STPA Applied to Autonomous Vehicles

Dr. John Thomas (MIT)
jthomas4@mit.edu

Jeff Stafford (Renesas)
Shaun Mooney (Codethink)
Matthew Green (Codethink)
Original Company Plan

- Goal: Demonstrate self-driving car on public roads
- Use Baidu’s Apollo software for self-driving functions
- Company is convinced that a systems approach to safety is required
- Decision: Use state-of-the-art STPA to demonstrate due diligence
Control Structure: Level 1

Program Management

Go/No-Go

System Integrators

Test Route Planners

STPA Team

Test route requirements

Technical requirements

Results

Procedures

Training requirements

Trainers

Safety Driver(s)

Autonomous Vehicle

Environment

Go/No-Go

Design

Test Route
Control Structure: Level 1

Program Management

System Integrators

Test Route Planners

Go/No-Go

Go/No-Go

Go/No-Go

Post-drive reviewers

Safety Driver(s)

Autonomous Vehicle

Environment

Fixes

Go/No-Go

© Copyright 2019 John Thomas
Control Structure Refinement

Level 1

Program Management

“Go” criteria

Objectives

Go/No-Go

Engineering Team

System Integrators

Test Route Planners

Post-drive reviewers

Safety Engineers

Legal

Go/No-Go

State Law, Insurance, Etc.

Trainers

Go/No-Go

Safety Driver(s)

Autonomous Vehicle

© Copyright 2019 John Thomas
Control Structure Refinement

Level 1

Program Management

Go/No-Go

“Go” criteria

Objectives

Engineering Team

System Integrators

Test Route Planners

Post-drive reviewers

Safety Engineers

Legal

Go/No-Go

State Law, Insurance, Etc.

Trainers

Safety Driver(s)

Autonomous Vehicle

Level 2

Pilot

Copilot

Apollo HMI

Apollo 2.0 Software System

Dataspeed

Sensors

Lincoln MKZ

Safety Drivers

Go/No-Go

ESTOP

© Copyright 2019 John Thomas
Complete Control Structure

Program Management

Drivers

Autonomous SW

Vehicle
Unsafe Control Actions

UCA: Pilot does not provide ESTOP when autonomy is providing excessive throttle

UCA :=
<Source Controller> → Pilot
<Type> → does not provide
<Control Action> → ESTOP
when
<Context> → Autonomy providing excessive throttle

Lincoln MKZ

Apollo 2.0
Software System

Dataspeed

Sensors

Pilot

Copilot

Apollo HMI
Human Interactions

• UCA: Pilot does not provide ESTOP when autonomy is providing excessive throttle
  • PM: Pilot believes autonomy was disabled due to manual braking cmds...

• Generated Requirement
  • Datataspeed must override all Apollo cmds when driver applies brake

• Existing Design/Requirements can cause this!
  • <22% braking will not override
  • Braking override independent from steering override
  • Will ignore driver braking overrides if Apollo sends IGNORE/CLEAR

Pilot

---

Copilot

Apollo HMI

Apollo 2.0
Software System

Dataspeed

Sensors

Lincoln MKZ

© Copyright 2019 John Thomas
Human Interactions

- UCA: Pilot does not provide ESTOP when autonomy is providing excessive throttle
  - PM: Pilot believes autonomy was disabled due to manual braking cmds...
- Generated Requirement
  - Dataspeed must override all Apollo cmds when driver applies brake
- Existing Design/Requirements can cause this!
  - <22% braking will not override
  - Braking override independent from steering override
  - Will ignore driver braking overrides if Apollo sends IGNORE/CLEAR
Generated Possible Requirements

• Dataspeed must override Apollo (all channels) when driver applies brake

• Apollo must not override driver (must not provide throttle, IGNORE/CLEAR, etc.

• Pilot/Copilot must be notified when manual commands do not result in automation override

• Pilot/Copilot test track training must include cases in which manual commands do not result in automation override (e.g. <22%)

• Post-drive review must identify any cases in which manual commands do not result in automation override

• Public road testing approval must stop if operation encounters manual commands that do not result in automation override (assumption violated)
Generated Possible Requirements

• Dataspeed must override Apollo (all channels) when driver applies brake

• Apollo must not override driver (must not provide throttle, IGNORE/CLEAR, etc.)

• Pilot/Copilot must be notified when manual commands do not result in automation override

• Pilot/Copilot test track training must include cases in which manual commands do not result in automation override (e.g. <22%)

• Post-drive review must identify any cases in which manual commands do not result in automation override

• Public road testing approval must stop if operation encounters manual commands that do not result in automation override (assumption violated)
Human Interactions

UCA: Pilot provides **manual steering** too late after autonomous mode exits

- Pilot believes vehicle is in autonomous mode
- Vehicle exits autonomous mode unexpectedly (e.g. fault occurs, ESTOP applied)
  - ESTOP applied by copilot during turn
  - ...

**Generated potential requirements**
- ESTOP must not cause sudden steering angle changes
- Pilot/copilot must have advance indication before autonomous mode ends
- ...

© Copyright 2019 John Thomas
Human Interactions

New question: Does existing system enforce this?

Answer: ESTOP activation results in immediate “return to position” torque.
- Can’t change.

Generated potential requirements
- ESTOP must not cause sudden steering angle changes
- Pilot/copilot must have advance indication before autonomous mode ends
- ...

Enforced?
UCA: Pilot provides **manual steering** too late after autonomous mode exits

- Pilot believes vehicle is in autonomous mode
- Vehicle exits autonomous mode unexpectedly (e.g. fault occurs, ESTOP applied)
  - ESTOP applied by copilot during turn
  - ...

Existing Design/Requirements will cause this!

- ESTOP designed to instantly remove power: “pull the plug”
- Results in immediate “return to position” steering wheel torque.
- Not configurable, can’t change.
New question: Does existing system enforce this?

Answer: ESTOP activation results in immediate “return to position” behavior.

- Can’t change

Resulting potential requirements

- **ESTOP must not cause sudden steering angle changes**
- **Pilot/copilot must have advance indication before autonomous mode ends**
- Copilot must confirm Pilot hands on wheel before providing ESTOP.
- Test track training must include copilot activation of ESTOP
- Test track training must include ESTOP activation during turns
- ...
New question: Does existing system enforce this?

Answer: ESTOP activation results in immediate “return to position” behavior.

- Can’t change

Resulting potential requirements

- **ESTOP** must not cause sudden steering angle changes
- Pilot/copilot must have advance indication before autonomous mode ends
- Copilot must confirm Pilot hands on wheel before providing ESTOP.
- Test track training must include copilot activation of ESTOP
- Test track training must include ESTOP activation during turns
- ...
UCA: Apollo provides throttle cmd when forward collision is imminent

- PM: Apollo incorrectly believes forward collision is not imminent

- Feedback: LIDAR, Camera, Braking status, AEB (automatic emergency braking)
  - Feedback inadequate, missing, etc.

Generated potential requirements

- Apollo must not provide throttle cmd when manual braking is applied

- Apollo must not provide throttle cmd when AEB engages

Enforced?
New question: Does Apollo respond to AEB feedback?

Answer: Apollo ignores AEB status. Operates independent of AEB.

Generated potential requirements
- Apollo must not provide throttle cmd when braking is applied
- Apollo must not provide throttle cmd when AEB engages
  How?
UCA: Apollo provides throttle cmd when forward collision is imminent
• PM: Apollo incorrectly believes forward collision is not imminent
• Feedback: LIDAR, Camera, Braking status, AEB (Automatic Emergency Braking)
• ...

Existing Design/Requirements will cause this!
- Apollo designed to ignore AEB and operate independently
- Apollo relies on AEB as an independent backup
- Apollo throttle commands are designed to spoof driver commands
- AEB is designed to never override driver commands
- Apollo is disabling AEB any time it sends a positive throttle command! (>50% of driving)
What needs to be monitored?
Should it ever intervene? If so, when?

Software team initially proposed 4 requirements
1. Each Apollo SW module is running
2. A single instance of each Apollo SW module is running
3. Each Apollo SW module is sending messages at the correct rate
4. Each Apollo SW module self-reports no internal faults

Is this everything? How do you know?
Decision: use STPA to check
Shall detect and warn copilot when:

- Apollo provides throttle commands while AEB is active
- Apollo provides throttle commands while driver applies brake
- Apollo provides throttle commands while parking brake engaged
- Apollo provides IGNORE/CLEAR cmd at any time
- ...

Shall block Apollo commands and report to copilot when:

- Apollo steering command specifies excessive steering rate (>TBD) that can destabilize vehicle
- Apollo positive throttle command when vehicle speed exceeds maximum velocity limit for planned test (>TBD)
- Apollo throttle command when not in autonomous mode
- Vehicle is in R when Apollo enters autonomous mode
- Vehicle door is open when Apollo enters autonomous mode
- ...

Company engineering decision:
Do not implement at this time
• 84 requirements identified
• Initially allocated to:
  • Safety Actuator Monitor (SAM)
  • Additional SW-based monitor tracking ROS (Robot Operating System) topics
• Team agreed to 20 SW requirements for Safety MCU
  • Highly dependent on a tight development schedule
  • Warning light used to inform Pilot/Co-pilot
# Existing Public Road Testing: Examples of Disengagements

<table>
<thead>
<tr>
<th>Date</th>
<th>Causal Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/11/2017</td>
<td>Localization error caused drift</td>
</tr>
<tr>
<td>01/13/2017</td>
<td>Perception discrepancy for an object caused braking with traffic behind</td>
</tr>
<tr>
<td>01/27/2017</td>
<td><strong>Planning discrepancy caused steering maneuver</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Causal Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/12/2017</td>
<td>System Fault</td>
</tr>
<tr>
<td>02/22/2017</td>
<td><strong>Planning discrepancy caused brake jabs</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Causal Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/03/2017</td>
<td><strong>Misclassification of traffic light detection</strong></td>
</tr>
<tr>
<td>03/13/2017</td>
<td>Braking upon engaging system</td>
</tr>
<tr>
<td>03/14/2017</td>
<td>Perception discrepancy caused no yield for cross traffic</td>
</tr>
<tr>
<td>03/14/2017</td>
<td>Planning discrepancy caused delayed braking for car that cut in and slowed quickly</td>
</tr>
<tr>
<td>03/15/2017</td>
<td>Perception discrepancy caused delayed yield at intersection</td>
</tr>
<tr>
<td>03/15/2017</td>
<td>Perception discrepancy caused proceeding during right on red with cross traffic</td>
</tr>
<tr>
<td>03/30/2017</td>
<td>Perception discrepancy for a pedestrian in crosswalk caused braking with traffic behind</td>
</tr>
</tbody>
</table>
Control Structure Refinement

Level 2

Pilot

Copilot

Apollo HMI

Apollo 2.0 Software System

Monitor / Guardian

Dataspeed

Sensors

Lincoln MKZ

Level 3

?
Control Structure vs. Data Processing Pipeline
Example data pipeline

Integrated Static and Dynamic Approaches to High-Assurance for Learning-Enabled Cyber-Physical Systems
https://rtg.cis.upenn.edu/assured-autonomy/
Another model – A control loop

- Localization
- Planning
- Perception
- Vehicle
- Control
Control Structure Refinement

Level 2

- Pilot
- Copilot
- Apollo HMI
- Monitor / Guardian
- Dataspeed
- Sensors
- Lincoln MKZ

Level 3

- Apollo 2.0
- Routing
- Planning
- Prediction
- Control
- Perception
- Localization

- New route request
- Route Waypoints
- Desired Trajectory
- Objects, Paths
- Objects, Scenery
- Location
- HD Map
- GPS
- Inertial reference
- Camera images
- Lidar images
- Radar images

Actuation (throttle, brake, steer, shift)

Vehicle status

Telephoto cam
Wide-angle cam
Lidar images
Radar images
Etc.

© Copyright 2019 John Thomas
Control Structure Refinement

**Level 2**

- Pilot
- Copilot
- Apollo HMI
- Monitor / Guardian
- Dataspeed
- Lincoln MKZ

**Level 3**

- Apollo 2.0
- Routing
- Planning
- Prediction
- Control
- Perception
- Localization
- HD Map
- GPS
- Inertial reference
- Camera images
- Lidar images
- Radar images

- New route request
- Route Waypoints
- Objects, Scenery
- Objects, Paths
- Location
- Vehicle status
- Telephoto cam
- Wide-angle cam
- Lidar images
- Radar images
- Etc.

- Desired Trajectory
- Location
- Objects, Scenery
- Objects, Scenery
- Actuation (throttle, brake, steer, shift)
- Vehicle status

- Env.

Thomas, 2019
New Scenario Approach using Basic Scenarios

1) Inadequate Controller Behavior
   - Feedback indicates …
   - Apollo does not …

2) Inadequate feedback/information
   - Vehicle is …
   - Controller receives inadequate feedback indicating …

UCA-1: Apollo does not continue providing brake control when vehicle stationary, vehicle path not clear

3) Inadequate Control Execution
   - Apollo provides …
   - <> not applied

4) Inadequate process behavior
   - <> applied
   - Vehicle is …

© Copyright 2019 John Thomas
1) Inadequate Controller Behavior
- Feedback indicates vehicle path not clear
- Apollo does not continue providing brake

2) Inadequate feedback/information
- Vehicle path is not clear
- Controller receives inadequate feedback indicating vehicle path is clear

UCA-1: Apollo does not continue providing brake control when vehicle stationary, vehicle path not clear

3) Inadequate Control Execution
- Apollo provides brake cmd
- Brakes not applied

4) Inadequate process behavior
- Brakes applied
- Vehicle does not stop in time

© Copyright 2019 John Thomas
### Basic Scenario Table:

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Not providing causes hazard (UCA-1)</th>
<th>Providing causes hazard (UCA-2)</th>
<th>Too early, too late, out of order causes hazard (UCA-3)</th>
<th>Stopped Too Soon / Applied too long causes hazard (UCA-4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario Type 1: Unsafe Controller Behavior</td>
<td>1) controller doesn't provide &lt;cmd&gt; 2) controller received feedback (or other inputs) that indicated &lt;context&gt;</td>
<td>1) controller provides &lt;cmd&gt; 2) controller received feedback (or other inputs) that indicated &lt;context&gt;</td>
<td>1) controller provides &lt;cmd&gt; too late/early/out of order 2) controller received feedback (or other inputs) that indicated &lt;context&gt; on time / in order</td>
<td>1) controller stops providing &lt;cmd&gt; too soon 2) controller received feedback (or other inputs) that indicated &lt;context&gt; on time</td>
</tr>
<tr>
<td>Scenario Type 2: Unsafe Feedback Path</td>
<td>1) feedback received by controller does not indicate &lt;context&gt; 2) &lt;context&gt; is reflected in information from controlled process</td>
<td>1) feedback received by controller does not indicate &lt;context&gt; 2) &lt;context&gt; is reflected in information from controlled process</td>
<td>1) feedback received by controller does not indicate &lt;context&gt; on time / in order 2) &lt;context&gt; is reflected in information from controlled process on time / in order</td>
<td>1) feedback received by controller does not indicate &lt;context&gt; 2) &lt;context&gt; is reflected in information from controlled process</td>
</tr>
<tr>
<td>Scenario Type 3: Unsafe Control Path</td>
<td>1) controller does provide &lt;cmd&gt; 2) &lt;cmd&gt; is not received by controlled process</td>
<td>1) controller does not provide &lt;cmd&gt; 2) &lt;cmd&gt; is received by controlled process</td>
<td>1) controller provides &lt;cmd&gt; on time / in order 2) &lt;cmd&gt; is received by controlled process too late/early/out of order</td>
<td>1) controller provides &lt;cmd&gt; with appropriate duration 2) &lt;cmd&gt; is received by controlled process with in appropriate duration</td>
</tr>
<tr>
<td>Scenario Type 4: Unsafe Controlled Process Behavior</td>
<td>1) &lt;cmd&gt; is received by controlled process 2) controlled process does not respond by &lt;…&gt;</td>
<td>1) &lt;cmd&gt; is not received by controlled process 2) controlled process does not respond by &lt;…&gt;</td>
<td>1) &lt;cmd&gt; is received by controlled process on time / in order 2) controlled process does not respond by &lt;…&gt;</td>
<td>1) &lt;cmd&gt; is received by controlled process with appropriate duration 2) controlled process does not respond by &lt;…&gt;</td>
</tr>
</tbody>
</table>

(Thomas, 2016), (Thomas, 2017)
1) Inadequate Controller Behavior
- Planning provides emergency stop cmd
- Feedback correctly indicates no imminent collision, traffic close behind

2) Inadequate Feedback / Other Info
- Feedback indicates imminent collision
- Collision is not imminent

3) Inadequate Control Execution
- Planning provides emergency stop cmd
- Brakes not applied
UCA-1: Planning provides emergency stop cmd when no imminent collision, traffic close behind

1) **Inadequate Controller Behavior**
- Planning provides emergency stop cmd
- Feedback correctly indicates no imminent collision, traffic close behind

Could occur if:
- Trajectory data is not within time-stamp tolerance
- Control command could not be computed
- Localization data has not been observed

© Copyright 2019 John Thomas
1) Inadequate Controller Behavior
- Planning does not provide stop cmd
- Feedback correctly indicates upcoming red light

2) Inadequate Feedback / Other Info
- Feedback indicates no upcoming red light
- Upcoming traffic light is red

Could occur if:
- Traffic-light not in map file
- Apollo not manually reset after map loads
- Apollo defaults to previous detected traffic light state
2) Inadequate Feedback / Other Info
- Feedback to planning indicates no obstacle
- Obstacle detected by sensors

Could occur if:
- Bike lane intersects with road
  (SW decision to filter out bicycles in bike lane)
UCA-4: Planning provides stop cmd when ....

3) Inadequate Control Execution
   - Planning provides stop cmd
   - Vehicle does not stop, loses control

Could occur if:
   - Command causes loss of traction
     (SW decision to ignore weather)
Level 2 Analysis (ad-hoc)

83 UCAs
~20 scenarios per UCA

Is there a better approach?

Scenario Analysis for Unsafe Control Actions: Brake Control

Unsafe by Not Providing

UCA-61: Apollo does not provide the brake control action when relative velocity and distance to an obstacle mean that a collision is imminent.

True statement from UCA context: The vehicle is approaching an obstacle with a velocity and acceleration vector that indicate a collision

Beit:
- Apollo incorrectly believes that the relative distance is lower than in reality so that there is no need to brake

Type 2 scenario:
- Controller receives incorrect feedback/information:
  - Information received: The feedback received is insufficient to accurately determine the relative velocity
  - How this could happen: The true statement above
    - The sensor (sensor) is compromised
    - Data on the vehicle CAN bus prevents accurate, up-to-date data being received
    - The process model receives an incorrect relative speed (e.g. sensor information) and fails to correlate the changing position of the object with a dangerous relative velocity

Type 1 scenario:
- Controller receives correct feedback but interprets it incorrectly or ignores it:
  - Information received: At least one sensor presents an accurate distance measurement, but it is overridden by the process model
  - How this could occur: The true statement above
    - A malfunctioning sensor yielding an incorrect value leads to the true value being overwritten, overridden, or distorted

Boex:
- Apollo incorrectly believes that the relative distance is higher than in reality

Type 2 scenario:
- Controller receives incorrect feedback/information:
  - Information received: The feedback received is insufficient to accurately determine the relative distance
  - How this could occur: The true statement above:
    - The range finding/actor tracking sensors are compromised.
    - Roof mounted optics are vulnerable to collision with a variety of unexpected obstacles, and could therefore have their optical alignment and focus compromised
    - Environmental or lead debris such as leaves or plastic bags could block or distort the images

Type 1 scenario:
- Controller receives correct feedback but interprets it incorrectly or ignores it:

https://gitlab.com/trustable/av-atpa/blob/...
UCA-6.1: Apollo does not provide brake control when relative velocity and distance to an obstacle mean that a collision is imminent

Type 1.1:
- Apollo does not provide brake control when relative velocity and distance to an obstacle mean that a collision is imminent
- The feedback does not indicate that an obstacle is in the vehicle's path

Type 2.1:
- The feedback does not indicate that an obstacle is in the vehicle's path
- An obstacle is in the vehicle's path

Feedback info:
- Inadequate vehicle speed
- Inadequate other vehicle / object velocity
- Inadequate other vehicle / object distance
- Inadequate object detection

UCA-6.2: Apollo does not provide brake control when in autonomous mode and vehicle speed exceeds limits (limits for controllability, stability, upcoming manoeuvre, speed limit, traffic flow limit, planned test limit, etc.)

Type 1.2:
- Apollo does not provide brake control when in autonomous mode and vehicle speed exceeds limits (limits for controllability, stability, upcoming manoeuvre, speed limit, traffic flow limit, planned test limit, etc.)
- The feedback does not indicate that the vehicle exceeds limits

Type 2.2:
- The feedback does not indicate that the vehicle exceeds limits
- The vehicle does exceed limits

Feedback info:
- Inadequate vehicle speed
- Inadequate speed limits

UCA-6.3: Apollo does not provide brake control when in autonomous mode, the vehicle is stationary, and vehicle path is not clear

Type 1.3:
- Apollo does not provide brake control when in autonomous mode, the vehicle is stationary, and vehicle path is not clear
- The feedback does not indicate that the vehicle is in autonomous mode, the vehicle is stationary and the vehicle path is not clear

Type 2.3:
- The feedback does not indicate that the vehicle is in autonomous mode, the vehicle is stationary and the vehicle path is not clear
- The vehicle is in autonomous mode, the vehicle is stationary and the vehicle path is not clear

Feedback info:
- Inadequate autonomous mode detection


~100% of 22 UCAs (~5x reduction)
STPA identified many vulnerabilities and unintended, designed behaviors in the product. STPA results were used to fix the system and improve the design while product in operation.
Examples of STPA Impact

✔ Unanimous Go/No-Go decision path (incremental acceptance increased over time):
  o Program management, System Integrators, Legal, Mechanical

✔ STPA scenarios -> Closed test tracks, test routes, technical req’s

✔ Test route criteria, proposed routes reviewed against UCAs

✔ Clear test start/end procedure

✔ ESTOP usage clearly defined, irrespective of who is in the rear seat (safety > marketing)

✔ Safety Actuator Monitor

✔ Identified incorrect autonomy SW behavior assumptions
  ✔ E.g. impact between v2.0 and v3.x SW

✔ Identified many actions not previously identified, such as throttle commanded with EPB activated

✔ Generated requirements: Driver training, procedures, test track, autonomy, etc.
Reflections from Codethink

• Open Source Safety
• https://gitlab.com/trustable/av-stpa
• Manage complexity
• Safety led software architecture
STPA Impact on Program

✔ STPA provided key feedback to Program Management to recognize risk, enable informed Go/No-Go decision

✔ STPA provided key feedback about market gap, triggered new products
Impact Discussion

- Program Management
  - Engineering Team:
    - System Integrators
    - Test Route Planners
    - Post-drive reviewers
    - Safety Engineers
  - Legal:
    - State Law, Insurance, Etc.
  - Trainers
  - Safety Driver(s)
- Autonomous Vehicle
  - Environments

© Copyright 2019 John Thomas