Situational Awareness
Architectural Patterns

Mike Gagliardi, Bill Wood, Len Bass – SEI
Manuel Beltran – Boeing

For additional Info: wgw@sei.cmu.edu
Motivation

- **Software Patterns** are the codification of common problems within a domain using a template
  - problem, context, rules for solution, examples

- **Pattern Languages** describe the relationships between patterns - straightforward

- Patterns have become prominent in Software Development

- **Templates** had been proposed for both System and System-of-Systems (SoS) patterns
  - Cloutier (Stevens Institute), Open Systems Group
Getting Started

- Primary interest is in SoS
- Common functionality to focus efforts
  - Situational Awareness (SA)
- Chose SoS Pattern Template
  - Added Quality Attributes (performance, availability, security)
- Developed a Questionnaire (somewhat SA specific)
- Conducted 4 interviews (1 was a follow-on)
Analysis

Extracted data from the interviews into individual 5 SoS patterns (functional)

- Capture the essentials of Quality Attributes and variants in the individual patterns (functional)
- Use Mission Threads to inter-relate pattern functionality
- Extract Quality Attribute requirements from steps in the Mission Thread
- Use a pattern language to select variants to support architects
Situational Awareness Patterns

- Doctrine for displaying track data by hierarchy and geography
- LoS Radio communication is low-bandwidth and subject to splits and merges
- Signal Corp. develops multi-cast channels based on force structure, geography, intelligence and enemy posture
Variants

- GPS sensor is treated differently from other sensors
- Fusion of tracks can be done
  - each platform
  - SoS wide basis
  - multi-regional basis
Context

Command and Control
- Brigade and below
- Manned and unmanned vehicles, unmanned sensors, dismounted soldiers
- Terrain map, streets, buildings, bridges etc
- Own, enemy, coalition forces, non-combatants

Fusion
- Each platform has multiple sensors, each developing sensor tracks
- Each platform fuses tracks from all own sensors forms platform tracks
- Options on cross-platform track formation
  - Send all tracks to everyone and fuse locally (high bandwidth, simple discovery)
  - Send tracks to hierarchical and geographical neighbors (low bandwidth, complex discovery)
  - Send all tracks to single brigade fusion engine (high bandwidth, simple discovery)
  - Send to multiple regional fusion engines (low bandwidth, reasonable discovery)
### Performance - Problem

<table>
<thead>
<tr>
<th>Quality Attribute</th>
<th>Concern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Timeliness</td>
<td>Correlating a new platform track with an existing regional track (formed from tracks from other platforms) requires a search through a number of nearby regional tracks and is compute-time sensitive. The RTF must fuse incoming platform tracks with existing regional tracks and create new regional tracks as necessary. The processing requirements to do this will depend on the number of sensed EoIs in the region and the number of platforms that sense these EoIs and send platform tracks to the RTF - the <em>workload</em> for the RTF.</td>
</tr>
<tr>
<td></td>
<td>Bandwidth</td>
<td>Sending platform tracks to the RTF should have a predictable, limited communication bandwidth usage. If a platform track is updated based on multiple asynchronous updates from different sensors, then sending after each update increases bandwidth usage. Since all platform tracks are sent to the RTF platform, it should have a central position in the dynamic network structure, such that few communication “hops” are needed to move data from individual platforms to the RTF.</td>
</tr>
</tbody>
</table>
## Performance - Solution

<table>
<thead>
<tr>
<th>Quality Attribute</th>
<th>Concern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Timeliness</td>
<td>Each platform track has a consistent identification and the RFT remembers the historically associated regional track.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The RTF has the capability to process a well defined workload that exceeds the maximum workload expected in the field.</td>
</tr>
<tr>
<td>Network Bandwidth Usage</td>
<td>Simulation exercises have been conducted with multiple regions, and a high concentration (above expectations) of EoIs and sensor platforms. They have demonstrated that the bandwidth consumed by sending tracks to multiple regional fusion engines is acceptable.</td>
<td></td>
</tr>
</tbody>
</table>
## Availability - Problem

<table>
<thead>
<tr>
<th>Quality Attribute</th>
<th>Concern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>RTF Failure</td>
<td>When an RTF fails, the recovery could be to an RTF in the same platform or in another “standby” platform without dropping tracks.</td>
</tr>
<tr>
<td></td>
<td>Splitting and Merging</td>
<td>When the communication network splits, the region will have to split into multiple fusion regions to accommodate the communication network split.</td>
</tr>
<tr>
<td></td>
<td>Saturation</td>
<td>When the workload being presented to the RTF approaches its processing limitations, the workload must be decreased. This can be done by informing some of the platforms to reduce their sending frequency for some platform tracks, or by re-configuring the regions.</td>
</tr>
<tr>
<td></td>
<td>Reliability of sending platform tracks</td>
<td>Since platform track updates are sent periodically, they need not be received reliably. However, new platform tracks must be received reliably.</td>
</tr>
</tbody>
</table>
## Availability- Solution

<table>
<thead>
<tr>
<th>Quality Attribute</th>
<th>Concern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>TF Failure</td>
<td>When the individual TF in a platform fails, a redundant copy will be standing by in the same platform to take over responsibility within 5 seconds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When an RTF fails, another platform was assigned as a standby RTF, and contains the necessary historical data to restart within 5 seconds. Simulation exercises demonstrated that no tracks were dropped under a heavy workload. A new standby platform is immediately initiated.</td>
</tr>
<tr>
<td></td>
<td>Splitting and Merging</td>
<td>A multi-region simulation with a heavy workload in each region was conducted. When the network split and a region crossed both networks, the region was successfully split without dropping tracks.</td>
</tr>
<tr>
<td></td>
<td>Saturation</td>
<td>During a multi-regional simulation, the workload in one region was cranked up till saturation of the RTF occurred. The RTF sent messages to platforms causing their frequency of sending platform tracks to be reduced, and no tracks were dropped.</td>
</tr>
<tr>
<td></td>
<td>Reliability of delivery</td>
<td>There is no reliable protocol within the communication system, so the reliable messaging will have to be done at the application level.</td>
</tr>
</tbody>
</table>
## Communications System

<table>
<thead>
<tr>
<th>Quality Attribute</th>
<th>Concern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Latency between Send and Receive</td>
<td>This depends on the number of hops that a message takes between the sender and receiver. And on the wait time in queues in hopping platforms. As the workload presented to the network increases, the wait time also increases.</td>
</tr>
<tr>
<td></td>
<td>Bandwidth</td>
<td>The network bandwidth used depends on the number of messages sent. If a company has a multi-cast channel assigned to all members, then a message sent by a platoon member will be received by all company members; not necessarily in a single hop, since they may not all be in LoS of the sender.</td>
</tr>
<tr>
<td></td>
<td>Bandwidth</td>
<td>The network will operate well when it is lightly loaded, but will become saturated at high traffic. When this occurs, critical messages must still get delivered with a reasonable latency, and less important messages must be suppressed and/or discarded</td>
</tr>
</tbody>
</table>
Mission Thread Context

- Each company has a single logical multi-cast channel connecting all platforms.
- Each company and the brigade commander share a separate logical multi-cast channel.
- The assignment is “defensive”, the brigade communication structure is well understood, and “discovery” is done statically.
Initial detection takes place via Ground Radar. A SPOT Report modifies the BSO to assign affiliation and classification to the vehicle. As the vehicle moves west on the dirt road, the vehicle track is transferred to adjacent fusion regions. Once the vehicle passes beyond the radar sensor Line of Sight (LoS), SPOT Reports maintain the track on the vehicle.

Assume deployment is Mountainous Terrain resulting in a Subscription Radius of 1.5 Km (Urban Canyon).

Once initial SPOT report is created, Co B 3rd PL, Co B 2nd PL, and Co A 3rd PL move to intercept the Goat Herder truck. This places Co C 2nd PL and Co B 3rd PL with subscription radius, resulting in geographical dissemination, and places Co B 2nd PL and Co A 3rd PL within geographical subscription radius.
Extractions from the Mission Thread

Performance

- The track is marked suspicious (yellow) or hostile (red) by a platoon vehicle and distributed both hierarchically and geographically (7km²) within 10 seconds
  - Worst case Compute + Communication < 10 seconds
- Radar ground tracks are distributed across the battalion from a single source and distributed hierarchically (168km²) within 1 minute.
  - (SM+TF)Radar+(TF+RTF+TD)CoC+(TD+RTF+TD)CoB+(TD+RTF+TD)CoA+(TD+TF+DT)3rdPL+4*T_{ios}+4*T_{i} < 1 minute

Availability

- Each company command post (XO) serves as the RTF for a region. The company commander platform (CDR) is in standby to take over and has all the timley updates.
Pattern Language

Platoon  Company  Others

$Tp2p + Tsm + 2 \cdot Ttf + 3 \cdot Ttd + Tdt + 3 \cdot Ti + 3 \cdot Tlos \leq 10$
Pattern Language- Availability

RTF Backup

TF → TD → DT

RTF Primary

TF → TD → DT

12/11/2012
Conclusion

- It’s a lot of work to establish the patterns from interviews and fill in the templates.
- Deciding what should be in the patterns and what should be in the language is easy to say and hard to do.
- The language can guide the architect, but leaves a complicated design for him to do.
- Having patterns is better than starting from scratch.
The SoS architect generates alternative high level designs and then chooses between them based on some measures of how they satisfy the architectural drivers (requirements, quality attributes).

The system capabilities involved in the alternatives are large scale and are distributed across different platforms.

The relationships between these capabilities are the focus of the SoS architect.
Capabilities

Track Fusion
• Fuse tracks of same environmental object from different sensors

Track Distribution
• Send tracks appropriately between platforms
• Take advantage of communication structure

Track Display
• Satisfy doctrine- hierarchical and geographical display of tracks
• Common Identifier for tracks of same object on different platforms
SoS Architects Alternatives

Fusion
Distribution
Display

Local Fusion

Centralized Fusion

Regional Fusion
## Important Capabilities

<table>
<thead>
<tr>
<th>#</th>
<th>Capability</th>
<th>Local</th>
<th>Central</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Network Bandwidth used</td>
<td>Highest (800)</td>
<td>Middle (720)</td>
<td>Lowest (640)</td>
</tr>
<tr>
<td>2</td>
<td>Unique Identifier</td>
<td>Highest</td>
<td>Lowest</td>
<td>Middle</td>
</tr>
<tr>
<td>3</td>
<td>Remote Display Latency</td>
<td>Lowest</td>
<td>Highest</td>
<td>Middle</td>
</tr>
<tr>
<td>4</td>
<td>TF Failure- No dropped Tracks</td>
<td>Lowest</td>
<td>Lowest</td>
<td>Middle</td>
</tr>
<tr>
<td>5</td>
<td>Hierarchical Display</td>
<td>Same</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td>6</td>
<td>Geographical Display</td>
<td>Highest</td>
<td>Lowest</td>
<td>Middle</td>
</tr>
<tr>
<td>7</td>
<td>Network Split</td>
<td>Easiest</td>
<td>Highest</td>
<td>Middle</td>
</tr>
</tbody>
</table>