Nature of Applications

<table>
<thead>
<tr>
<th>“Systems” Software</th>
<th>Business Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Tools, Database, Word Processors, Spreadsheets, etc.)</td>
<td>(Financials, Supply Chain, Human Capital Management, Manufacturing, etc.)</td>
</tr>
<tr>
<td>Building a Car</td>
<td>Building a House</td>
</tr>
<tr>
<td>Low need for adaptation</td>
<td>High need for adaptation and change</td>
</tr>
<tr>
<td>Quality is defined by static operation</td>
<td>Quality is defined by dynamic fit</td>
</tr>
<tr>
<td>Mechanical Systems</td>
<td>Organic Systems</td>
</tr>
<tr>
<td>Decomposition Effects</td>
<td>Network Effects leading to emergent properties</td>
</tr>
<tr>
<td>Engineering Science</td>
<td>Complex Adaptive Systems Theory</td>
</tr>
</tbody>
</table>
Business applications are about constant adaptation

- Causing problems with
  - Quality
    - Inability/Reluctance to make changes
  - 7 X 24 / Upgrades
    - The difficulty and downtime associated with upgrades
  - Training
    - The time and cost of training

Delivering Simplicity, Adaptability, Innovation Through a New Way of Constructing Applications

Traditional Application Construction Pipeline

- One way, low-fidelity communication
- Business User
- Designer / Business Analyst
- Programmer
- 10,000 lines
- 1,000 lines
- Machine code 100,000 lines

Lawson Application Construction Pipeline

- Short, two-way, high-fidelity communication
- Business User
- Designer / Business Analyst
- Live Application Blueprint
- 1,000 lines
- Machine code 100,000 lines
Lawson Pattern Language: Relates To The Domain Expert Instead Of The Programmer

- **Design Language versus Engineering Language**
  - Domain Specific versus General Purpose
  - Specifications based on Patterns
  - Built for adaptation and change

- **Language for the Domain Expert**
  - Uses terminology a domain expert can easily comprehend and use
  - Allows domain expert to fully model the system
  - Critical Innovation: solved the disconnect between the Domain Model and the Implementation. The Domain Model IS the implementation

- **Lawson's internal experience**
  - Less than a week for non-programmer to become proficient
  - Business analyst (with ZERO programming experience) can code complete application
  - Order of magnitude reduction in design instructions
  - Increased Quality and Productivity

Order of Magnitude Reduction in Lines of Code

<table>
<thead>
<tr>
<th></th>
<th>Vendor</th>
<th>Cash req.</th>
<th>Avg. Daily Balance</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>13,262</td>
<td>6,612</td>
<td>3,730</td>
<td>40,320</td>
</tr>
<tr>
<td>To</td>
<td>762</td>
<td>690</td>
<td>254</td>
<td>980</td>
</tr>
</tbody>
</table>

- Major New Module (larger than Requisitions): 5,260 lines
- Built entirely by Domain Experts
Documents

- On http://spec.lawson.com/twiki/bin/view/Spec/PatternLanguage
- LawsonPatterns.doc
- LawsonPatternLanguageV3.rtf
- Example documents
- IDEConceptualDesign.doc

- Also see http://jupiter.lawson.com/Landmark for full documentation set

Books

Metaphysics:

- Chance Love and Logic: Philosophical Essay
  - By Charles Sanders Peirce
- Charles S. Peirce’s Philosophy of Signs: Essays in Comparative Semiotics
  - By Gerald Deledalle
- Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought
  - By Lakoff and Johnson
Books (cont.)

Complex Adaptive Systems Theory:
- Emergence: The Connected Lives of Ants, Brains, Cities and Software
  - By Steven Johnson
- Hidden Order: How Adaptation Builds Complexity
- Emergence: From Chaos to Order
  - By John Holland

Books (cont.)

Pattern Language Concepts:
- The Timeless Way of Building
- A Pattern Language
- The Nature of Order: The Phenomenon of Life
- The Nature of Order: The Process of Creating Life
- Notes on the Synthesis of Form
  - By Christopher Alexander
- Patterns of Software: Tales from the Software Community
  - By Richard Gabriel
Books (cont.)

Information Modeling:
- Semiotics in Information Systems Engineering
  - By Kecheng Liu
- Meta Pattern: Context and Time in Information Models
  - By Peter Wisse

Books (cont.)

Software Patterns:
- Pattern Languages of Program Design Vol: 1-4
  - Edited by Vlissides, Coplien and Kerth
- Analysis Pattern
  - By Gamma, Helm, Johnson and Vlissides
- Design Patterns
  - By Gamma, Helm, Johnson and Vlissides
Thank You

www.lawson.com