Software Reuse for Modeling & Simulation

April 2011

Authors:
Chuck Turnitsa, VMASC
Dr Andreas Tolk, ODU
Emily Andrew, Raytheon Company
M&S and Software Reuse

• Software Reuse, as a concept, has been with Computer Science for a long time

• The Authors’ position for this paper is there are particular concerns of Software Reuse for the M&S community

• This presentation attempts to show those concerns, both the theoretical and the applicable
Software Reuse is Pervasive

- Different stages
  - Development
  - Integration
  - Maintenance

Software Reuse Can Take Several Forms

• For the developer
  – Algorithms
  – Functions
  – Objects

• For the integrator
  – Whole components
  – Whole packages

• For the maintainer
  – Both/Either

  Code level - often used impromptu

  Component level - this requires some degree of certification/trust

  Both - this requires some organizational support
Increasing Benefits of Reuse

## Reuse Maturity Framework

<table>
<thead>
<tr>
<th>Motivation/culture</th>
<th>Planning for reuse</th>
<th>Breadth of reuse involvement</th>
<th>Responsibility for making reuse happen</th>
<th>Process by which reuse is leveraged</th>
<th>Reuse inventory (assets)</th>
<th>Classification activity</th>
<th>Technology support</th>
<th>Metrics</th>
<th>Legal, contractual, account considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial/Chaotic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuse is discouraged</td>
<td>Nonexistent</td>
<td>Individual worker</td>
<td>Individual initiative (personal goal, as time allows)</td>
<td>Development process chaotic; unclear where reuse comes in</td>
<td>Salvage yard (no apparent structure to collection)</td>
<td>Informal, individualized (&quot;in the head&quot;, &quot;in the drawer&quot;)</td>
<td>Personal tools, if any</td>
<td>No metrics on level of reuse, payoff, or cost of reuse</td>
<td>Inhibitor to getting started</td>
</tr>
<tr>
<td>Monitored</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuse is noted indifferentially reinforced, rewarded</td>
<td>Grassroots activity</td>
<td>Work group</td>
<td>Shared initiative</td>
<td>Reuse questions raised at design reviews (after the fact)</td>
<td>Catalog identified language-and-platform-specific parts</td>
<td>Multiple independent schemes for classifying parts</td>
<td>Classification aids and synthesis aids</td>
<td>Number of lines of reused code factored into cost models</td>
<td>Internal accounting scheme for sharing costs, allocating benefits</td>
</tr>
<tr>
<td>Coordinated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuse is incentivized</td>
<td>Targets of opportunity</td>
<td>Department</td>
<td>Dedicated individual</td>
<td>Design emphasis placed on reuse of off-the-shelf parts</td>
<td>Catalog organized along application-specific lines</td>
<td>Single scheme, catalog published periodically</td>
<td>Electronic library separate from development environment</td>
<td>Manual tracking of reuse occurrences of catalog parts</td>
<td>Data rights and compensation issues resolved with customer</td>
</tr>
<tr>
<td>Planned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuse is indoctrinated</td>
<td>Business Opportunity</td>
<td>Division</td>
<td>Dedicated Group</td>
<td>Focus on developing families of products</td>
<td>Catalog includes generic data processing functions</td>
<td>Some domain analysis performed to determine categories</td>
<td>Automated support with integrated development system</td>
<td>Analyses performed to identify expected payoffs from developing reusable parts</td>
<td>Royalty scheme for all suppliers and customers</td>
</tr>
<tr>
<td>Ingrained</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuse is “the way we do business”</td>
<td>Part of a strategic plan</td>
<td>Enterprise</td>
<td>Corporate group (for visibility not control) with division liaisons</td>
<td>All software products generalized for future reuse</td>
<td>Planned activity to acquire or develop missing pieces in catalog</td>
<td>Formal, complete, consistent, timely classification</td>
<td></td>
<td></td>
<td>Software treated as key capital asset</td>
</tr>
</tbody>
</table>

---

All Reuse is Interoperability

• Whether it is old code in new coding projects or old components in new federations

• Knowledge of what is being reused can be:
  – White Box
  – Black Box

**Interoperable** – The property of diverse systems to work together.

Whenever software (either code or components) are used in a new environment, they must be interoperable with that environment.
Special Considerations for M&S

• What separates M&S from other software domains?

• M&S software is based on a model and, to be interoperable, the model must work in the new setting
Reuse Issues for:

M&S SOFTWARE DEVELOPMENT
Levels of Conceptual Interoperability Model (LCIM)

Useful method for determining amount of interoperability

- **Level 6**: Conceptual Interoperability
- **Level 5**: Dynamic Interoperability
- **Level 4**: Pragmatic Interoperability
- **Level 3**: Semantic Interoperability
- **Level 2**: Syntactic Interoperability
- **Level 1**: Technical Interoperability
- **Level 0**: No Interoperability

Network / Connectivity

Simulation / Implementation

Modeling / Abstraction

Composability

Increasing Capability for Interoperation
Interoperability Levels
- Technical – not considered due to no meaningful data exchanged
- Syntactic – data exchange occurs using common syntax

Developer Requirements
- Technical - data capable of being exchanged
- Syntactic - syntax known
Interoperation Capable – Components Intended to Exchange Info

- Interoperability Levels
  - Semantic – syntactic interoperability followed with meaning behind labels
  - Pragmatic – components have a complex relationship and a shared data context

- Developer Requirements
  - Semantic – common syntax and data labeling followed
  - Pragmatic – aware of the context and share the context meaning
Composition Capable – Components Intended to be Harmonized

- Interoperability Levels
  - Dynamic – aware of how context changes
  - Conceptual – model conceptualizations must align

- Developer Requirements
  - Dynamic – know what the context of the other model is when data is exchanged
  - Conceptual – are no differences in conceptualizations to include constraints and assumptions

### Levels of Interoperability

1. **Level 0**
   - No Interoperability

2. **Level 1**
   - Technical Interoperability

3. **Level 2**
   - Syntactic Interoperability

4. **Level 3**
   - Semantic Interoperability

5. **Level 4**
   - Pragmatic Interoperability

6. **Level 5**
   - Dynamic Interoperability

7. **Level 6**
   - Conceptual Interoperability

---

**Increasing Capability for Interoperation**

- **Modeling/Abstraction**
- **Simulation/Implementation**
- **Network/Connectivity**

**V M A S C – ODU**

**Raytheon**

**Network Centric Systems**

**11S-SIW-042**

**2011 Spring SIW**
Reuse Issues for:

M&S SOFTWARE INTEGRATION
Be Aware of Your Software’s Model

- A conceptual model is behind every piece of simulation software – it is the conceptualization of what the software is simulating
  - Sometimes this is written down formally
  - If elusive, then the developer must deduce the apparent conceptual model based on simulation behavior
Be Aware of your Model’s Differences

Two crucial types of differences between the model and its new environment:

- Different assumptions and constraints about the synthetic environment
  
- Different ways objects and processes are handled within your model and how the new environment handles them (includes Scope, Resolution, Structure)

And, differences in linguistics for objects and processes:
- Same context but referred to differently
- Different context but referred to using the same name
Reuse Issues for the:

M&S SOFTWARE ENTERPRISE
The M&S Reuse Enterprise

• Entire company or a single business unit

• Dedicated to M&S development and/or integration

• Experienced in M&S issues

• Interested in enterprise improvement issues such as reuse

Example: Raytheon Company vs Raytheon NCS
M&S Reuse Maturity

- Follows the Reuse Maturity model as presented earlier
- Easier to adopt when the Enterprise works with the same models over time
- Involves a cultural change to treating Conceptual Modeling as a distinct but important contribution to success

<table>
<thead>
<tr>
<th>Reuse Maturity</th>
<th>M&amp;S Reuse Maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial/Chaotic</td>
<td>No planned efforts to support Reuse</td>
</tr>
<tr>
<td>Monitored</td>
<td>Individuals and Teams do ad hoc reuse</td>
</tr>
<tr>
<td>Coordinated</td>
<td>Archive for models; incentives for reuse</td>
</tr>
<tr>
<td>Planned</td>
<td>Conceptual Modeling required; models are recorded</td>
</tr>
<tr>
<td>Ingrained</td>
<td>Model awareness and reuse is required</td>
</tr>
</tbody>
</table>
Thank You!

Emily Andrew
Raytheon Company
Network Centric Systems
Marlborough, MA 01752
Emily_B_Andrew@raytheon.com

Charles D. Turnitsa
VMASC/ODU
Suffolk, VA 23435
cturnits@odu.edu

Andreas Tolk, Ph.D.
EMSE/ODU
Norfolk, VA 23529
atolk@odu.edu