Chapter 17

Model-Based Testing for Systems of Systems
Some Early Definitions
(each of these is too vague)

- A “super system”, not a monolithic system
- A collection of cooperating systems
- A collection of autonomous systems
- A set of “Component Systems”
- A set of “Constituent Systems”

What differentiates these from an automobile? The human body? an airplane?

adapted from: “Architecting principles for systems-of-systems”, Mark Maier, in Systems Engineering, vol 1 Issue 4, pp. 251 - 313
Mark Maier’s Examples

• “air defense networks”
• “the internet”
• An emergency response team

adapted from: “Architecting principles for systems-of-systems”, Mark Maier, in Systems Engineering, vol 1 Issue 4, pp. 251 - 313
Mark Maier’s Distinguishing Characteristics

• Operationally independent components (Constituent Systems)
• Managerial Independence
• “Evolutionary nature” (of their development)
• “Emergent behaviors”
• “Geographic distribution limits information sharing”

adapted from: “Architecting principles for systems-of-systems”, Mark Maier, in Systems Engineering, vol 1 Issue 4, pp. 251 - 313
Levels of “Collaborative Interaction”

- “Directed”—designed, built, and managed for a specific purpose. (e.g., an air defense system)
- “Acknowledged”—known objectives, designated central command/control
- “Collaborative”—limited central management
- “Virtual”—no central management.
  - Deliberate
  - Accidental

adapted from: “Architecting principles for systems-of-systems”, Mark Maier, in Systems Engineering, vol 1 Issue 4, pp. 251 - 313
Generic View of Systems of Systems

- Constituent System 1
- Control Center
- Constituent System 2
- Constituent System 3
- Constituent System 4
Sample Systems of Systems

- Garage Door Controller—a directed SoS
- Air Traffic Management System—an acknowledged SoS
- The GVSU Snow Emergency System—a collaborative SoS
- A Credit Union—a virtual SoS
- Each SoS diagrammed in SysML
Garage Door Controller

- Digit Keypad
- Wireless Receiver
- Portable Opener
- Wall-mount button
- Light Beam
- Garage Door Controller
- Drive Motor
- Lamp
- Extreme Limit Sensor
- Obstacle (Resistance) Sensor
Air Traffic Management (ATM) System

- Arriving Aircraft
- Departing Aircraft
- Runways Monitor
- Weather Instruments
- Lateral Separation
- Vertical Separation

Air Traffic Management
GVSU Snow Emergency System

Information Technology

Snow Emergency Notification

U.S. Weather Bureau

GVSU Campus Safety

GVSU Grounds

Parking Lot Snow Removal

County Highway Dept.

Local Snowplow Services

Television and Radio Stations
The Rock Solid Credit Union

- National Credit Union Association
- RSFCU (Credit Union)
  - Fannie Mae
  - U.S. Federal Reserve Bank
  - Credit Card Service
  - Credit Rating Bureau
  - Mortgage consultant
  - Mortgage Broker
Software Engineering for Systems of Systems (SoS)

- Requirements Elicitation
- Specification (modeling)
- Protocol selection/definition
- Communication
- Testing

Lane, Jo Ann, “System of Systems Capability-to-Requirements Engineering, University of Southern California, Viterbi School of Engineering, webinar given February, 2012
Requirements Elicitation for SoS

- Identifying resources—potential constituent systems, and modeling them with SysML
- Determining options—responsibilities and dependencies
- Assessing options—expressed as Use Cases
- Identifying a workable combination of constituent systems
- Allocating responsibilities to constituent systems

Lane, Jo Ann, “System of Systems Capability-to-Requirements Engineering”, University of Southern California, Viterbi School of Engineering, webinar given February, 2012
Specification (modeling) for SoS

- Use cases to show interaction among constituent systems
- SysML (context diagram) of constituents
- Classes per constituent
  - Attributes describe assets, capabilities
  - Methods describe responsibilities

Lane, Jo Ann, “System of Systems Capability-to-Requirements Engineering”, University of Southern California, Viterbi School of Engineering, webinar given February, 2012
Air Traffic Management System “Classes”

- Arriving and Departing Aircraft
  - Responsibilities: Communicate with air traffic controller
  - Services:
    - Fly aircraft
    - Land aircraft
    - Remain prepared for emergency situations
- Separation Instruments
  - report vertical and horizontal separation
- Weather Instruments
  - report wind speed and direction
  - report visibility
Air Traffic Controller “Class”

• Responsibilities to other constituents
  – Incoming aircraft
  – Departing aircraft
  – Runway (status)
  – Separation instruments
  – Weather instruments

• Services
  – Assign runways based on weather conditions
  – Monitor separation instruments
  – Assign landing clearance
  – Assign take-off clearance
  – Maintain runway status
<table>
<thead>
<tr>
<th>Description</th>
<th>SoS UC1: Normal aircraft landing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The procedure that governs an arriving aircraft under normal conditions.</td>
</tr>
<tr>
<td>Actor(s)</td>
<td>1. Air Traffic Controller</td>
</tr>
<tr>
<td></td>
<td>2. Incoming aircraft</td>
</tr>
<tr>
<td></td>
<td>3. Separation sensors (vertical, lateral, and time)</td>
</tr>
<tr>
<td>Pre-conditions</td>
<td>1. Designated runway clear</td>
</tr>
<tr>
<td></td>
<td>2. Incoming aircraft ready to land</td>
</tr>
</tbody>
</table>

**Action Sequence**

<table>
<thead>
<tr>
<th>Actor</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming aircraft</td>
<td>1. Requests clearance to land</td>
</tr>
<tr>
<td>Air Traffic Controller</td>
<td>2. Checks all separation sensors</td>
</tr>
<tr>
<td>Lateral Separation</td>
<td>3. OK</td>
</tr>
<tr>
<td>Vertical Separation</td>
<td>4. OK</td>
</tr>
<tr>
<td>Time Separation</td>
<td>5. OK</td>
</tr>
<tr>
<td>Air Traffic Controller</td>
<td>6. Landing clearance given</td>
</tr>
<tr>
<td>Incoming aircraft</td>
<td>7. Initiates landing procedures</td>
</tr>
<tr>
<td>Incoming aircraft</td>
<td>8. On assigned runway</td>
</tr>
<tr>
<td>Incoming aircraft</td>
<td>9. Taxi to assigned gate</td>
</tr>
<tr>
<td>Air Traffic Controller</td>
<td>10. Landing complete.</td>
</tr>
<tr>
<td>Post-conditions</td>
<td>1. Runway available to other aircraft</td>
</tr>
</tbody>
</table>
ATM SoS (partial) Sequence Diagram

Air Traffic Controller

Incoming Aircraft

Lateral Separation Sensor

Vertical Separation Sensor

Landing Time Separation Sensor

Landing Request
Lateral Separation OK?
Vertical Separation OK?
Landing Time Separation OK?
Communication Primitives for SoS

- Depends on Maier’s Levels of “Collaborative Interaction”
- Key consideration: How do constituent systems communicate?
- Express Use Cases as EDPNs
- Use “swim lanes” to show interaction
  - Petri Net Primitives
  - ESML Prompts
  - Additional Primitives
Petri Net Conflict

The transitions function 1 and function 2 are both enabled. Firing either one consumes the token in place p2, thereby disabling the other.
Petri Net Interlock

The only way the secondary action can fire is for the preferred action to fire first, thereby enabling the interlock place $i$. 

![Petri Net Diagram]

- $p1$ to $p2$:
  - Preferred action
- $i$:
  - Interlock place
- $p2$ to preferred action:
  - Secondary action
ESML Prompts

• The Extended System Modeling Language (ESML) was an industry initiative to combine the best features of rival “real-time” extensions to dataflow diagrams.

• The ESML prompts describe how one task can affect others...
  – Enable
  – Disable
  – Activate (Enable followed by Disable)
  – Trigger
  – Suspend
  – Resume
  – Pause (Suspend followed by Resume)
Enable, Disable, and Activate

The e/d place input to the controlled action must be marked by the Enable action. (It is an interlock.) The controlled action and the Disable action are in Petri Net conflict with respect to the e/d place.
Trigger

The t place input to the controlled action must be marked by the Trigger action. (It is an interlock.) The difference between Enable and Trigger is that the controlled action must be ready to fire.
Suspend, Resume, and Pause

• By itself, Suspend acts like a Disable prompt.
• Similarly, a Resume acts like an Enable.
• The intent of the Pause prompt is to Suspend an ongoing activity for a while, and then let it resume “where it left off.”
• Notice that a Resume must follow a Suspend, due to the s place (which is an interlock).
• (See the next slide for the Petri Net.)
Suspend, Resume, and Pause

Diagram:
- Step 1
- Intermediate step
- Final step
- Resume
- Suspend
Swim Lane Petri Nets

- Swim Lane Petri Nets borrow the swim lane concept from the UML, where it is used to show parallel activities.
- Since constituents in a System of Systems are separate, Swim Lane Petri Nets exactly show the communication among constituents.
- (This corresponds to the use of orthogonal regions in the StateChart notation to denote concurrent activities.)
- (Swim Lane Petri Nets have been shown to be formally equivalent to three types of UML StateCharts. See text for the reference.)
Additional Prompts for Systems of Systems

- Request: constituent A requests a service from Constituent B
- Accept: constituent B honors the request from constituent A, completes the task, and returns to constituent A.
- Reject: constituent B rejects the request from constituent A, and returns (immediately) to constituent A.
- Postpone: constituent B postpones the request from constituent A, completes its more important action, and then honors the request from constituent A.
Request
(Constituents A and B are in separate Swim Lanes)
Accept
(Constituents A and B are in separate Swim Lanes)
Reject
(Constituents A and B are in separate Swim Lanes)
Accept and Reject as Petri Net Conflict
(Constituents A and B are in separate Swim Lanes)
Connections among Request, Accept, and Reject
(Constituents A and B are in separate Swim Lanes)
Postpone
(Constituents A and B are in separate Swim Lanes)
November 1993 Incident

Landing Aircraft

- Landing Clearance
  - lower landing gear
    - gear down
      - lower flaps
        - flaps down
          - land aircraft

Air Traffic Controller

- resume
  - s
    - recognize emergency
      - Trigger
        - suspend
November 1993 Incident

In November, 1993 a commercial aircraft was on its final landing approach to a runway at Chicago's O'Hare International Airport. When the incoming aircraft was at an altitude of about 100 feet, a pilot waiting to take off saw that the landing aircraft had not lowered its landing gear. There is no direct communication between landing and departing aircraft, so the pilot contacted the O'Hare field control tower about the impending disaster. The control tower waved off the landing aircraft, and a disaster was avoided. This is the subject of our second use case and sequence diagram. In this use case, Aircraft L is the landing aircraft, and aircraft G is the one on the ground. We can imagine that the second use case could be a continuation of the first one at action step 7. We can also imagine that everyone involved was very relieved once the post-condition was attained.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SoS UC2: November 1993 Incident at O'Hare Field</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Aircraft on final approach had landing gear up. Pilot on taxi way saw this and notified control tower.</td>
<td></td>
</tr>
<tr>
<td><strong>Actor(s)</strong></td>
<td>1. Air Traffic Controller</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Incoming aircraft L</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Aircraft G waiting to take off.</td>
<td></td>
</tr>
<tr>
<td><strong>Pre-conditions</strong></td>
<td>1. Aircraft L cleared to land</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Aircraft G waiting to take off</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Aircraft L has landing gear up</td>
<td></td>
</tr>
</tbody>
</table>
| **Action Sequence** | | |}

<table>
<thead>
<tr>
<th>Actor</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Traffic Controller</td>
<td>1. Authorizes aircraft L to land</td>
</tr>
<tr>
<td>Aircraft L</td>
<td>2. Initiates landing preparation</td>
</tr>
<tr>
<td>Aircraft L</td>
<td>3. Fails to lower landing gear</td>
</tr>
<tr>
<td>Aircraft L</td>
<td>4. 100 feet above end of assigned runway</td>
</tr>
<tr>
<td>Aircraft G</td>
<td>5. Aircraft G pilot radios Air Traffic Controller</td>
</tr>
<tr>
<td>Air Traffic Controller</td>
<td>6. Terminates landing permission</td>
</tr>
<tr>
<td>Aircraft L</td>
<td>7. Aircraft L aborts landing</td>
</tr>
<tr>
<td>Aircraft L</td>
<td>8. Aircraft L regains altitude over runway</td>
</tr>
<tr>
<td>Air Traffic Controller</td>
<td>9. Instructs aircraft L to circle and land</td>
</tr>
<tr>
<td>Air Traffic Controller</td>
<td>10. Thanks pilot of aircraft G</td>
</tr>
<tr>
<td>Air Traffic Controller</td>
<td>11. Authorizes aircraft L to land</td>
</tr>
<tr>
<td>Aircraft L</td>
<td>12. Landing complete.</td>
</tr>
<tr>
<td><strong>Post-conditions</strong></td>
<td>1. Runway available to other aircraft</td>
</tr>
</tbody>
</table>
November 1993 Incident Sequence Diagram

Incoming Aircraft L

→ Landing Authorized

← Landing Cancelled

← New landing instructions

Air Traffic Controller

← Pilot reports emergency

Waiting Aircraft G

→ Thanks pilot
November 1993 Incident Swim Lane Petri Net

Landing Aircraft
- Landing Clearance
  - lower landing gear
    - gear down
      - lower flaps
        - flaps down
          - land aircraft

Air Traffic Controller
- resume

Pilot on Ground
- recognize emergency
  - Trigger
  - suspend

Chapter 17 Systems of Systems