Application of Object Oriented Technologies in Developing the Resource Management Segment for the Air Force Satellite Control Network

Mike Farmer (michael.farmer@lmco.com)
Lockheed Martin Federal Systems

Tom Gallini (gallini@courier3.aero.org)
The Aerospace Corporation
Outline

- What is RMS?
- Overview of Methodology
- Capturing Customer/User Inputs
- Modeling System Behavior
- Detailed Design
- User Benefits
- Acquisition Agency Benefits
What is RMS?


- Resource Scheduling
  - Develops and Maintains Schedules
  - Automatic Schedule Deconfliction
  - “What-if” Analysis Tools for Hard Conflict Resolution

- Orbit Analysis
  - Generate Station and SV Crosslink Events
  - Detect RFI Conflicts
  - Perform Collision Detection and Avoidance Analysis

- Resource Control and Monitoring
  - Authorize Configuration/Deconfiguration of Comm/Range Resources
  - Tracks Status of Reportable SCN Resources

- Network Performance Evaluation
  - Report Resource Status, Usage, and Performance
Overview of Methodology

Based on The 4+1 Model View -- Philippe Kruchten

Logical View

Process View

Scenarios

Dev View

Physical View
Capturing Customer/User Inputs
Requirements & Use Cases

Users, Customer, and Developer

Identify Actors

Identify Uses

Develop Textual Scenarios

<table>
<thead>
<tr>
<th>Scenario Step</th>
<th>Exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduler selects schedule for what if processing.</td>
<td>Selection not valid</td>
</tr>
<tr>
<td>RMS does ...</td>
<td></td>
</tr>
</tbody>
</table>

Actor | Use Case | Description
--- | --- | ---
Scheduler | “What If” | Conduct what if on working schedule
Capturing Customer/User Inputs
Uses of Use Cases

- System Test
  - Maintain traceability from Segment Specification into System Concept
  - Scenarios within use cases form test scenarios
- Users Manual
  - Scenarios define uses of the system, used as starting point for users manual.
  - Actor steps are expanded in the users manual.
- HMI Prototyping
  - Scenarios used to identify required screens.
  - HMI prototype used to verify scenario steps.
- Scenarios for System Design
  - Provides scenarios for design (more on this next)
Modeling System Behavior
(Allocation of Services)

- Develop a high level logical model.
  - Abstractions (class category) based on subject matter.
- Develop a message trace for each scenario.
  - Each step in scenario is traced to one or more steps in message trace.
  - Additional system detail discovered.
- Operations on a class category defines it’s public services.
Detailed Design

- A detailed logical model is developed for each class category.
- Each public service is defined within a class.
- The behavior of the service is defined.
- Changes to services controlled at the system model.

System Model

:Actor → :Obj1 → :Obj2 → :Objn

Oper1() → Oper2() → Oper3() → Display()
Detailed Design
Performance Requirements and Construction

• Develop a high level process model.
  • Map logical elements into processes.
  • Evaluate system performance on physical architecture.
• Formalize Development View
  • Map logical elements into subsystems
  • Design make file, directory structure, etc.
Putting it Together

Traceability
Verification

Use Case
Scenario 1
Scenario 2

System Model

Process View

Dev View

RMS Segment Spec
RMS Sys Con

:Actor :Obj1 :Obj 2

Oper()

Oper()

Oper()

Oper()

Oper()

Oper()

Oper()

Oper()

Oper()

Oper()

Oper()

Oper()
User Benefits

- User Benefits
  - Early Insight Into System Operations
  - Common Understanding of Requirements
  - Users Follow System from Concept through High Level Design
  - Early User Buy-in

- Acquisition Agency Benefits
  - Early Insight Into Design
  - Early Customer Buy-in
  - Common Understanding of Requirements
  - Used as Starting Point for ECP Activity
    - Eases Estimation
    - Eases Reviews
Summary

- Object Oriented technologies provide a sound basis for architecture definition.
- An approach employing a common thread through architecture definition ensures a complete design.
- Customer and user involvement within the methodology framework is key to success.
  - Involvement affords numerous benefits to the customer, user and developer.