A Resilient Real-Time Traffic Control System: Software Behavior Monitoring and Adaptation

Axel Krings
Ahmed Serageldin
Ahmed Abdel-Rahim

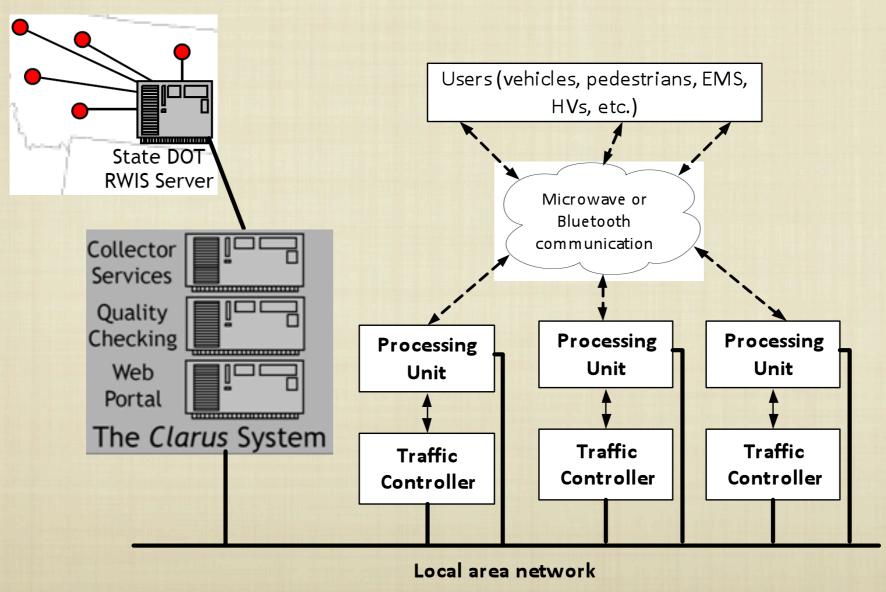
This research was supported by grant DTFH61-10-P- 00123 from the Federal Highway Administration - US DoT

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

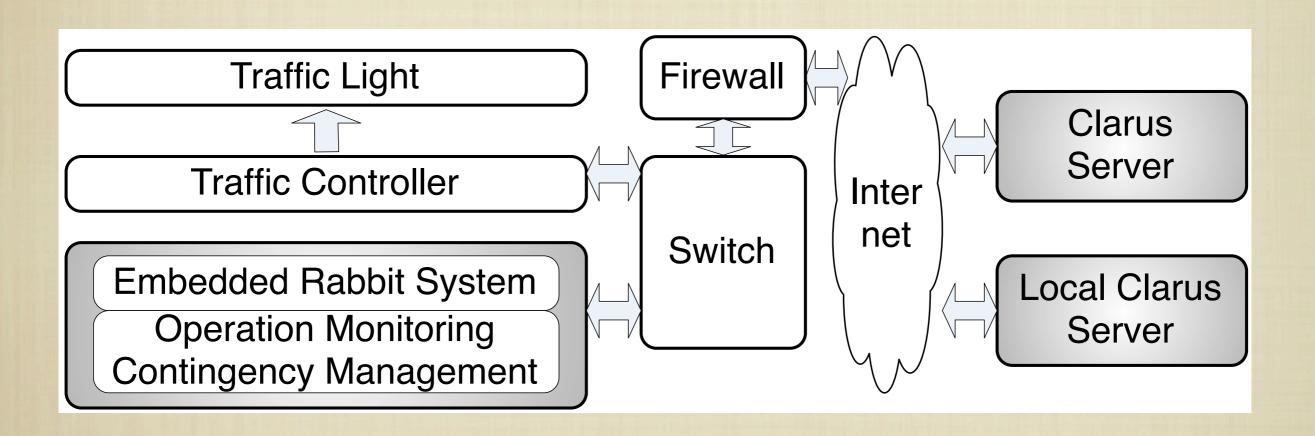
- A system operating in an unbounded environment
- Inheriting all problems from such environment



The big picture

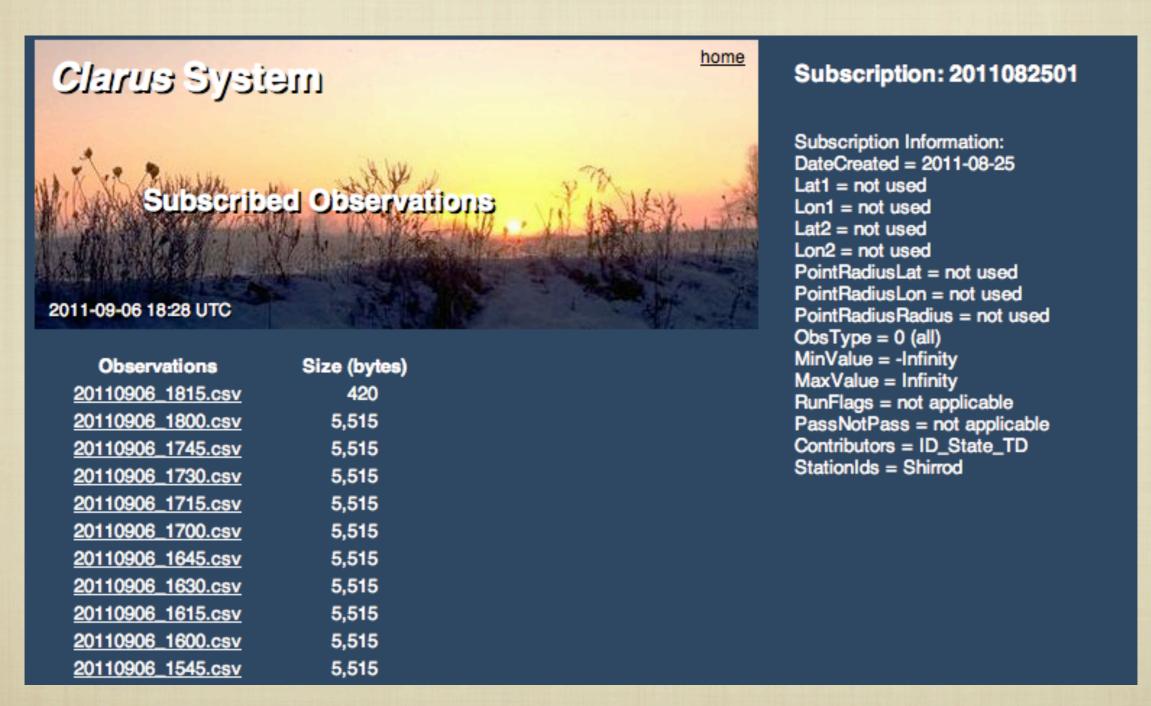
The problem:

Should we connect the control network to the Internet?



Clarus...

Utilizing local sensor data to do what?



Clarus Subscription Data

Access Clarus data files from the web



Highly Critical (Essential) Clarus Data

essPrecipSituation	Describes the weather situation in terms of precipitation, integer values indicate situation
essPrecipYesNo	Indicates whether or not moisture is detected by the sensor: (1) precip; (2) noPrecip; (3) error
essPrecipRate	The rainfall, or water equivalent of snow, rate
essRoadwaySnowpackDepth	The current depth of packed snow on the roadway surface
essAirTemperature	The dry-bulb temperature; instantaneous
essVisibilitySituation	integer value, describes the travel environment in terms of visibility
essVisibility	Surface visibility (distance)
essSurfaceStatus	integer value, a value indicating the pavement surface status

PROTOTYPE



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



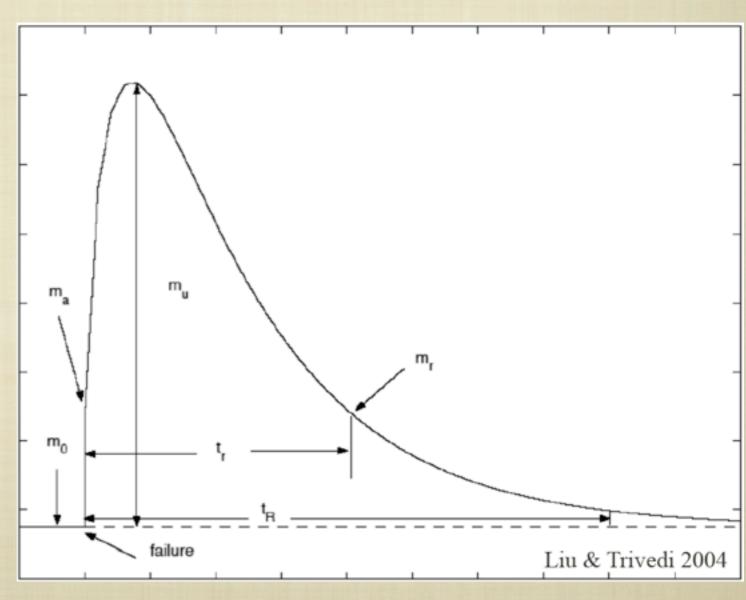
■ It is hard or impossible to predict the behavior of an attack

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

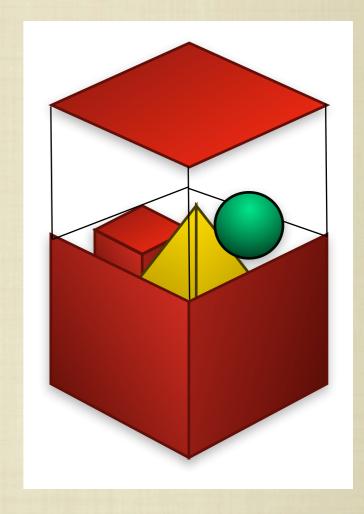
What is Survivability

- Closely related Terms
 - Intrusion Tolerance
 - Resilience
- Relationship to
 - Fault-tolerance
 - Security



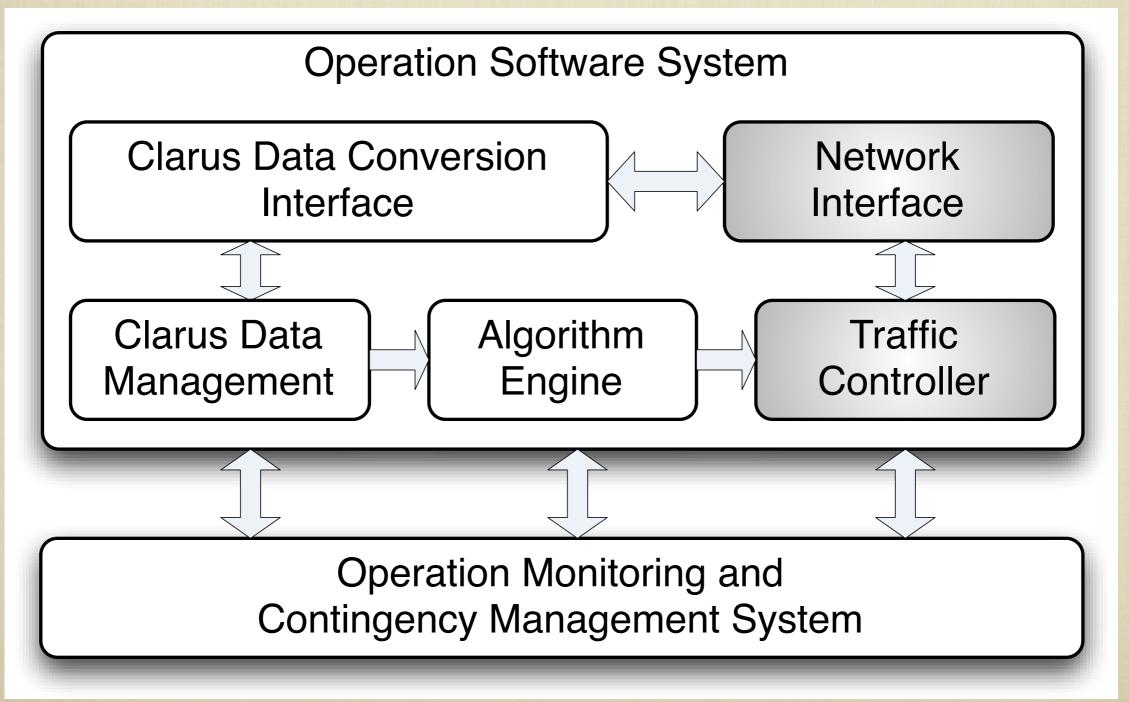
Design for Survivability

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



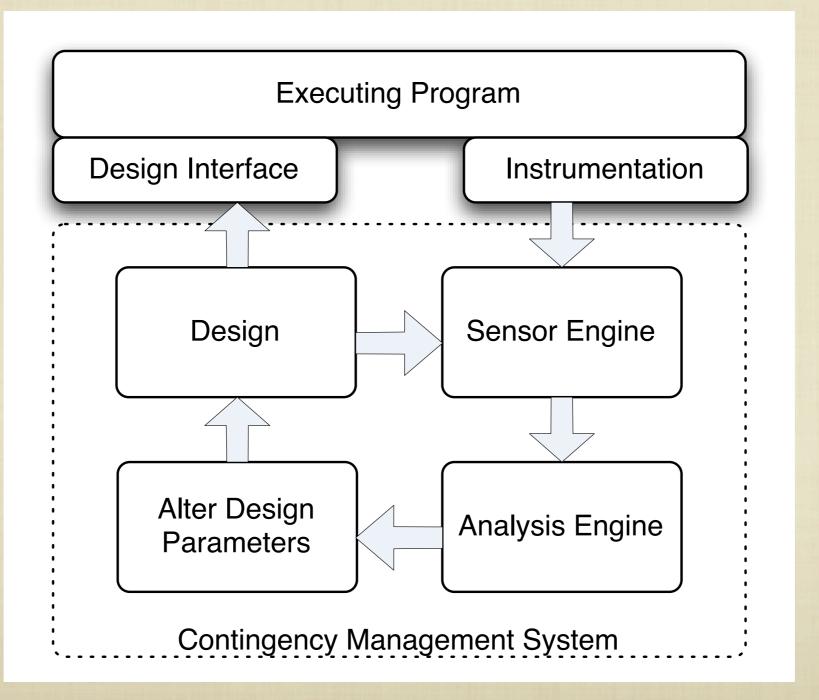
Software Architecture

Overview



Design Methodology

Measurement-based design and operation

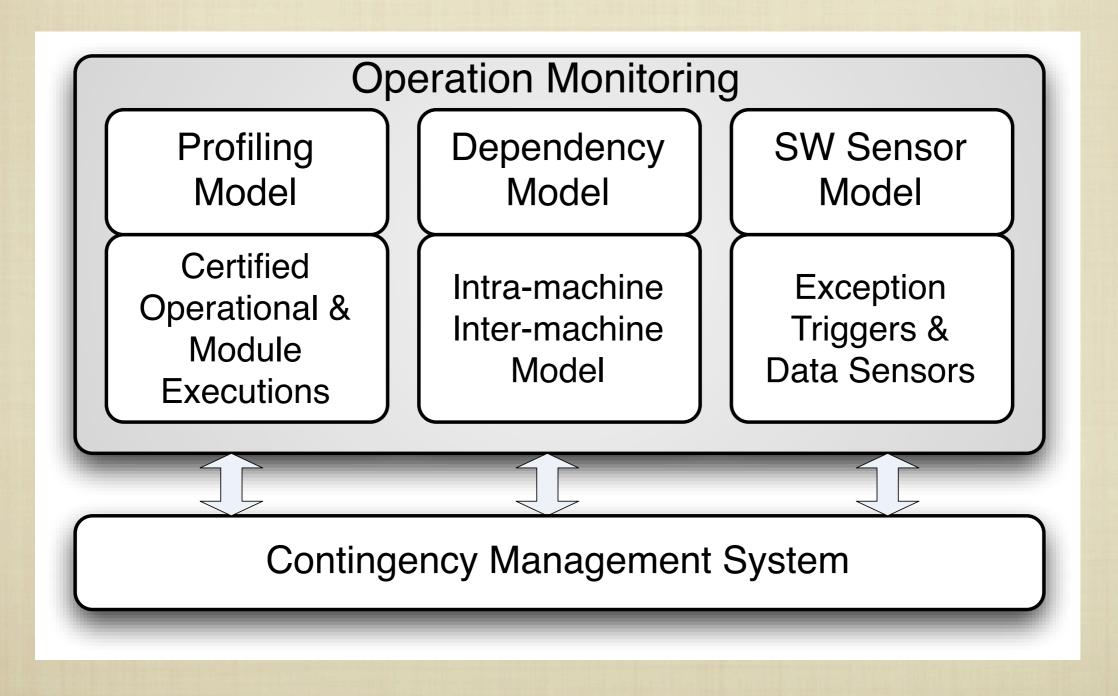


Our view of a System

- Different "machines"
 - Operations
 - Functions
 - Modules
- Epoch
 - defined by transitions

Formal Model of Sys. Arch.

Measurement-based design and operation



University of Idaho

Profiling Model

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!

Profiles

- Module Profile
 - $\mathbf{p} = \langle p_1, p_2, ..., p_{|M|} \rangle$

where p_i is probability that m_i is executing

Profiles

Observed Profile

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

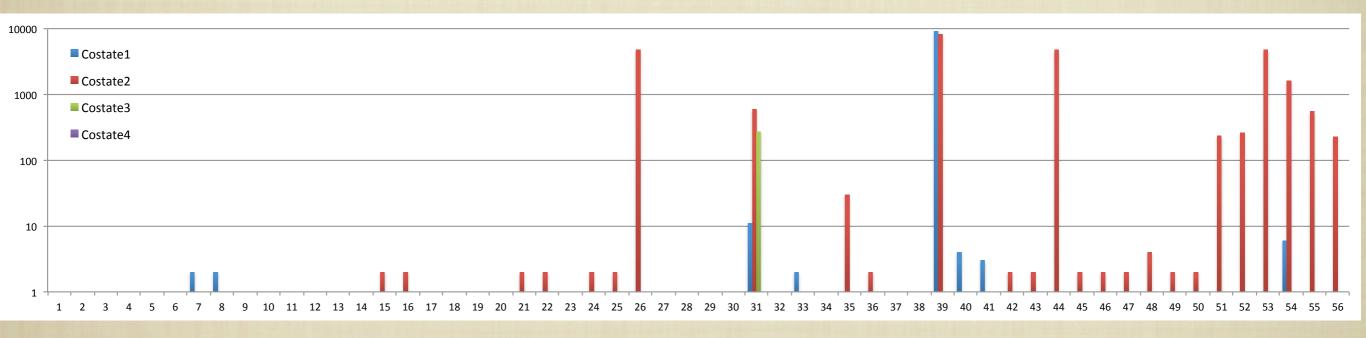


Fig. 5. Typical observed profile of 4 costates (module IDs and frequencies on the axis)

Profile Vector & Scalar

- Observe h sequences of n epochs
- Define a centroid $\overline{\mathbf{p}} = (\overline{p}_1, \overline{p}_2, ..., \overline{p}_{|M|})$, where

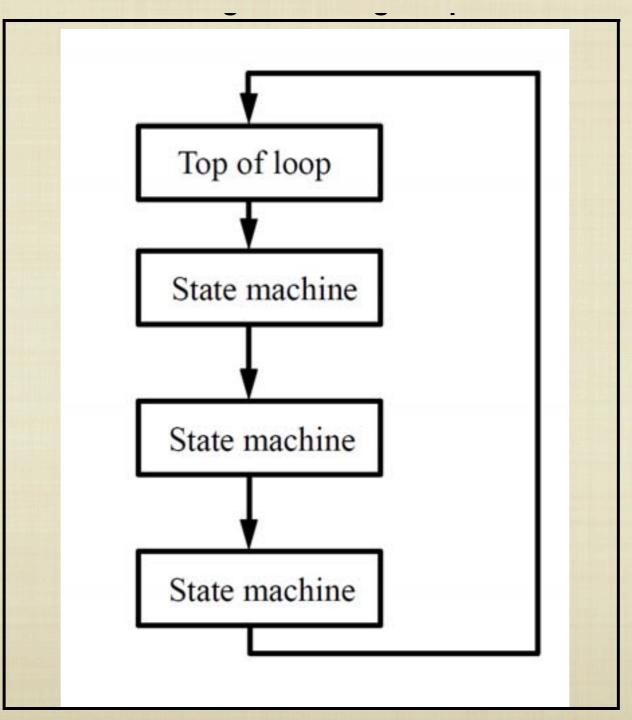
$$\overline{p}_i = \frac{1}{h} \sum_{j=1}^h \hat{p}_i^j$$

and the distance of $\hat{\mathbf{p}}^k$ from centroid $\overline{\mathbf{p}}$ is given by

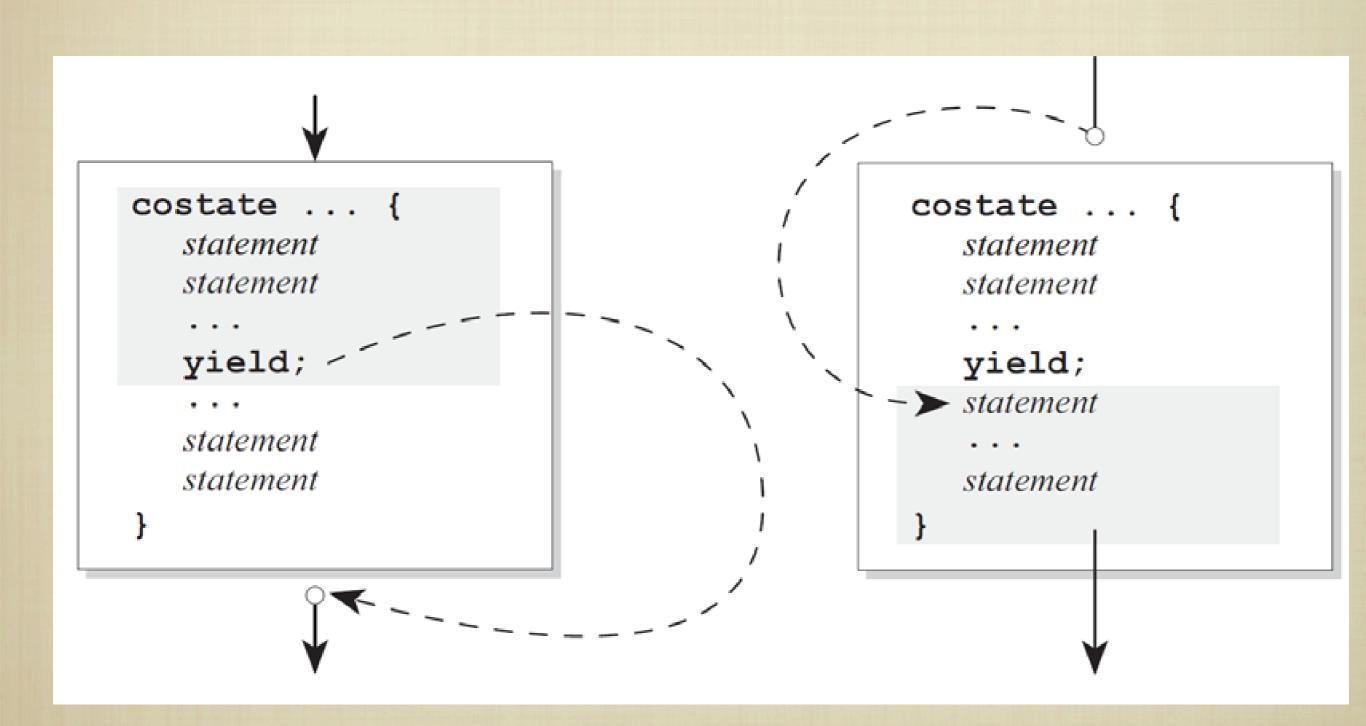
$$d_k = \sum_{i=1}^n (\overline{p}_i - \hat{p}_i^k)^2$$

Multitasking Model

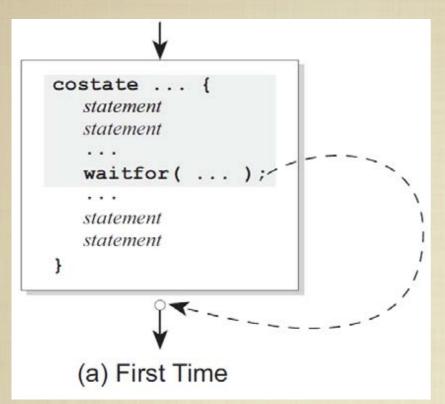
Rabbit runs Dynamic C which support costatements

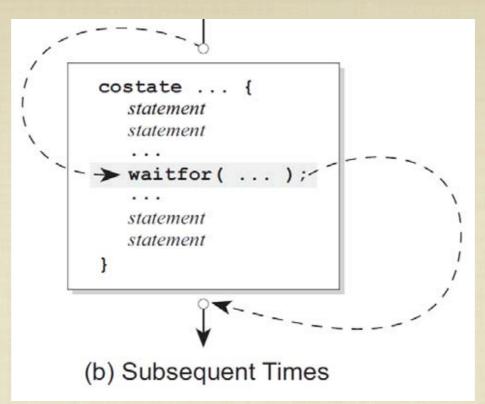


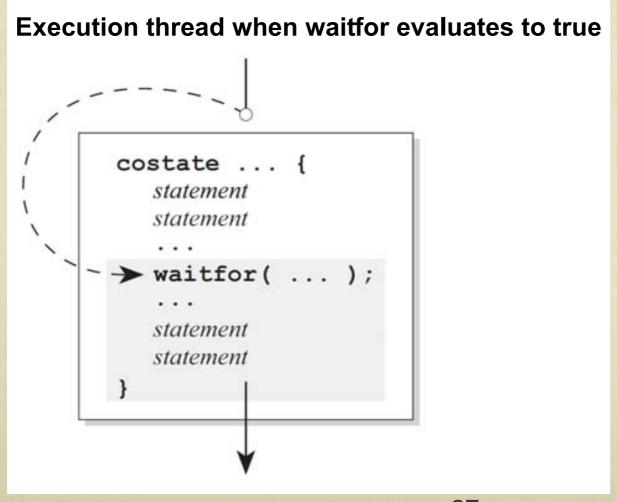
Dynamic C, costates and yield (Figure from Dynamic C Users Manual)



Dynamic C, costate and waitfor (Figure from Dynamic C Users Manual)







Profiles considering costates

Definitions based on costate α :

$$\hat{\mathbf{p}}[\alpha], \, \hat{\mathbf{p}}^k[\alpha], \, \overline{\mathbf{p}}[\alpha] \text{ and } d_k[\alpha]$$

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

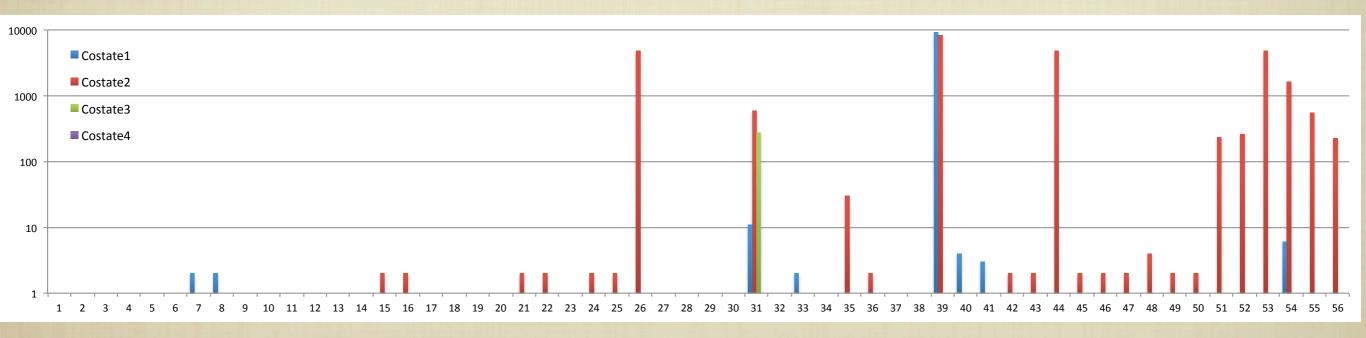


Fig. 5. Typical observed profile of 4 costates (module IDs and frequencies on the axis)

Certified Behavior

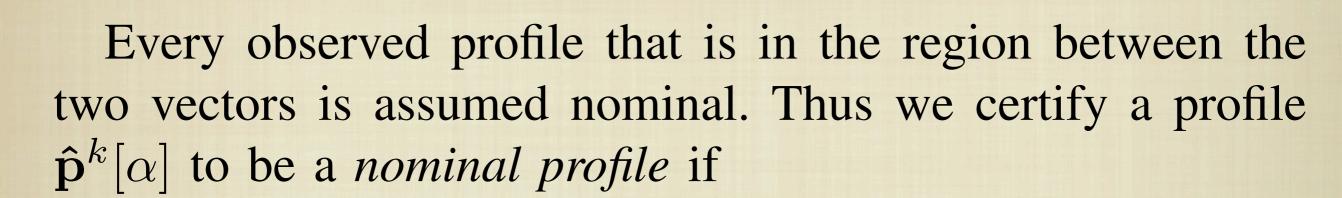
The distance of the observed costate profiles $\hat{\mathbf{p}}^k[\alpha]$ from $\overline{\mathbf{p}}[\alpha]$ can be used so that departure beyond it indicates non-certified behavior of costate α . Two threshold vectors:

$$\epsilon^{max}[\alpha] = (\epsilon_1^{max}[\alpha], ..., \epsilon_{|M|}^{max}[\alpha]) \tag{3}$$

$$\epsilon^{min}[\alpha] = (\epsilon_1^{min}[\alpha], ..., \epsilon_{|M|}^{min}[\alpha]) \tag{4}$$

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

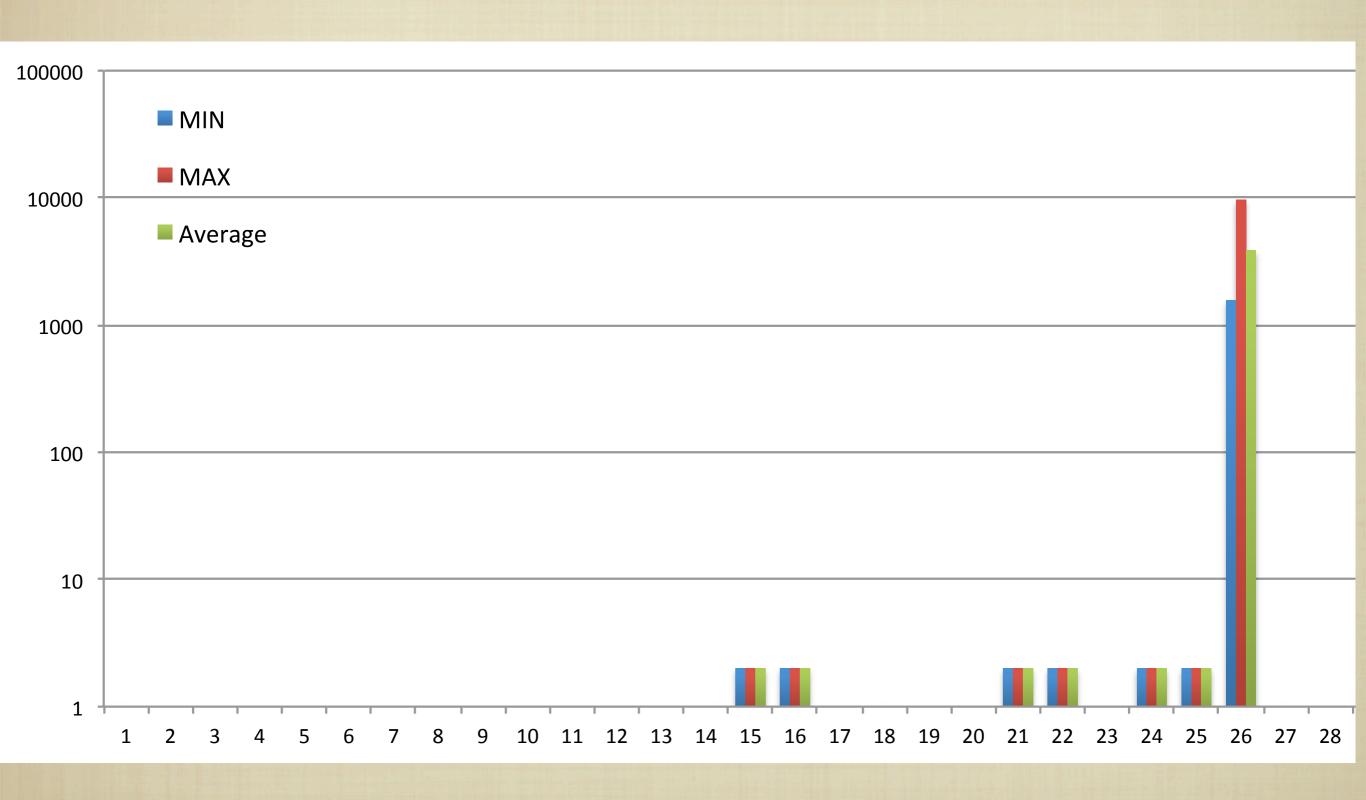
Certified Behavior



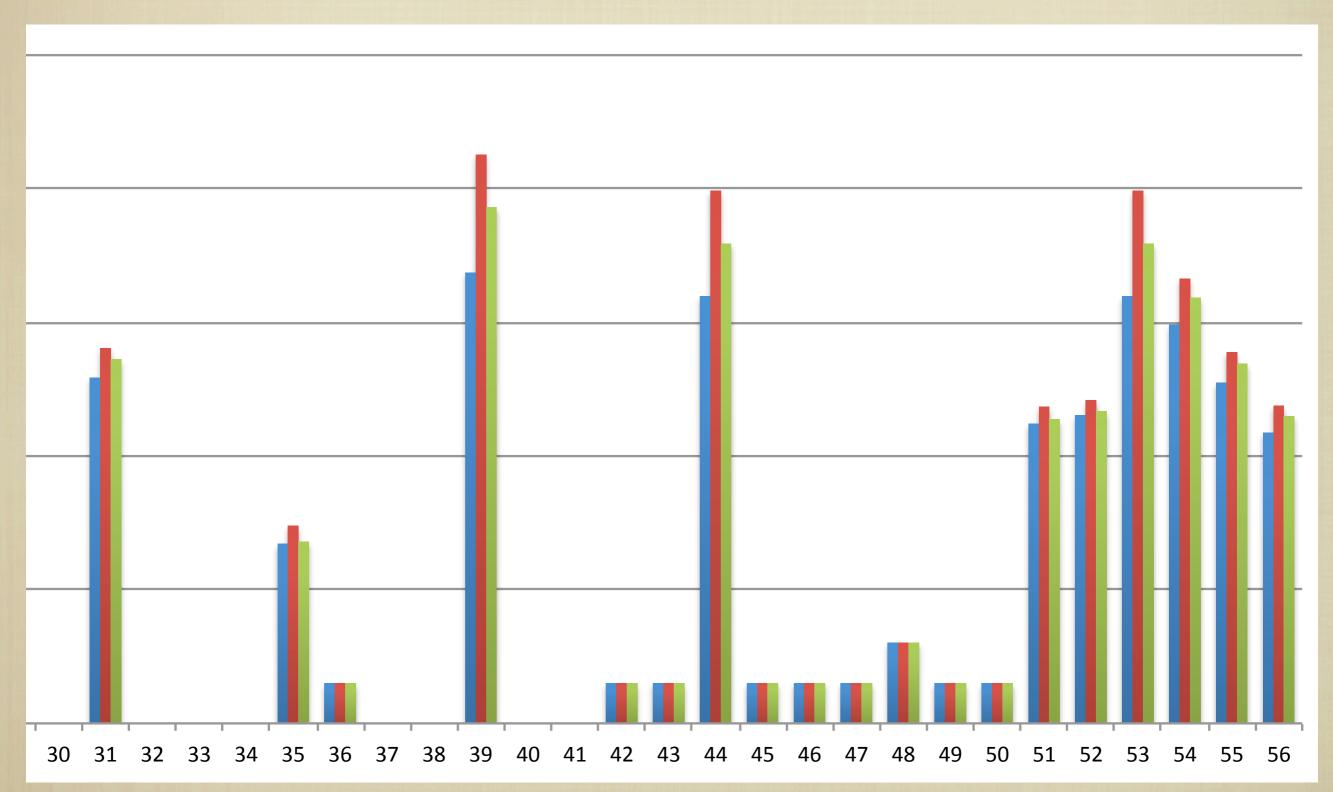
$$\epsilon^{min}[\alpha] \le \hat{\mathbf{p}}^k[\alpha] \le \epsilon^{max}[\alpha]$$

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

Centroid



Centroid



Synchronized Profiling

So fare we assumed that there is only one single behavior. However, there could be multiple.

Considering h sequences of n epochs each, we define a centroid of sets $\overline{\mathbf{P}} = (\overline{P}_1, \overline{P}_2, ..., \overline{P}_{|M|})$, where

$$\overline{P}_r = \overline{P}_r \cup p_i, \quad 1 \le r \le |M| \quad p_i = \frac{1}{h} \sum_{j=1}^h \hat{p}_i^j \tag{2}$$

for each behavior i. Thus $\overline{\mathbf{P}}$ is a |M|-dimensional structure of sets, and again using the above financial metaphor, each element represents the "h-day moving average" of a specific set of stocks (module), where a day is measured as n epochs, and again we want to track the past in order to establish "nominal", i.e., expected, behavior from a set of behaviors.

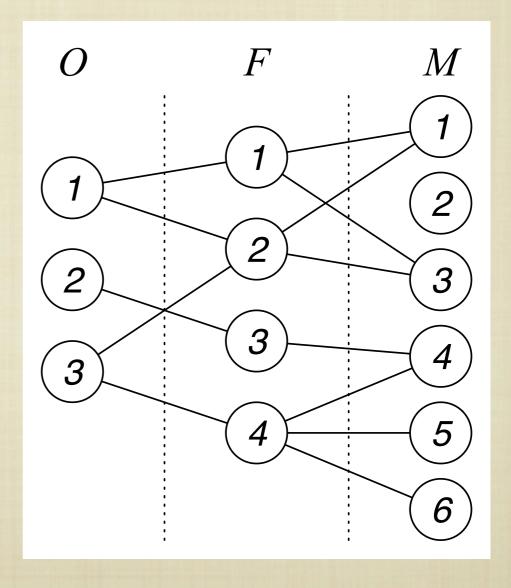
University of Idaho

Dependency-based Model

Inter-dependencies

Relationship between Operations, Functionalities, and Modules

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



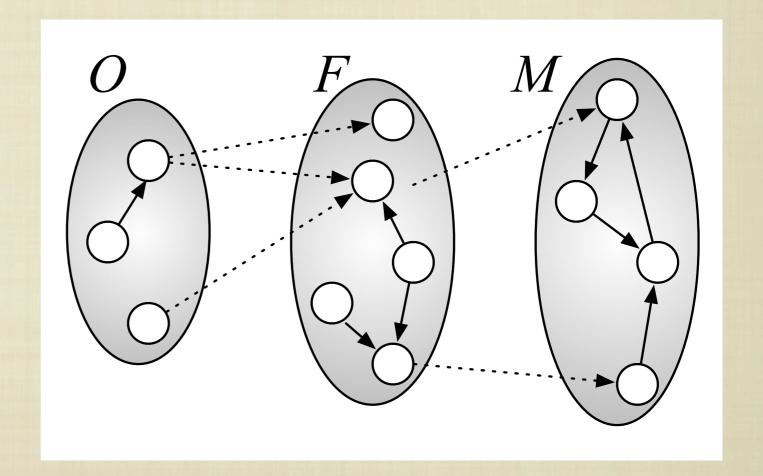
Intra-dependencies

Relationship within Operations, Functionalities, and Modules

$$\mathcal{G}^O = (O, \prec^O)$$

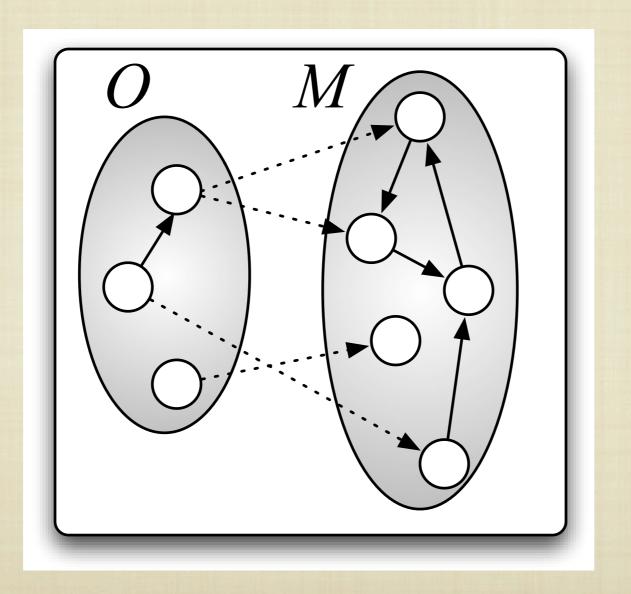
$$\mathcal{G}^F = (F, \prec^F)$$

$$\mathcal{G}^M = (M, \prec^M)$$

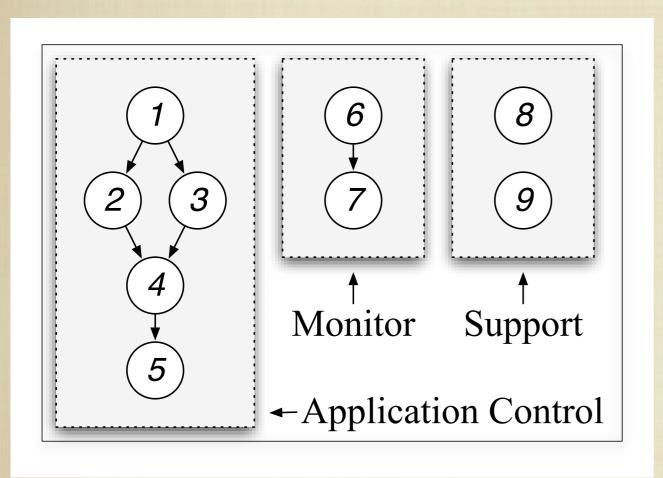


Intra-dependencies

In our current system we simplify to



Operations & Costates



- 1 Get Clarus data
- 2 Receive data from LCS
- 3 Receive data from Clarus
- 4 Analyze Clarus data
- 5 Adjust controller
- 6 Monitor analysis
- 7 Monitor adaptive reconfiguration
- 8 Time synchronization
- 9 Support routines

Figure 3: Costates and Operations

Sensor-based Model

Sensor-based Model

- Not every behavior can be extracted from profiles or dependencies.
- Specific data sensors are needed to observe specific data values or trigger exceptions.

Exception Triggers

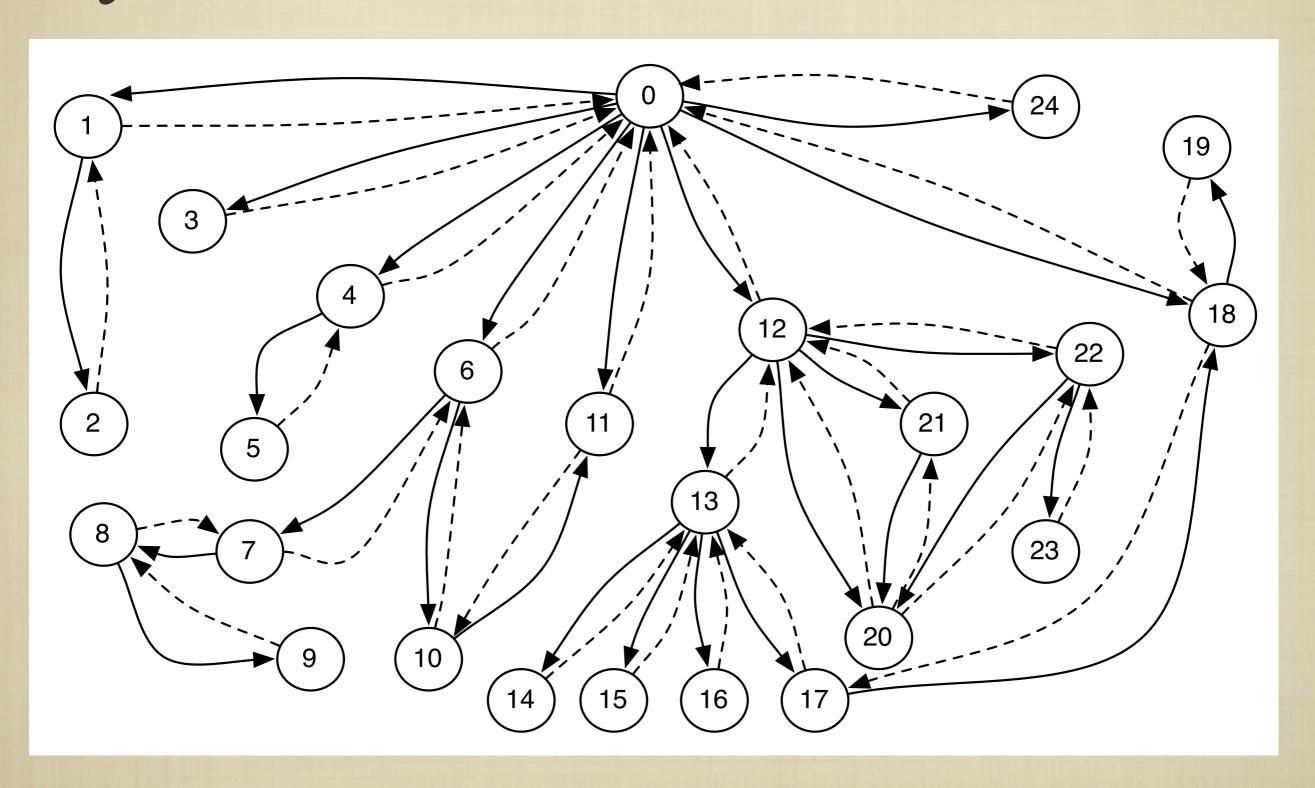
- Exception trigger array
 - identify and profile exceptions, e.g., file does not exist, specific sensor data is not longer available.
 - any error condition can be viewed as an exception trigger

Data Sensors

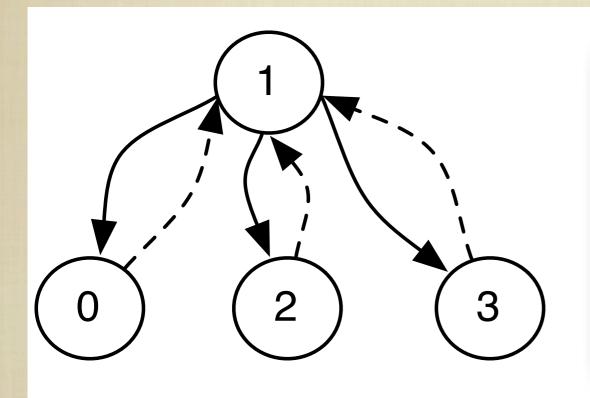
- Observation of specific numeric values for analysis
- Example: the adjustment to the yellow timing
- What happens when someone changes to yellow time to zero? Is that possible?

System Operation & Contingency Management

System Module State Machine



System Operations State Machine



Operations:

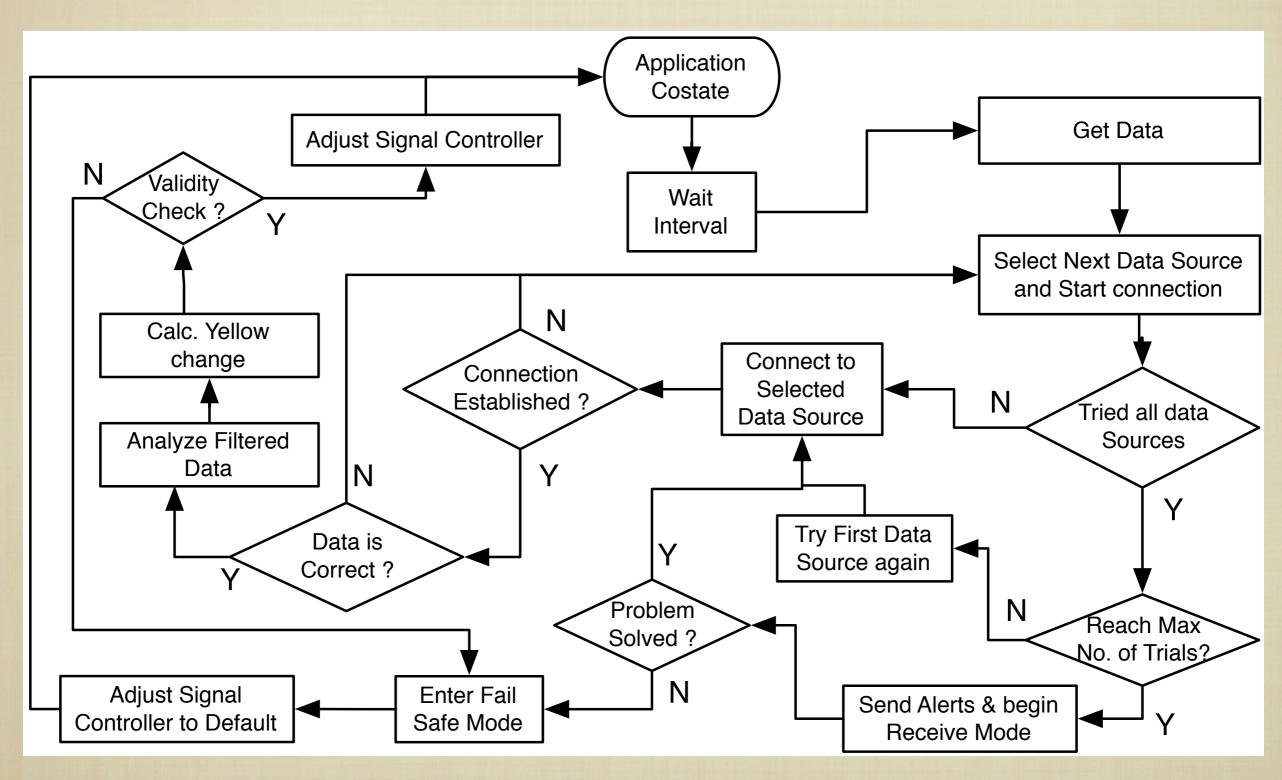
0: Initialize Program

1 : Runtime Timing Module

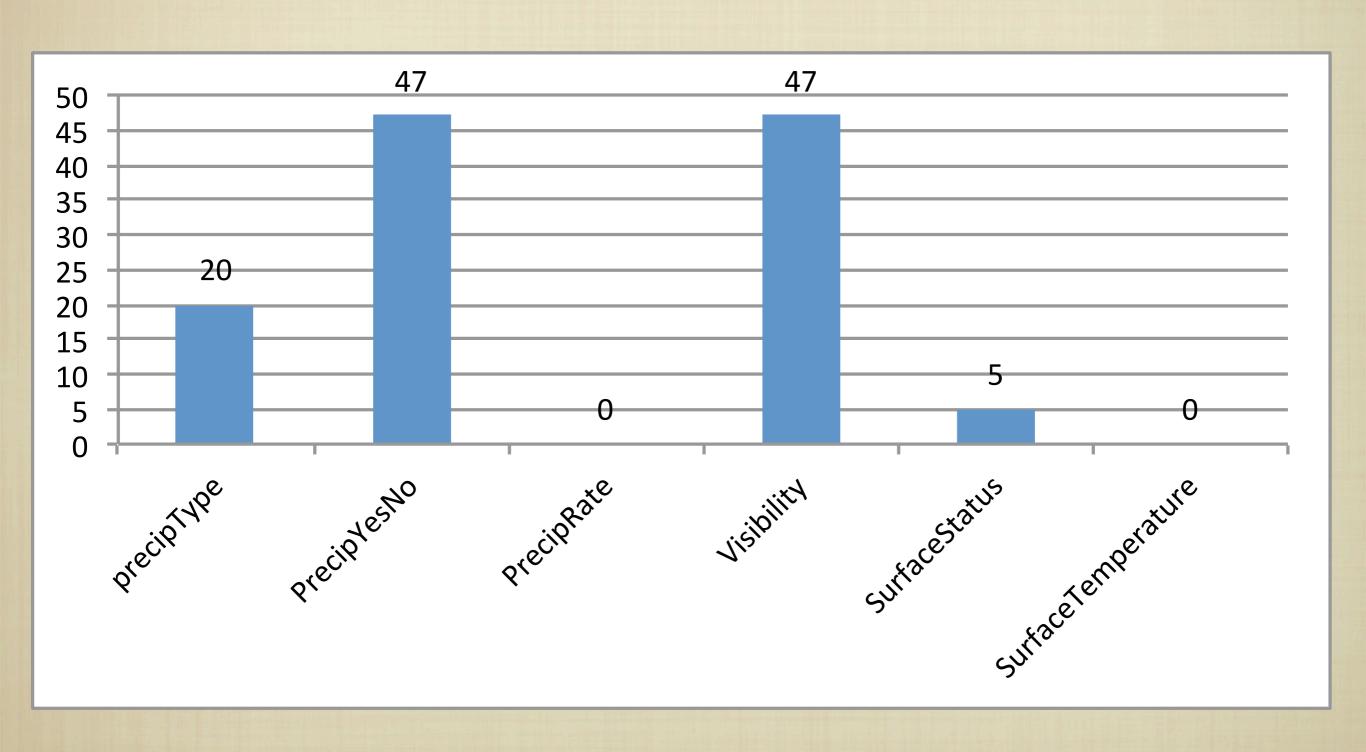
2: Get Weather Data

3: Update Controller

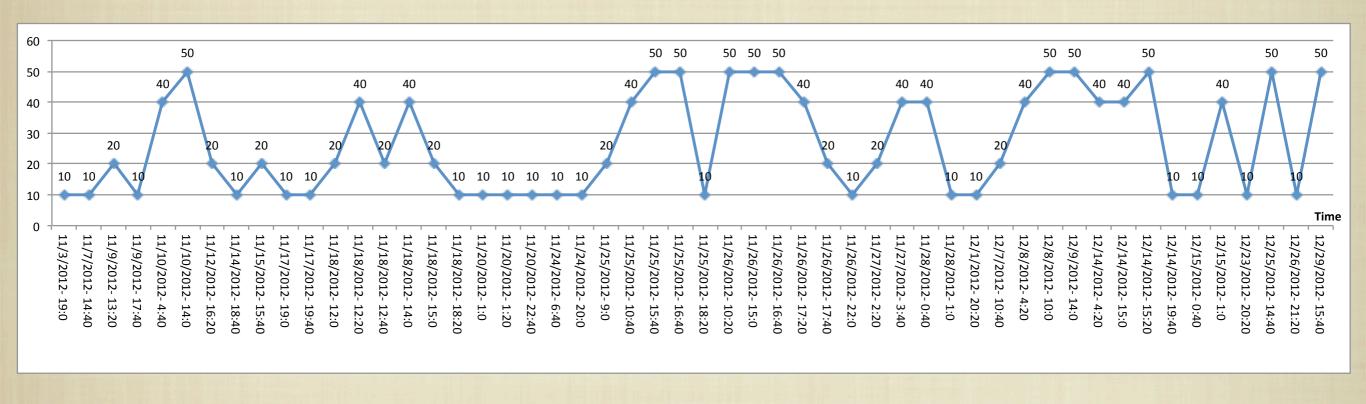
Application Control Costatement



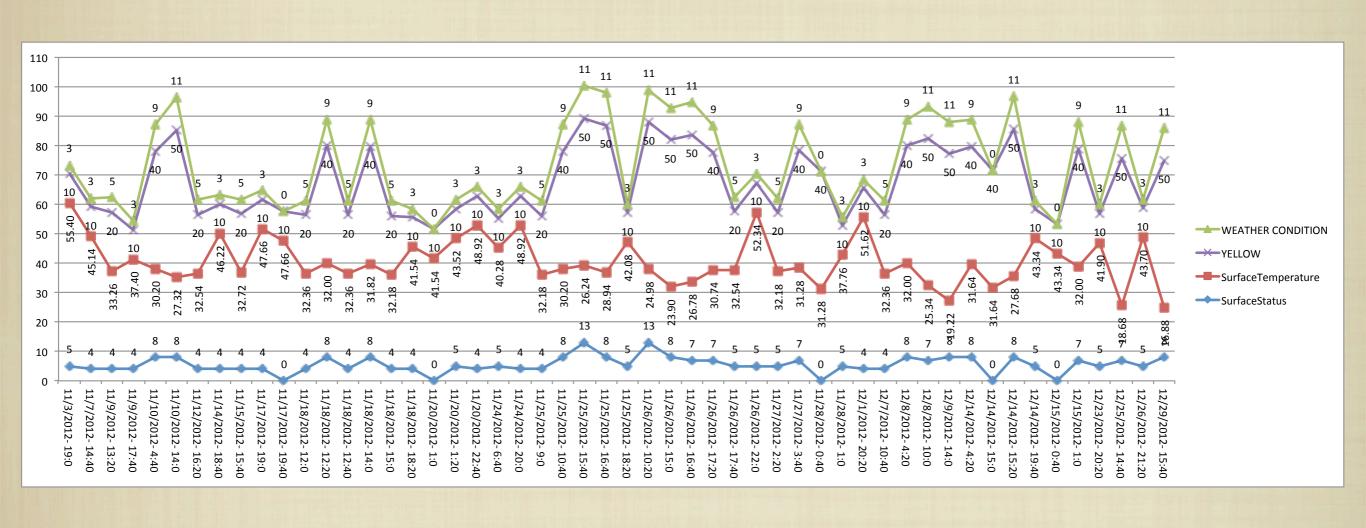
Exception Triggers



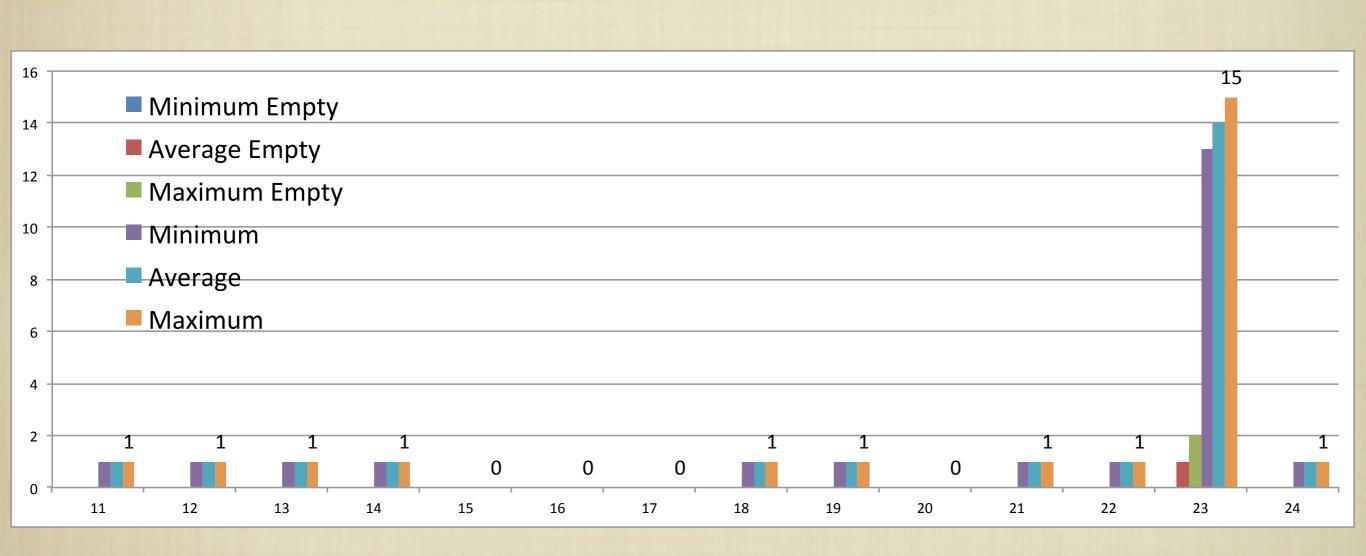
Yellow adjustment in % over winter months



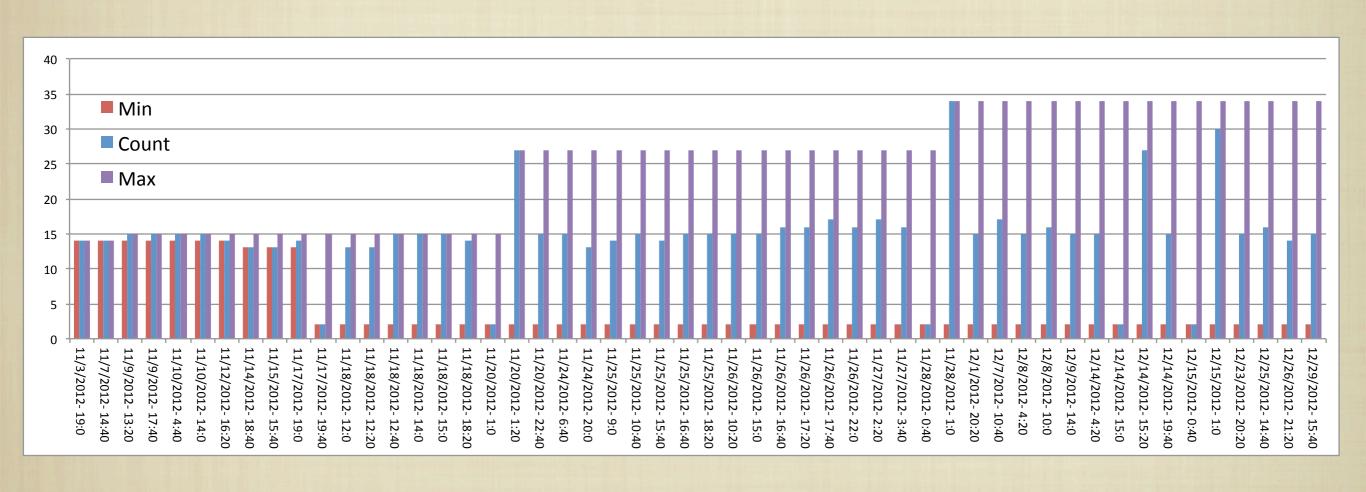
Yellow adjustment in % over winter months



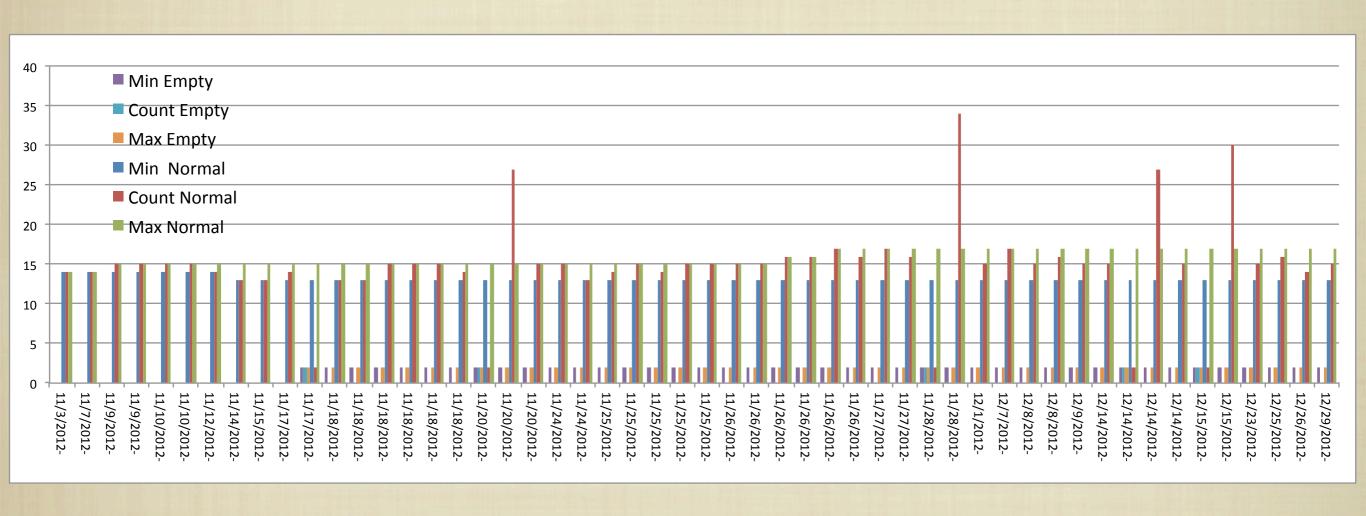
Profiles of key modules and two nominal behaviors



Profiles of module m23 with behavior set size equal 1



Profiles of module m23 with behavior set size equal 2



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

Axel Krings, Ahmed Serageldin and Ahmed Abdel-Rahim, "A Prototype for a Real-Time Weather Responsive System", in Proc. Intelligent Transportation Systems Conference, ITSC2012, Anchorage, Alaska, 16-19 September, pp. 1465 - 1470, 2012.

- Gaining Experience: prototype started running 24/7
 - Mature in setting thresholds.
 - Dealing with realities of Internet access in Intersection

Conclusions

- Prototype has been running over 1 year
 - 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