product excluding sales, marketing and company overhead.

**Product Definition Data** - Data that describes the product definition, such as CAD data.

**Product Family** - A group of products having a common classification criteria. Design engineering may classify items by function, size, shape or material in order to retrieve all items having common characteristics when required for a specific design purpose. This avoids duplication of design, routings, items, stock accounting, etc. The sales department may classify items by product groups according to potential users, function, size, etc. [CMSG].

**Product Information** - Product related sales information (price, availability, documentation ...).

**Production** - 1. The manufacturing of goods. 2. The act of changing the shape, composition, or combination of materials, parts, or subassemblies to increase their value. 3. The quantity of goods produced.

**Production Baseline** - The resulting state of product after preliminary design, detail design, and qualification and proto-
type testing are complete.

Production Capacity - The highest, sustainable output rate which can be achieved with the current product specifications, product mix, worker effort, plant, and equipment.

Production Limits - Actual rate of production capacity by unit, per product.

Production Order - A document or group of documents conveying authority for the manufacture or assembly of specified products or components in specified quantities [APIC].

Production Order Planned - A suggested order quantity and due date created by MRP processing, when it encounters net requirements. Planned orders only exist within the computer and can be changed or deleted before they are released. Planned orders at one level are exploded into gross requirements for the next level [APIC].

Production Order Reporting - A report of the data collected during the manufacture of the product.

Production Orders - Forecasted and accepted orders.

Production Performance - Throughput, yield, rates, quality.

Production Plan - A plan for setting the overall level of manufacturing output. Its prime purpose is to establish production rates that will achieve management's goals of lowering/raising inventories or backlogs while keeping the production
force relatively stable. It is usually stated in broad terms such as product families or end products [APIC].

Production Planning - 1. The systematic scheduling of workers, materials, and machines by using lead times, time standards, delivery dates, work loads, and similar data for the purpose of producing products efficiently and economically and meeting desired deliver dates. 2. Routing and scheduling.

Production Rate - The rate at which a particular operation produces output.

Production Schedule - A plan which authorizes the factory to manufacture a certain quantity of a specific item. Usually initiated by the production planning department.

Production Variables - Validated measurement related to production performance weights, rates, analysis, levels.

Productivity Shell - A combination of underlying knowledge representations and user interfaces that is highly domain-specific. It includes objects, decision making processes and interface representations that are user, rather than artificial intelligence dependent.

Product Proposal - A proposal for developing a new product.

Products and Waste Materials and Utilities - This output includes finished product, intermediate products including subassemblies and modules, as well as process generated utilities (i.e., steam, electric). It further includes all waste ma-
terials produced by the process including consumables, scrap material, heated water, toxic waste and solvents.

Product Specification
- A specification describing the requirements that a product must meet.

Product Strategy
- A strategy on how to produce or market a product. It includes product mix, etc.

Product Structure
- The definition of the way components go into a product during its manufacture. A typical product structure would show the relation of one component to another; i.e., the relationship of the raw material being converted into fabricated components, components being put together into subassemblies, subassemblies going into assemblies, etc. [APIC].

Product Structure Lead-Time
- A product structure that describes the lead time needed for raw material, time to manufacture parts, assemble parts, etc.

Product Structure Make-Buy
- A product structure that specifies which components will be made and which will be bought.

Product Structure Material
- A product structure describing the raw material needed to produce a part, and the material needed to support the production operation.

Product Structure Resource
- A product structure describing the resources needed to produce each component.

Product Technology
- Technologies that are introduced
in the development of new or existing products.

Program
- 1. A plan of a course of action, activities, duties. 2. A logical sequence of coded instructions specifying the operations to be performed by a computer in solving a problem or in processing data.

Programmable Logic Controller
- A specialized industrial computer used to programmatically control production and process operations by interfacing software control strategies to input/output devices.

Progress Costs
- Progress cost reporting of running projects.

Project Costs
- Project costs, estimates, projected cost.

Project Planning
- Project schedule work planning.

Protocol
- A formal definition (semantic or syntax) that describes how data is to be formatted for communication between a data source and a data sink.

Prototype
- 1. In engineering, a prototype is a full-scale model and/or the functional form of a new system or subsystem. A prototype does not have to be the complete system but only the part of interest. A prototype is sometimes referred to as a model. 2. In software engineering, a software prototype would be a computer program that implements some part of the system requirements. This prototype can be used to assist in defining requirements or, in this case, to assist in evaluating alternatives. Examples
include determining how to obtain the required accuracy in realize performance issues, as well as user interface suitability. The process of building the prototype will also expose and eliminate a number of the ambiguities, inconsistencies, blind spots, and misunderstandings incorporated in the specifications.

Prototyping

- 1. A software requirements analysis strategy that includes the idea of building a prototype of a system to give early visibility of the actual functioning and operational behavior to the designer and the customers. [STARTS Guide 1987] 2. In software engineering, construction of a “quick and dirty” system, beginning with the very minimum of specifications that have been prepared. The purpose of this system is to show the users what they are asking for and to give them a working knowledge of the result that can be achieved by the system they have defined. [Scharer 1981]

PROWAY

- A standard for a process control highway based on IEEE 802.4 token bus immediate acknowledged MAC (Media Access Control), a physical layer utilizing a phase-contiguous signaling technique. Developed by ISA SP72.

Purchased Energy, Material and Supplies

- The tangible items that are brought into the manufacturing entity to facilitate the manufacturing process. They are converted to materials leaving the entity in the form of produced or waste material.
Purchasing Lead Time - The total time required to obtain a purchased item. Included here are order preparation, release time, vendor lead time, transportation time, receiving, inspection and put away time [APIC].

QA - Quality Assurance.

QA Approvals - Classification of QA results after testing.

QA Results - Classification and test results for finished product.

QA Standards - Limits, specifications and standards for in-process quality control.

Qualification Results - Results and conclusions of tests, examinations, documents and processes showing that an item meets or exceeds the expected environmental stresses without failure or malfunction [SAMA].

Quality Audit - An independent review conducted to compare some aspect of quality performance against a standard for that performance. Audits span the entire spectrum of the quality function. Quality audits are used mainly by the company to evaluate its own quality activities or the quality activities of vendors, licensees, etc. Audits can also be required by regulatory agencies to judge different organizations [JURA].

Quality Control - The procedure of establishing acceptable limits of variation in size, weight, finish, and so forth
for products or services and of maintaining the resulting goods or services within these limits.

Quality Plan - Planning performed to meet quality objectives. The quality plan is a detailed breakdown and assignment of responsibilities of activities and deeds that must be performed through each phase of the product life cycle (cradle to grave) [JURA].

Quantities Locations - Updates to inventory of locations, quantity, specification of RM.

Quantities Movements - Actual quantities and status of transfer.

Queue Time - The amount of time a job waits at a work center before setup or work is performed on the job. Queue time is one element of total manufacturing lead time. Increases in queue time result in direct increases to manufacturing lead time.

R

Rapid Entry into New Markets - Extent to which the plant can take advantage of its new production capacity with AMT to enter new markets quickly.

Real Time - 1. Pertaining to the actual time during which a physical process transpires. 2. Pertaining to computations performed while the related physical process is taking place so that results of the computation can be used in guiding the physical process.

Real Time System - Always provides responses (both
periodic, time initiated responses and input or interrupt driven responses) within a specified window of time. The time is determined by the time constant of the dynamic process. Example time constants for external processes are milliseconds for machining or electric power systems, seconds for flow processes, minutes for thermochemical compositions and weeks for social/economic processes.

**Receipt Bondroom** - The physical acceptance of an item into a stocking or storage location. This also refers to the transaction reporting of this activity [APIC].

**Receipt Finished Goods** - Physical acceptance of an item into a stocking or storage location. This also refers to the transaction reporting of this activity [APIC]. In this context, this is a receipt for accepting finished goods from WIP (Work in Process).

**Receipt Stockroom** - The physical acceptance of an item into a Stocking or storage location. This also refers to the transaction reporting of this activity [APIC].

**Receipt WIP** - This is the physical acceptance of an item into a stocking or storage location. This also refers to the transaction reporting of this activity [APIC].

**Recruiting Pool** - The group of workers from which workers will be chosen.

**Reference Architecture** - 1. A generic framework to guide all the disciplines involved in
every step of an enterprise development program throughout its whole life cycle. [PERA] 2. A non-industry specific reference base from which to construct models of a particular enterprise, which contains both Building Blocks and Guidelines. [CIMOSA]

Reference Model - A standard definitive document or conceptual representation of a system or process.

Reference Model CIM - A reference for computer integrated manufacturing.

Relationship - 1. An interaction between entities, usually expressed as verbs in English. 2. A named business association between occurrences of one or more entity types which provides some relevant information value.

Relationship Type - Is a classification of relationships based on certain criteria. The numerical relationship between two entities is called the cardinality.

Relay Ladder Logic - A control language for programming Programmable Logic Controllers.

Release for Shipment - Quality, location, destination of product to be shipped.

Released Order - Within MRP, open production orders and open supply orders are considered as released, are assumed to be available on their due dates, and will be treated as part of the available inventory during the netting process for the time period in question [APIC].

Reliability - The actual degree of dependability
with which the equipment performs (note: actual versus hoped-for).

**Repeater**
- A device that amplifies or regenerates data signals in order to extend the distance between data stations.

**Reporting**
- Customer credit limits, gross margin reporting, overhead costs reports.

**Requests for Services, Capital and Goods**
- Plant requests for raw materials, supplies, and energy versus established vendor contracts; capital appropriation requests versus plant expansion and renovation needs; requests for external services such as shipping, security, janitorial, construction, etc., as needed.

**Resource Build Schedule**
- A schedule of the activities needed to make, install or procure production resources, such as equipment, facilities and tools (excluding material), which are needed to support production.

**Resource Definition**
- The layout, requirements, configuration, design and capacity of resources needed to perform a particular function or operation.

**Resource Plan**
- Long-range resource planning is based on the production plan, on the plan for the long-term capacity needs out to the time period necessary to acquire gross capacity additions, such as a major factory expansion [APIC].

**Resource Requirements Planning**
- The process of converting the production plan and/or the master production schedule into the impact on key resources, such as man
hours, machine hours, storage, standard cost dollars, shipping dollars and inventory levels. Product load profiles or bills of resources could be used to accomplish this. The purpose of this is to evaluate the plan prior to attempting to implement. It is sometimes referred to as a rough-cut check on capacity. Capacity requirements planning is a detailed review of capacity requirements.

Resource View - Describes an enterprise's resources and their relations to the functional structure, control structure and organizational (responsibilities) structure. [CIMOSA]

Response Time - The total time necessary to send a message and receive a response back at the sender exclusive of application processing time.

Return-to-Vendor - Material that has been rejected by the buyer's inspection department and is awaiting shipment back to the Supplier for repair or replacement [APIC].

Rework - 1. The process of correcting a defect or deficiency in a product or part. 2. Units of product requiring correction.

Rigidity - The inability of the production process to reroute around equipment if breakdown occurs.

RISC - Reduced Instruction Set Computer. A computer architecture using a small set of instructions at the hardware level. RISC enables a complex processor to be built from
very high-speed, simple components. IBM invented the RISC concept.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>Raw Materials.</td>
</tr>
<tr>
<td>RM Energy Utilization</td>
<td>Total of raw materials, parts, tools and incoming energy, consumed or transferred on hourly, daily, monthly basis.</td>
</tr>
<tr>
<td>RM Order Request</td>
<td>Request to order raw materials, quality, type specifications, special, requirements ...</td>
</tr>
<tr>
<td>Route Sheet</td>
<td>A document that specifies the operations on a part and the sequence of these operations, with alternate operations and routings where feasible. It also can include material requirements (kind and quantity), machining tolerances (tool, jig and fixture requirements and the time allowances for each operation [APIC]).</td>
</tr>
<tr>
<td>Router</td>
<td>A network device that interconnects two computer networks that have the same network architecture. A router requires OSI Level 1, 2 and 3 protocols (see Bridge and Gateway).</td>
</tr>
<tr>
<td>Routes</td>
<td>Routing and commands to initiate a physical transfer of materials.</td>
</tr>
<tr>
<td>Routing</td>
<td>A document describing the manufacture of a particular item and specifying the sequence of operations, transportation, storage and inspections to be used. It usually includes the standard times applicable, the machines, equipment, tools, operations, labor requirements and materials that are required for each operation.</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>RPC</td>
<td>Remote Procedure Call. A mechanism that extends the notion of local procedure call to a fully distributed computing environment.</td>
</tr>
<tr>
<td>RS511</td>
<td>A messaging standard, also known as MMS, under development in EIA for communication between factory floor devices. It uses ASN.1 for data encoding (see ASN.1 and MMS).</td>
</tr>
<tr>
<td>SAA</td>
<td>Systems Application Architecture. IBM announced SAA as a collection of selected software interfaces, conventions and protocols that were published in 1987. SAA will be the framework for development of consistent applications across future offerings of the major IBM computing environments: MVS, VM, OS/400 and OS/2.</td>
</tr>
<tr>
<td>Safety</td>
<td>Likelihood that use of the equipment will not cause worker accidents.</td>
</tr>
<tr>
<td>Safety Stock</td>
<td>1. In general, this is a quantity of stock planned to be in inventory to protect against fluctuations in demand and/or supply. 2. In the context of master scheduling, safety stock can refer to additional inventory and/or capacity planned as protection against overplanning or as a market hedge [APIC].</td>
</tr>
<tr>
<td>SAMA</td>
<td>Scientific Apparatus Manufacturer Association. An organization of companies which produce control equipment sensors, etc.</td>
</tr>
</tbody>
</table>
Sample Plan - Sampling inspection conducted to learn about the quality of a product lot by inspecting a small number of units of the product (the sample) drawn from that lot. The prime purpose is to classify lots as acceptable or unacceptable. A sampling plan is a set of instructions on how to conduct the sampling inspection [JURA].

Samples - Samples of materials sent to QA for testing.

SAP - Service Access Point. The connection point between a protocol in one OSI layer and a protocol in the layer above. SAPs provide a mechanism by which a message can be routed through the appropriate protocol as it is passed up through the OSI layers.

SC - Standing Committee.

SCADA - Supervisory Control and Data Acquisition. A software package that obtains data from production environment activities and uses it for a variety of applications. These applications include simple control, monitoring, trending and data collation for upstream use.

Schedule Audit - Audits of quality plans are usually conducted on scheduled basis. The schedule provisions for new products or projects (during the original planning or after launching the product), for stable product lines (audits are performed on a regular cycle), and for particular quality problems, such as product, performance, customer feedback and audit reports [JURA].
Schedule of Operations - The actual assignment of starting and/or completion dates to operations or groups of operations to show when these must be done if the manufacturing order is to be completed on time. These dates are used in the dispatching operation [APIC].

Schedule Maintenance - Maintenance schedules at a factory-wide level, for example, if power is going to be shut down for the entire factory or the plant will be shut down for two weeks, etc.

Scheduling - The process of setting operation start dates for jobs to allow them to be completed by their due date.

Schema - In data processing, a set of relationships among data elements in a complex file structure.

SCN - Specification Change Notice. A document that describes changes to an approved specification. The SCN is incorporated as part of the specification after customer approval [SAMA].

Search - The process of trying different actions in a system until a sequence of actions is discovered that will achieve a goal state. [DEC]

Selection Criteria - The degree to which work force choice, machine skills, human relations skills, and individual interest are used as criteria in selecting workers.

Semantics - A branch of linguistics that is concerned with the nature, struc-
<table>
<thead>
<tr>
<th><strong>Sensor</strong></th>
<th>A device that monitors a process variable (temperature, pressure, flow, position) and transmits a voltage or current proportional to that measured variable to a control system.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Serial Identification</strong></td>
<td>A serial number or block number used with a part number to denote each unit (a lot has a quantity of several units) in a family of similar items. It provides for affectivity identification of design changes [SAMA].</td>
</tr>
<tr>
<td><strong>Service Level</strong></td>
<td>A measure of delivery performance in the form of a percentage of the number of items or dollars on actual customer orders that were shipped on schedule for a specific time period compared to the total that were planned to be shipped for that time period [APIC]. This can also include back orders, items shipped, orders past due and age by week. In manufacturing, the service level can be measured by shortages, missed promise dates and excess.</td>
</tr>
<tr>
<td><strong>Shipment Confirmation</strong></td>
<td>Signal to product administration that actual shipment has been done.</td>
</tr>
<tr>
<td><strong>Shipments</strong></td>
<td>The net value of actual shipments represents the standard material and outside processing cost of the unit shipped multiplied by the quantity shipped. This is usually the standard material and outside processing costs.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Shipping Costs</td>
<td>Total and actual cost of transport per shipment, per month, etc. ...</td>
</tr>
<tr>
<td>Shipping Documents</td>
<td>Bill of lading, customer clearance ...</td>
</tr>
<tr>
<td>Shrinkage Factor</td>
<td>A percent factor that compensates for expected loss during the manufacturing cycle by either increasing the gross requirements or by reducing the expected completion quantity of planned and open orders. The shrinkage factor affects all uses of the part and its components, and the scrap factor. Relates only to one usage [APIC].</td>
</tr>
<tr>
<td>Simulation</td>
<td>The representation of certain features of the behavior of a physical or abstract system by the behavior of another system.</td>
</tr>
<tr>
<td>SME</td>
<td>Society of Manufacturing Engineers (see CASA/SME).</td>
</tr>
<tr>
<td>SNA</td>
<td>Systems Network Architecture. A layered network architecture developed by IBM. The layers isolate applications from system networks, services, enabling users to write applications independent of the lower networking software layer.</td>
</tr>
<tr>
<td>SNAP</td>
<td>Sub-Network Access Protocol. Provides a mechanism to uniquely identify private protocols above LLC.</td>
</tr>
<tr>
<td>Software</td>
<td>In computer science, data processing, and software engineering, a sequence of instructions suitable for processing by a computer. Processing may include the use of an assembler, a compiler, an interpreter, or a translator to pre-</td>
</tr>
</tbody>
</table>
pare the program for execution, as well as to execute it. [ANSI/IEEE Standard 729-1983]

Software Configuration Item (SCI) - In software system engineering, a software entity that has been established as a configuration item. The software configuration item exists where functional allocations have been made that clearly delineate the separation between equipment functions and software functions and the software has been established as a configuration item.

Software Crisis - The term used since the 1960s to describe the recurring system development problem in which software problems cause the system to be late, over cost, and not responsive to the users' or customers' requirements including functionality performance, usability, and maintainability.

Software Interface - In software engineering, a shared boundary. An interface might occur between two software components, between hardware and software components, or between an operator and a software components, or between an operator and a software system.

Software Maintenance - In software engineering, 1. the performance of activities required to keep a software system operational and responsive after it is accepted and placed into production. [NBS Special Publication 500-106 1983] 2. Modification of a software product after delivery to correct faults, to improve performance or other attributes, or
to adapt the product to a changed environment. [ANSI/IEEE Standard 729-1983]

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort</td>
<td>Material that has been rejected by an inspection sample plan can be dispositioned as either rejected or acceptable if it goes through 100 percent inspection; a disposition that requires sorting inspects all of the material.</td>
</tr>
<tr>
<td>Source Address</td>
<td>The physical (hardware) address of the node that transmitted the frame.</td>
</tr>
<tr>
<td>Source of Primary Training</td>
<td>Use of either vendors, corporate training departments, consultants, local educational institutions, or in-house training programs as the primary source of worker training.</td>
</tr>
<tr>
<td>Spares</td>
<td>Parts used for the repair/maintenance of an assembled product. Typically, they are ordered and shipped at a date later than the shipment of the product itself [APIC].</td>
</tr>
<tr>
<td>Spares Inventory</td>
<td>Inventory of spare parts.</td>
</tr>
<tr>
<td>SPC/SQC</td>
<td>Statistical Quality Control/Statistical Process Control. A set of techniques based on statistical principles and methods used to regulate the quality of products and processes.</td>
</tr>
<tr>
<td>Specialist Requirements</td>
<td>Outlines of needs for specialist assistance.</td>
</tr>
<tr>
<td>Specification</td>
<td>A document that describes the major technical requirements for an item and the procedure for determining that the requirements have</td>
</tr>
</tbody>
</table>
been met [SAMA].

**Specification List**  -  Lists which are used to maintain the current status of specifications for the product. They list all released specifications in consecutive or numerical order, the current status of each specification (change status/completion, etc.) and the title [SAMA].

**Specifications**  -  Technical specifications for new equipment.

**Specification Tree**  -  A drawing showing the indentured relationships among specifications independent of the assembly or installation relationships of the items specified. The tree shows the dependency of specifications on other specifications [SAMA].

**Spent Labor**  -  Total spent labor per work order.

**SQL**  -  Structured Query Language. A relational data language that provides a consistent, English keyword-oriented set of facilities for query, data definition, data manipulation and data control. It is a programmed interface to relational DBMSs. IBM Research introduced SQL as the main external interface to its experimental relational DBMS, System R, which it developed in the 1970s.

**Standard**  -  1. A standard is an approved, documented, and available set of criteria used to determine the adequacy of an action or object. 2. A document that sets forth the standards and procedures to be followed on a given project or by a given organization.
Standard Costs - The normal expected (target) costs of an operation, process or product, including labor, material and overhead charges, computed on the basis of past performance costs, estimates or work measurement [APIC].

Standards - Limits, specifications, standards and testing methods for quality control.

Standard Time - The length of time that should be required to (a) set up a given machine or operation, and (b) run one part/assembly/batch/end product through that operation. This time is used in determining machine requirements and labor requirements. Also, it is frequently used as a basis for incentive payrolls or cost accounting [APIC].

Standards Organizations - Many different national and international organizations are involved in the task of MAP, TOP and LAN standards. Some of the key organizations are:

ANSI - American National Standards Institute. ANSI X3T9.5 is working on high speed (50 to 100 Mbit/second) LAN standards.

CBEMA - Computer and Business Equipment Manufacturers Association. CBEMA committee X3T9.5 is working on high speed (50-100 Mbit) LAN standards.

CCITT - International Consulting Committee on Telephone and Telegraph. CCITT standards important to MAP and TOP are the X.25 family standards.
of standards that are used to gateway MAP or TOP to wide area networks (WANs) and X.409 which provided the basis of ASN.1.

ECMA – European Computer Manufacturers Association. ECMA is also working on LAN standards in cooperation with IEEE 802.


EIA – Electrical Industries Association. Work is currently in progress in EIA on RS511, a messaging standard for use between factory floor applications.

IEC – International Electrical Technical Commission. An IEC standards committee (IEC TC97/WG6) has defined a LAN for use in process control environments (PROWAY) in cooperation with IEEE 802.4 and is working on a Field Bus standard. Also known as EIC.

IEEE – Institute of Electrical and Electronic Engineers. An IEEE standards committee (IEEE 802) is chartered to work on LAN standards for data rates of 1 to 10 Mbit/second and had produced the standard for CSMA/CD (IEEE 802.3) used by TOP and the standard for Token Bus (IEEE 802.4) used by MAP. These standards have also been approved by ISO (DIS 8802/3 and DIS 8802/4).

ISA – Instrument Society of America. The ISA is responsible for the PROWAY standard in the United States. ISA SP50 is working on a field bus standard. ISA is the
American cognizant organization for EIC-developed standards.

ISO – International Organization for Standardization. ISO takes standards submitted by its member national standards bodies, ballots the standards internationally, and approves international standards. The major ISO standards used by MAP are ISO FTAM (DP 8571), ISO Session (IS 8327), ISO Transport (IS 8073) and ISO Internet (DIS 8473).

NBS – National Bureau of Standards (now NIST, National Institute for Standards and Technology). An organization of the United States government that is responsible for the standards used by other government agencies (e.g. FIPS, Federal Information Processing Standards). NBS also provides compliance testing services, and hosts standard development workshops.

SME – Society of Manufacturing Engineers (see CASA/SME).

<table>
<thead>
<tr>
<th>Standards Updates</th>
<th>- Updating of standards, methods and practices to actual need and state-of-the-art.</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the-Art</td>
<td>- In engineering, the current (latest) technology, but not necessarily in use.</td>
</tr>
<tr>
<td>State-of-the-Practice</td>
<td>- In engineering, the current (latest) technology in use.</td>
</tr>
<tr>
<td>Station Management</td>
<td>- The portion of Network Management that applies to the lowest two OSI layers.</td>
</tr>
<tr>
<td>Statistical Quality</td>
<td>- A means of controlling the quality</td>
</tr>
</tbody>
</table>

State-of-the-Art – In engineering, the current (latest) technology, but not necessarily in use.

State-of-the-Practice – In engineering, the current (latest) technology in use.

Station Management – The portion of Network Management that applies to the lowest two OSI layers.

Statistical Quality – A means of controlling the quality.
Control of a product or process by the application of the laws of probability and statistical techniques to the observed characteristics of such product or process.

Status - Actual values of operating variables, flows, pressures, temperatures, etc.

Stop Order - An order used to notify manufacturing or testing to stop work on an item for the C1 because an engineering change is in process. It is used to avoid waste of labor or material on an item that will have to be scrapped or reworked due to the change [SAMA].

Strategy - Basic decision of "what business we are in".

Sublayer - A subdivision of an OSI layer (e.g. the IEEE 802 Standard divides the link layer into the LLC and MAC sublayers).

Subsystem - A collection of logically connected functions that implement a particular function in the system.

Subsystem Specification - A specification which establishes the functional, performance and design criteria for the design, development, testing and production of equipment that performs a major function and is essential to the completeness of a system [SAMA].

Supervision - Supervision and coordination of construction activities.

Supervision Coordination - Supervision of outside contractors.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Order</td>
<td>A time-phased replenishment order from MRP for purchasing material or parts.</td>
</tr>
<tr>
<td>Supply Order Planned</td>
<td>A suggested order quantity and due date created by MRP processing, when it encounters net requirements. Planned orders only exist within the computer and can be changed or deleted before they are released. Planned orders at one level are exploded into gross requirements for the next level.</td>
</tr>
<tr>
<td>Supply Order Status</td>
<td>The status of the supply order includes whether it is on schedule or if it will be overdue.</td>
</tr>
<tr>
<td>Support</td>
<td>Labor and services report from outside contractors.</td>
</tr>
<tr>
<td>Symbolic Processing</td>
<td>A type of processing that primarily uses symbols rather than numeric representations of data. In expert systems, symbols are not restricted to a numeric context, but may represent objects, concepts, and processes.</td>
</tr>
<tr>
<td>Syntax</td>
<td>A branch of linguistics that is concerned with the arrangements of words or elements in a sentence and to their relationships with each other. Syntax is the relationships of word groups, phrases, clauses, and sentences, that is, sentence structure.</td>
</tr>
<tr>
<td>System</td>
<td>An organized collection of personnel, machines, and methods required to accomplish a set of specific functions.</td>
</tr>
<tr>
<td>System Development</td>
<td>The process of selecting and integrating functionally distinct de-</td>
</tr>
</tbody>
</table>
vices, mechanism, and subsystems necessary for optimum performance of the operation.

System Engineering - A formal, phased approach to producing a significant new system or major changes to an existing system. It stresses teamwork among users and technical personnel, a series of major milestones, and thorough documentation to assure compliance with performance and schedule goals.

System Parts List - A tabulation of applications, cumulative quantities and affectivity of all parts, components and assemblies within a system [SAMA].

System Requirements Definition Phase - The portion of system development whose purpose is to investigate a company, or part of a company, in sufficient depth to develop a firm business proposition involving a changed method of operation. It results in a statement of the functional requirements of new systems.

System Specification - The specification which establishes the functional, performance and design criteria for the design, development, testing and production of a complete system. This specification allocates the system into functional entities identified as configuration items [SAMA].

TAGS - Technical Assistance Groups. (See IEEE 802)

Task - A recognized action or set of ac-
tions comprising a specific part of the operations of a functional entity of the production plant in fulfilling the established manufacturing policy of the company. It is the lowest level of functional decomposition of an enterprise that corresponds to the function of a single person or machine at a point in time. A task corresponds to an informational or physical transformation in the enterprise studied.

| TC | - Technical Committee. |
| Technical Feedback | - Feedback on installation and specific equipment performance. |
| Technical Instructions | - Operation instructions data on process and equipment. |
| Test Results | - Report on performed tests, including quality. |
| Time | - Time of spent work of individual, by work order. |
| Time-Phased Decomposition | - Involves reducing the complexity of a system by decomposing the solution into a number of hierarchically arranged modules. Each level of the hierarchy represents: |

(1). A shift in the time domain (lower-level layers are closer to real-time).

(2). A corresponding narrowing of the "scope-of-control" in lower levels.

(3). Each level provides planning (control) input to lower layers and accepts process status from lower layers.
Token Bus - An access procedure where the right to transmit is passed from device to device via a logical ring on a physical bus.

Token Passing - A LAN-access technique in which participating stations circulate a special bit pattern (the token) that grants network access to each station in sequence; it is often used in ring topology networks, which have a single cable or dual cables strung from station to station in the shape of a ring. Bus topologies utilize a single strand of cable to which stations are attached. Tokens are special bit patterns or packets that circulate from node to node when there is no network traffic. Possession of the token gives a node exclusive outgoing access to the network, thus avoiding conflict with other nodes that wish to transmit.

Tools - A means for solving an engineering problem in the performance of engineering tasks. A tool may be manual or automatic, formal or informal. Examples of these tools are from software packages for design, like CAD tools, to word processors.

TOP - Technical and Office Protocol. A development of the CSMA/CD (Carrier Sense Multiple Access with Collision Detection) protocol (also IEEE 802.3) under the auspices of Boeing Computer Services for office and laboratory automation use. This has been combined with MAP and further development will be under the auspices of the MAP/TOP Users Group.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>The surface structure or arrangement of parts of any object.</td>
</tr>
<tr>
<td>TQC</td>
<td>Total Quality Control. A theory of quality whereby the maker of the part has the responsibility for the quality of that part.</td>
</tr>
<tr>
<td>Traceability Identification</td>
<td>An identifier used with a part number to denote each unit (a lot has a quantity of several units) in a family of similar items. It provides for affectivity identification of design changes [SAMA].</td>
</tr>
<tr>
<td>Transfers</td>
<td>Source, destination, route, quantity of material to be transferred.</td>
</tr>
<tr>
<td>Transport Order</td>
<td>Order to a transport company to arrange transportation.</td>
</tr>
<tr>
<td>TTP</td>
<td>Telephone Twisted Pair. A network medium that uses existing telephone wiring. Standards work is in process on a TTP standard for IEEE 802.3 STANDLAN and IEEE 802.5 token ring.</td>
</tr>
<tr>
<td>U</td>
<td></td>
</tr>
<tr>
<td>Unit Cost</td>
<td>The total labor, material and overhead cost for one unit of production [APIC].</td>
</tr>
<tr>
<td>UNIX</td>
<td>A family of operating systems licensed by AT&amp;T that are known for their relative hardware independence and portable applications interface. It is used widely in technical and scientific computing applications.</td>
</tr>
<tr>
<td>Usage</td>
<td>The number of units or dollars of an inventory item consumed over a</td>
</tr>
</tbody>
</table>
period of time [APIC].

Usage Variance - The deviation of the actual consumption of materials as compared to a standard. Standard Costs - Actual Costs becomes Usage Variance [APIC].

Use As Is - Material that has been dispositioned as unacceptable as per the specifications but can be used as is [APIC].

Used Parts - Cost report on used parts for each work order.

Utilization Losses - Utilization and unbalances of raw materials and energy.

Variable Costs - An operating cost that varies directly with the production volume including direct material, direct labor and variable overhead costs (fixed factory overhead is not included in variable costs). For inventory order quantity purposes, unit costs must include both variable and fixed costs to determine the unit costs [APIC].

Vendor Lead Time - The time that normally elapses between the time an order is received by a supplier and shipment of the material.

Vendor Performance Rating - The result of measuring a vendor's performance to contract. Performance ratings usually cover delivery, quality and price (vendor measurement) [APIC].

Virtual Memory - A programming method that allows the operating system to provide
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMS</td>
<td>DEC's VAX operating system.</td>
</tr>
<tr>
<td>Waiver</td>
<td>A written authorization report to accept material that during production or after inspection has been found to depart from the specified requirements, but the material is considered to be suitable for use 'as is' or after repair by an approved method. This allows production to proceed. A deviation permits a change before it occurs and a waiver permits a change after it has occurred [SAMA].</td>
</tr>
<tr>
<td>Waiver Approval</td>
<td>A sign-off or approval of a submitted waiver.</td>
</tr>
<tr>
<td>Waiver Description</td>
<td>May include the reason for the waiver or deviation, the impact on production if approved, the impact on production or the schedule if not approved, and corrective action information.</td>
</tr>
<tr>
<td>WG</td>
<td>Working Group.</td>
</tr>
<tr>
<td>WIMPS</td>
<td>Windows, Icons, Mice, Pointers, Scroll Bars.</td>
</tr>
<tr>
<td>Window</td>
<td>An application software &quot;design concept&quot; that allows several programs to be run and displayed on the screen simultaneously and supports integration of data between applications programs. [DEC]</td>
</tr>
<tr>
<td>Wiring Closet</td>
<td>The room or location where the telecommunication wiring for a</td>
</tr>
</tbody>
</table>
building, or section of building, comes together to be interconnected.

Work Authorization - Documents or information conveying authority to a function to perform specified work per a specified priority or schedule. This can include maintenance of equipment and manufacture of specified parts or products in specific quantities [APIC].

Work Authorization Identification - A unique identifier for each work authorization.

Work In Process (WIP) - Work in process is the products in various stages of completion throughout the plant, including raw material that has been released for initial processing, up to the completed processed material awaiting final inspection and acceptance as a finished product [APIC].

Work Order - Transmittal to maintenance of order to perform work (work description, symptoms, special precautions, clearance procedures ...).

Work Report - Report on performed maintenance time, diagnosis, used parts ...

Work Request - A request for work to be performed that has not been scheduled, such as requesting tools and material to be moved to a workstation. This could include a work cell that is idle and needs more work, or it could be a request for unscheduled maintenance to be performed on a piece of equipment.

Workflow Unpredictability - Degree to which the anticipated inputs, processes, and outputs to
the human-machine system are unpredictable.

**Workstation**
- 1. The assigned location where a worker performs their job. 2. A human-machine interface devices that are composed of coordinated input/output devices which include video displays, keyboards, functional menus and may or may not include off-line storage capabilities. Used for graphics, text, data and retrieval functions.

**World Federation**
- The joining together of the three international regions related to MAP and its promotion and standardization: (1) The Americas (Canadian MAP interest group and U.S. MAP/TOP Users Group) and Western Pacific (Australian MAP Interest Group), (2) Asia (Japan MAP users Group), and (3) Europe (European MAP Users Groups).

**Y**

**Yield**
- The ratio of usable output from a process to the material of value put into the process. Yield is usually expressed as a percentage and may be expressed in terms of total input of a specific raw material.