DESIGN FOR AGILITY

Andrew Kusiak
Intelligent Systems Laboratory
2139 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 - 1527
Tel: 319 - 335 5934    Fax: 319-335 5424
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Outline

• The concept of agile manufacturing
• Design for agility rules
• Design for reconfigurability

WHAT IS "AGILE MANUFACTURING"?

Agility: The measure of a manufacture’s ability to react fast to sudden, unpredictable change in customers demand for its products and services and make a profit (Noaker 1994)

The real issue in agility is reconfiguration, i.e., the ability to assemble the resources needed quickly … (Industrial Week 1993)

In an 'agile' enterprise, products will be built quickly and cheaply for a customer based on detailed date received at the point of sale (Brooke 1993)

Characteristics of Agile Manufacturing

1. Greater product customization - product variety at low unit cost
2. Quick response to changing market requirements
3. Upgradable products - designed for modularity, disassembly, recyclability, and reconfigurability
4. Dynamic reconfiguration of processes and systems - to accommodate swift changes in product designs or the introduction of new products

History of Agility

• Flexible
• Integrated
• JIT
• Lean
• Agile
• e-Manufacturing
Agility and e-Service Marketplace

Parallel concepts

Agility - operations perspective

e-Business - information processing perspective

Design for Agility Rules

Rule 1 Modular System Design

Decompose a complex system into several independent units
Minimize \( \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij} \)

s.t.

\( \sum_{j=1}^{N} x_{ij} = 1, \quad j = 1, \ldots, N \)

\( \sum_{i=1}^{N} x_{ij} = 1, \quad i = 1, \ldots, N \)

\( u_i - u_j + N x_{ij} \leq N - 1, \quad i = 2, \ldots, N, \quad j = 2, \ldots, N, \quad i \neq j \)

\( x_{ij} \in \{0, 1\}, \quad i, j = 1, \ldots, N \)

\( u_i \geq 0, \quad i = 1, \ldots, N \)

**TSP Formulation**

Before decomposition

Relax subtour constraint

After decomposition

Assignment problem

**Example 1**

Cost matrix \([G_{ij}]\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>∞</td>
<td>15</td>
<td>35</td>
<td>22</td>
<td>13</td>
<td>20</td>
<td>17</td>
<td>11</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>∞</td>
<td>39</td>
<td>12</td>
<td>16</td>
<td>26</td>
<td>46</td>
<td>21</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>18</td>
<td>∞</td>
<td>26</td>
<td>35</td>
<td>38</td>
<td>24</td>
<td>28</td>
<td>23</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>50</td>
<td>41</td>
<td>∞</td>
<td>12</td>
<td>24</td>
<td>38</td>
<td>31</td>
<td>50</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>35</td>
<td>23</td>
<td>21</td>
<td>∞</td>
<td>46</td>
<td>10</td>
<td>25</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>41</td>
<td>17</td>
<td>24</td>
<td>50</td>
<td>22</td>
<td>∞</td>
<td>21</td>
<td>26</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>50</td>
<td>14</td>
<td>36</td>
<td>31</td>
<td>21</td>
<td>10</td>
<td>∞</td>
<td>12</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>21</td>
<td>26</td>
<td>38</td>
<td>13</td>
<td>12</td>
<td>17</td>
<td>∞</td>
<td>15</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>40</td>
<td>35</td>
<td>24</td>
<td>10</td>
<td>26</td>
<td>30</td>
<td>20</td>
<td>∞</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>25</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>21</td>
<td>12</td>
<td>12</td>
<td>15</td>
<td>∞</td>
</tr>
</tbody>
</table>

Each product processed on two subsystems

\([P5, P7, P6, P10, P1, P9, P8, P2, P4, P3, P5]\)

\([P1, P7, P6, P10, P1, P9, P8, P2, P4, P3, P5]\)

COST: 180

\([P5, P3, P2, P4, P5]\)

\([P1, P7, P6, P10, P9, P1]\)

COST: 132

**Gain**

- Total setup reduction (132 vs 180)
- Makespan reduction

Two subsystem schedule vs sequential one (Cyclic schedule)

**Any Magic?**

- No precedence constraints
- Each subsystem has to be able to perform the operations assigned
- The reduced cost is due to relaxation of the cycle elimination constraint
**Rule 2: Robust Product Design**

**Design a product with robust scheduling characteristics**

**Design Goal**

Minimize the impact of disruptions on a production schedule due to changes in the product mix and production demand

**Products with a Special Structure - Two Machine Model**

Product family with a linear assembly structure

**Example 2**

Electronic assembly

Mechanical assembly

**Changes in Product Mix Do Not Affect the Production Schedule**

Production of P2 is cancelled

{P3, P2, P1}  \rightarrow  {P3, P1}

Planned schedule  Actual schedule

**Moreover**

Old system

New system

Same character schedule

Similar system layout
Rule 3  Streamlining the Flow of Products

Example 3  Streamlined system
Non-streamlined system

Example 3 (cont.)
Streamlined system
Non-streamlined system

Example 4

Problems with Long Assembly Lines
(1) Difficult to balance
(2) Behavioral problems
(3) NP-completeness of the scheduling problem

Flow shop type flow
Job shop type flow

Rule 4  Design Short Assembly Lines

Reduce the number of stations in an assembly line

Three station system
Two station system

Johnson's algorithm:
Makespan = 40
Heuristic rule:
Makespan = 58
Example 4 (cont.)

Short system

\[ S_1 S_2 \]

Long system

\[ S_1 S_2 S_3 S_4 \]

Johnson's algorithm:
Makespan = 38

Heuristic scheduling rule:
Makespan = 47

Rule 6 Simplify the flow of products

Design products to simplify the flow of products in a multi-product assembly system

Five Types of Product Flows

- Repeat
- Serial
- By-pass
- Backtracking
- Branch/Merge

Cycles ----> Backtracking

Superimposed graph

Assign operations to stations

Eliminate Cycles by Redesign

Redesign

New System Design

Assign operations to stations

Early (Machining-Driven) Product Differentiation Strategy

Parts + Assemblies

Machining

Complex Components
Delayed (Assembly-Driven) Product Differentiation