Flight Simulation (case study)

• Among the most sophisticated software systems:
  – Highly distributed
  – Hard real-time performance requirements
  – Modifiability: changes in requirements, simulated vehicles & environment
  – Scalability: continuous improvement of the simulation of the real world
Initial stages of the ABC for the flight simulator
Modern flight simulators
Courtesy of the Boeing Company
Highlights

• **Architectural solution:**
  – reference model Fig 8.3

• **Treatment of time:**
  – Periodic Time Management (for real-time)
  – Event-based Time Management (non-RT)

• **Skeletal System:** *Architecture Prototype*
  – Flight Simulator has 6 module types!
  – What are the benefits?
Reference model for flight simulator
Integrability

• Both the data connections and the control connections have been minimized.
  – Integrating another controller has been reduced to a problem that is linear, not exponential.
  – Integrating two subsystems is again reduced to ensuring that the two pass data consistently.
Integrability

- A driving concern in large systems
- Those developed by distributed teams or separate organizations
- Definition: the ease with which separately developed elements, including COTS, can be made to work together to fulfill the system’s requirements.
Applicable Tactics

• Keep interfaces small, simple, and stable.
• Adhere to defined protocols.
• Loose coupling or minimal dependencies between elements.
• Use a component framework.
• Use “versioned” interfaces that allow extensions while permitting existing elements to work under the original constraints.
Pattern: a Structural Model

With the emphasis on:
- Simplicity and similarity of system’s substructures
- Decoupling of data- and control-passing strategies from computation
- Minimizing module types
- A small number of system-wide coordination strategies
- Transparency of design
Details

• The pattern includes an object-oriented design to model the subsystems and controller children of the air vehicle. It marries real-time scheduling to this OOD as a mean of controlling the execution order of the simulation’s subsystems so that fidelity can be guaranteed.
Cost/Benefit Ratio

• The cost is that the subsystem controllers often act purely as data conduits, increasing complexity and performance overhead.

• In practice (for applicable) systems, the benefits far outweigh the cost:
  – Architectural skeleton allows incremental development and easier integration.
  – Every project that has used structural modeling has reported easy, smooth integration.
Pattern: a Structural Model

• With the emphasis on:
  – Simplicity and similarity of system’s substructures
  – Decoupling of data- and control-passing strategies from computation
  – Minimizing module types
  – A small number of system-wide coordination strategies
  – Transparency of design
The Structural Modeling pattern of an air vehicle system processor with focus on the executive
The application module types
How the Structural Modeling Pattern Achieves Its Goals

<table>
<thead>
<tr>
<th>Goal</th>
<th>How Achieved</th>
<th>Tactics Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>• Periodic scheduling strategy using time budgets</td>
<td>Static scheduling</td>
</tr>
<tr>
<td></td>
<td>• Separation of computation from coordination</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Indirect data and control connections</td>
<td>Restrict communication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use intermediary</td>
</tr>
<tr>
<td>Integrability</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modiﬁability</td>
<td>• Few module types</td>
<td>Restrict communication</td>
</tr>
<tr>
<td></td>
<td>• Physically based decomposition</td>
<td>Semantic coherence</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interface stability</td>
</tr>
</tbody>
</table>