Applying UML to System Engineering
Some Lessons Learned
Murray Cantor
Principal Consultant
Mcantor@rational.com
Topics

- Background
- Customers needs
- What has worked
- Strengths of UML
- Shortfalls
- Next steps
Background

- Ongoing interest in systems throughout last 15 years
- Increasing demand over last 3 years in applying to UML semantics and associated architecture processes
  - Ongoing engagements at various sites of Lockheed-Martin, Boeing, Raytheon, others
- Increasing commitment of Rational to this community
  - Whitepaper published and small cadre of experienced field staff
  - In process development of Rational Unified Process add-in
  - Participation in OMG SE DSIG
<table>
<thead>
<tr>
<th>Customers Needs</th>
</tr>
</thead>
<tbody>
<tr>
<td>✷ Better communication between hardware software communities</td>
</tr>
<tr>
<td>✷ Better architecture semantics for large systems</td>
</tr>
<tr>
<td>- Less stovepipes – many current systems have unacceptable cost of ownership</td>
</tr>
<tr>
<td>- Top-down OOAD methods</td>
</tr>
<tr>
<td>- Support for iterative system development</td>
</tr>
<tr>
<td>✷ Better support for conceptual, analysis designs</td>
</tr>
<tr>
<td>- Technology opportunities require large trade space</td>
</tr>
</tbody>
</table>
What has worked

- Use case flowdown: Derivation of use cases to elements of logical architecture
  - Derivation of system services
  - Derivation of subsystem use cases, services
  - Use case anatomy: traceability between level of requirements in logical decompositions

- ‘Locality’ Diagrams for conceptual physical partitioning
  - Support design trades
**Strengths of UML**

- Encapsulation, abstraction, generalization
- Environment for dealing with complex systems
- Relationship between static and dynamic views
- Rich association semantics
UML Shortfalls: System semantics

- Standard definitions of system have two aspects:
  - Black box: A system is an entity that has behavior and provides services that enable it to collaborate with other entities to meet a business purpose or mission
  - White Box: A system is made up of hardware, software, workers, data,

- Need semantics to describe system blackbox elements
  - Attributes
  - Operations/services
  - Constraints

- System semantics are important for specifying the context more formally than the use case diagram found in UML as well as providing a means for modeling system specifications
Subsystems in 1.X are Classifier and a package
- inconsistent,
- Incomplete

Need semantics to describe associations of resources, components to subsystems

Need relationship of system and subsystem semantics
UML Shortfalls: Distribution of resources

- The distribution of the finite physical resources that provide the system services are captured in node diagrams
  - Confusion between hosting resources and system resources
- There needs to be more clarity on specifying levels of specificity in node diagrams
  - The two currently defined levels
    - descriptor - design
    - Instance - implementation
  - There needs to be a corresponding version at the analysis level to support ‘conceptual’ design and design trades
  - Additionally, the semantics of each type should be made explicit. Also the distinction of a worker as an actor and a worker as a system component should be expressible
System engineers need a term to mean physically delivered item such as a hardware assembly or an executable file.

The term ‘component’ is commonly used for such an item. However, the use of component in various UML 2.0 drafts seems to mean something more abstract.

If component can no longer be used to mean a ‘delivered item’, a new term is needed.

The typically many-to-many association between conceptual and design entities and delivered items needs to be modeled.
UML Shortfalls: Assemblies

- A system can be decomposed into parts from which it is physically assembled,
- Assemblies have physical properties, weight, thermal conductivity, distribution of mass, and the like, which are of interest to the hardware engineers on the system development effort.
- Further assemblies interact physically. The semantics of the components with their budgeted or derived physical properties is not found in UML 1.X.
- The current UML 2.0 draft does have inclusion semantics that may suffice, but these semantics need to be considered from the SE point of view.
system elements often interact through the sequencing of events. The current sequence and collaboration diagrams lack sufficient semantics to deal with these interactions.

The improved semantics in UML 2.0 need to be considered from an SE perspective.
Next steps

- Continued participation in OMG
  - U2P group
  - SE DSIG
- Adding precision to RUP SE process descriptions
- RUP product add-in web pages in process