A Prototype for a Real-Time Weather Responsive System

Axel Krings
Ahmed Serageldin
Ahmed Abdel-Rahim

This research was supported by grant DTFH61-10-P-00123 from the Federal Highway Administration - US DoT

Integrating Clarus data into RT-App.

- Challenges
  - The Engineering Challenge
  - The Security Challenge
  - The Real-time Challenge
  - The Survivability Challenge (includes all “illities”)
- Apply the newest technology to a survivability architecture
  - Design Methodology based on Design for Survivability
Project Architecture

- A system operating in an unbounded environment
- Inheriting all problems from such environment

Clarus...

- Utilizing local sensor data to do what?
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>essPrecipSituation</td>
<td>Describes the weather situation in terms of precipitation, integer values indicate situation</td>
</tr>
<tr>
<td>essPrecipYesNo</td>
<td>Indicates whether or not moisture is detected by the sensor: (1) precip; (2) noPrecip; (3) error</td>
</tr>
<tr>
<td>essPrecipRate</td>
<td>The rainfall, or water equivalent of snow, rate</td>
</tr>
<tr>
<td>essRoadwaySnowpackDepth</td>
<td>The current depth of packed snow on the roadway surface</td>
</tr>
<tr>
<td>essAirTemperature</td>
<td>The dry-bulb temperature; instantaneous</td>
</tr>
<tr>
<td>essVisibilitySituation</td>
<td>integer value, describes the travel environment in terms of visibility</td>
</tr>
<tr>
<td>essVisibility</td>
<td>Surface visibility (distance)</td>
</tr>
<tr>
<td>essSurfaceStatus</td>
<td>integer value, a value indicating the pavement surface status</td>
</tr>
</tbody>
</table>
What could possibly go wrong?

- What assumptions should one place on a system?
  - Anything is possible!
    - and it will happen!
  - Malicious act will occur sooner or later

- It is hard or impossible to predict the behavior of an attack
Unique Opportunity

What is unique about this project?

- The application domain is part of a Critical Infrastructure
- The project is just small enough to demonstrate a survivability architecture
- The code is relatively small
- The execution is relatively deterministic
- The run-time support is relatively mature

What is Survivability

Closely related Terms

- Intrusion Tolerance
- Resilience

Relationship to

- Fault-tolerance
- Security
Design for Survivability

- When Systems become too complex
- Design by Integration of Survivability mechanisms
- Build-in *not* add-on
- Design for Survivability has surfaced in different contexts

Software Architecture

- Overview
Design Methodology

- Measurement-based design and operation

Our view of a System

- Different “machines”
  - Operations
  - Functions
  - Modules
- Epoch
  - defined by transitions
Profiles

- Frequency Spectrum (...and more)
  - count invocations
  - probability of invocation
  - defined for an epoch
  - defined for operations, functions and modules
  - does not say anything about dependencies!

Module Profile

\[ p = \langle p_1, p_2, \ldots, p_{|M|} \rangle \]

where \( p_i \) is probability that \( m_i \) is executing
Profiles

- Observed Profile

\[ \hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_{|M|}) \]

where \( \hat{p}_i = c_i / n \) is the fraction of system activity due to invocations of module \( m_i \) and \( c_i \) is the count of invocations of \( m_i \).

\[ \hat{\mathbf{p}}^k \] denotes the \( k^{th} \) observed module profile, observed over \( n \) epochs

Profiles and Certification

- System behavior

  - Analyze the observed profiles
  - What is the threshold for “normal” behavior?
  - How do we detect deviation from thresholds for “normal” executions?
  - Set the threshold of “normal” to “certified”
  - Looks like anomaly detection in IDS, or?
Profiles and Certification

- Interpretation of Certified Behavior
- If profiles are beyond the certified threshold we simply have not seen such behavior before!
- Could be benign or malicious reasons

- What is our response?
- We could simply not allow the operation to continue and go to fail-safe state

Profile Vector

- Vector \( \hat{P} = (\hat{p}_1, \hat{p}_2, ..., \hat{p}_{|M|}) \)
- notice log scale

Fig. 5. Typical observed profile of 4 costates (module IDs and frequencies on the axis)
Profile Vector & Scalar

- Observe \( h \) sequences of \( n \) epochs

- Define a centroid \( \bar{p} = (\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_{|M|}) \), where

\[
\bar{p}_i = \frac{1}{h} \sum_{j=1}^{h} \hat{p}_i^j
\]

and the distance of \( \hat{p}^k \) from centroid \( \bar{p} \) is given by

\[
d_k = \sum_{i=1}^{n} (\bar{p}_i - \hat{p}_i^k)^2
\]

Multitasking Model

- Rabbit runs Dynamic C which support costatements

Multitasking Model Diagram:

- Top of loop
- State machine
- State machine
- State machine
- State machine
- State machine
Dynamic C, costates and yield
(Figure from Dynamic C Users Manual)

Figure 5.3 shows the execution thread.

Figure 5.4 shows the execution thread with yield statement.
Profile Scalar

- Definitions based on costate $\alpha$:

  $\hat{p}[\alpha], \hat{p}^k[\alpha], \bar{p}[\alpha]$ and $d_k[\alpha]$

Multitasking Model

- One knows which costate is executing

- Profiles of costates are not polluted with activity from other costates

- Result is lower degree of non-determinism of execution

![Graph](https://example.com/graph.png)

Fig. 5. Typical observed profile of 4 costates (module IDs and frequencies on the axis)
Certified Behavior

The distance of the observed costate profiles $\hat{p}_k^k[\alpha]$ from $p[\alpha]$ can be used so that departure beyond it indicates non-certified behavior of costate $\alpha$. Two threshold vectors:

$$\epsilon^{max}[\alpha] = (\epsilon^{max}_1[\alpha], ..., \epsilon^{max}_{|M|}[\alpha])$$ (3)

$$\epsilon^{min}[\alpha] = (\epsilon^{min}_1[\alpha], ..., \epsilon^{min}_{|M|}[\alpha])$$ (4)

where $\epsilon^{max}_i[\alpha]$ and $\epsilon^{max}_i[\alpha]$ are the upper and lower threshold values of $m_i$, representing a dual-bound threshold.

Certified Behavior

Every observed profile that is in the region between the two vectors is assumed nominal. Thus we certify a profile $\hat{p}_k^k[\alpha]$ to be a nominal profile if

$$\epsilon^{min}[\alpha] \leq \hat{p}_k^k[\alpha] \leq \epsilon^{max}[\alpha]$$ (5)

i.e., if $\epsilon^{min}_i[\alpha] \leq \hat{p}_i^k[\alpha] \leq \epsilon^{max}_i[\alpha]$ for every $1 \leq i \leq |M|$. 
Centroid

![Graph of Centroid](image)

**IV. C**

**CONCLUSIONS**
Dependencies

- Relationship between Operations, Functionalities, and Modules

Mappings in \((O \times F \times M)\)

\[ G^O = (O, \prec^O) \]
\[ G^F = (F, \prec^F) \]
\[ G^M = (M, \prec^M) \]
Operations & Costates

Figure 3: Costates and Operations

Current Status

- Contingency Management Description:
  A. Serageldin, A. Krings, and A. Abdel-Rahim, “A Survivable Critical Infrastructure Control Application”, 8th Annual Cyber Security and Information Intelligence Research Workshop, Oct. 30 Sept. 2 2012, ORNL

- Gaining Experience: prototype started running 24/7
  - What is more effective: vector or scalar profiling/certification?
  - Mature in setting thresholds. Need more experimentation to get more experience
  - Looking forward to include 2012 Winter Data in real time
  - Dealing with realities of Internet access in Intersection
Conclusions

- Prototype is up and running
  - uses real-time weather data to modify traffic signal timing within safety standard
- Utilization of Design for Survivability
  - Off-nominal executions detected (dual-bound thresholds)
  - Violation of dependencies detected
  - Contingency Management to Recover from anomalies
- Future potential: apply concept to other applications, e.g., Dedicated Short Range Communication (DSRC) apps

Questions
Design for Analyzability

- Not a new concept
- e.g., Series-Parallel RBD
  - Not all systems are Series-Parallel!

Fault Models: The world in which we live/operate

- All Faults
  - Malicious
    - Asymmetric
      - Transmissive Asymmetric
      - Strictly Omissive Asymmetric
    - Symmetric
      - Omissive Symmetric
      - Transmissive Symmetric
  - Benign
    - Benign