A Survivable Architecture for Real-Time Weather Responsive Systems

Technical Approach

Axel Krings
Ahmed Abdel-Rahim

DTFH61-10-P-00123
National Institute of Advanced Transportation Technology
University of Idaho
Project Overview

- Develop a prototype of a real-time weather-responsive traffic signal control system to improve the efficiency and safety of traffic signal operations during inclement weather.

- The system receives weather information from the FHWA’s Clarus system database, analyzes it, and makes necessary changes to signal timing parameters in response to inclement weather conditions.

- The system will operate and achieve its potential using current traffic controller and controller cabinet technologies. Minimal hardware, in addition to traffic controllers, are required.

- The system will be compatible with future applications within the connected-vehicle initiative.

- Software design addresses survivability concerns.
Potential Safety and Operational Benefits

- Increase the value of yellow and all-red interval and coordination offset values for each weather condition
- Based on microscopic-simulation (VISSIM) and using surrogate safety measures:
  - 46% reduction in vehicles in dilemma zone
  - 34% reduction in conflicts
  - 19% increase in corridor throughput
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
Prototype
Clarus...

Utilizing local sensor data to do what?
Clarus Subscription Data

Access Clarus data files from the web

http://www.clarus-system.com/SubShowObs.jsp?subId=2011082501&file=20110906_2200.csv
### Highly Critical (Essential) Clarus Data

<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>
<tr>
<td><strong>Highly Critical (Essential) Clarus Data</strong></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>essSurfaceTemperature</td>
<td>The current pavement surface temperature</td>
</tr>
<tr>
<td>windSensorGustSpeed</td>
<td>The maximum wind gust recorded by the wind sensor during the 10 minutes preceding the observation</td>
</tr>
<tr>
<td>essSnowfallAccumRate</td>
<td>The snowfall accumulation rate</td>
</tr>
<tr>
<td>essIceThickness</td>
<td>Indicates the thickness of the ice on surface</td>
</tr>
<tr>
<td>essPrecipitationStartTime</td>
<td>The time at which the most recent precipitation event began</td>
</tr>
<tr>
<td>essPrecipitationEndTime</td>
<td>The time at which the most recently completed precipitation event ended</td>
</tr>
<tr>
<td>essMobileFriction</td>
<td>Indicates measured coefficient of friction</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>windSensorAvgSpeed</td>
<td>A two-minute average of the windspeed</td>
</tr>
<tr>
<td>essPrecipitationOneHour</td>
<td>The total water equivalent precipitation over the one hour preceding the observation</td>
</tr>
<tr>
<td>essSurfaceIceOrWaterDepth</td>
<td>The current thickness of ice or depth of water on the surface of the roadway</td>
</tr>
<tr>
<td>essSurfaceBlackIceSignal</td>
<td>integer, A value indicating if Black Ice is detected by the sensor</td>
</tr>
<tr>
<td>essPavementTemperature</td>
<td>The current pavement temperature 2-10 cm below the pavement temperature.</td>
</tr>
<tr>
<td>pavementSensorTemperatureDepth</td>
<td>The depth at which the pavement temperature is detected</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

Liu & Trivedi 2004
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
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
A Measurement-based Design and Evaluation Methodology for Embedded Control Systems

A. Krings, V. Balogun, S. Alshomrani A. Abdel-Rahim, M. Dixon

Computer Science Department
Civil Engineering
University of Idaho
University of Idaho
Moscow, ID 83844-1010
Moscow, ID 83844-1022

ABSTRACT

A measurement-based design and evaluation methodology for embedded control systems is presented that features better control of non-deterministic executions through reduction of non-determinism, certified behavior of executions, real-time monitoring of operations, functionalities, and modules, which allows adaptation to non-certified behavior.

Using principles of Design for Survivability, the software system is broken down into costatements with low degree of non-determinism in their executions, thereby allowing a significant increase in the accuracy of runtime profiling. This in turn results in better detection of deviation from certified executions. The formal model to achieve these goals is introduced and the effectiveness is demonstrated in a real control application.

1. INTRODUCTION

As the components controlling our critical infrastructures are increasingly relying on networked computing systems, this connectivity also becomes the focal point for security and survivability considerations. It is thus more important than ever to include security and survivability starting at the specification and design stage, rather than in an addition fashion.

1.1 Application

The application for which the design and evaluation methodology was developed is a traffic control application, however, the general principles described in the paper are application-independent and hold for a wide range of control applications. Our application considers traffic signals as part of a networked Intelligent Transportation System. A database server called Clarus, which resulted from the Clarus Initial Road Weather Management Program, in conjunction with the Intelligent Transportation Systems Joint Program, collects real-time sensor data about weather conditions of a large number of sensor stations maintained by most states in the US.

The data from the Clarus server is now accessed and used by traffic controllers in order to adjust the timing of traffic lights to compensate for the impact of adverse local weather conditions in order to improve safety and thus reduce accidents.

The system consists of the traffic controller, everything found in a modern traffic light-controlled intersection, and a Rabbit-based system referred to as “Rabbit” in the rest of the paper, that accesses the Clarus system or a Local Clarus Server, which is a local mirror used for redundancy and scalability purposes, via the Internet.

1.2 Contributions

Clarus Data Conversion Interface
Network Interface

Clarus Data Management
Algorithm Engine
Traffic Controller

Operation Monitoring and Contingency Management System

Figure 1: Software Architecture Overview
Design Methodology

- Measurement-based design and operation

Contingency Management System

Executing Program

Design Interface

Design

Alter Design Parameters

Instrumentation

Sensor Engine

Analysis Engine

Contingency Management System
Our view of a System

- Different “machines”
  - Operations
  - Functions
  - Modules

- Epoch
  - defined by transitions
Profiles

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

Module Profiles of Costates

Figure 4: Sample Profiles
Dependencies

- Relationship between Operations, Functionalities, and Modules

![Diagram](image.png)

**Figure 2: Mappings in \((O \times F \times M)\)**
Dependencies cont.

- Operations $G^O = (O, <)$
- Functionalities $G^F = (F, <)$
- Modules $G^M = (M, <)$
Operations & Costates

Figure 3: Costates and Operations

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
Certificate executions

- Certified profiles
- Based on profiles
- Costate profiles reduce non-determinism

For each costate: If we consider \( m \) sequences of \( n \) epochs each, we can define a costate centroid \( \overline{u} = \langle \overline{u}_1, \overline{u}_2, \ldots, \overline{u}_{|O|} \rangle \) where

\[
\overline{u}_i = \frac{1}{m} \sum_{j=1}^{m} \hat{u}_i^j
\]

and the distance from \( \hat{u}^k \) from centroid \( \overline{u} \) is given by

\[
d_k = \sum_{i=1}^{n} (\overline{u}_i - \hat{u}^k_i)^2
\]
Conclusions

- Unique opportunity to apply new Design Methodology
  - Real-time Control Application
  - Utilize Design for Survivability
  - Allows for integration of key features necessary for CI
  - Derivation of real-time self-monitoring via Instrumentation

- Future potential
  - Apply the concept to other applications
System Demonstration
Questions