Carnegie Mellon

RoSES: Robust Self-configuring Embedded Systems

RoSES

Phil Koopman, Charles Shelton, Beth Latronico, Jen Morris

Carnegie Mellon University
HH D-202
Pittsburgh, PA 15213
USA
koopman@cmu.edu
www.ices.cmu.edu/roses
ph: +1 412/268-5225
fax: +1 412/268-6353

Affiliated with:

Institute for Complex Engineered Systems

Institute for Complex Engineered Systems
http://www.ices.cmu.edu

Electrical & Computer ENGINEERING

Electrical & Computer Engineering Department
http://www.ece.cmu.edu

Institute for Software Research, International
http://www.isri.cs.cmu.edu

Sponsored by:

General Motors Collaborative Laboratory at Carnegie Mellon

Bombardier Transportation

Bosch Corp.

The RoSES project seeks a general approach to building flexible, robust, maintainable and logistically supportable distributed embedded systems. Our emphasis is on achieving *graceful degradation.*

A PRODUCT FAMILY POINT OF VIEW

As highly distributed computers are embedded into everyday products, it is becoming vital to ensure that systems degrade gracefully rather than break due to software brittleness. For example, the failure of a single component shouldn't cause an entire system to fail, but instead cause a modest reduction in capabilities in keeping with the severity of the component failure.

Graceful degradation has traditionally been viewed as a fault tolerance issue. Redundant components and fail-over code are the usual approaches to managing failures. These approaches can be made to work, but are too expensive for many applications, and tend to scale poorly as the complexity of a system increases.

Instead, we view graceful degradation as a novel application of product family architecture approaches. If a product contains dozens or hundreds of processors contained within distributed sensors, actuators, and other components, each different member of a product family can be specified to be the system built from a particular configuration of those components. Adding or removing a component is simply a way of creating a different member of that same product family. Building a gracefully degrading system therefore becomes equivalent to building a system that can configure itself to be a particular member of a product family in the field rather than in the factory. Thus, the graceful degradation problem is transformed from one of fault tolerance into one of field reconfiguration.

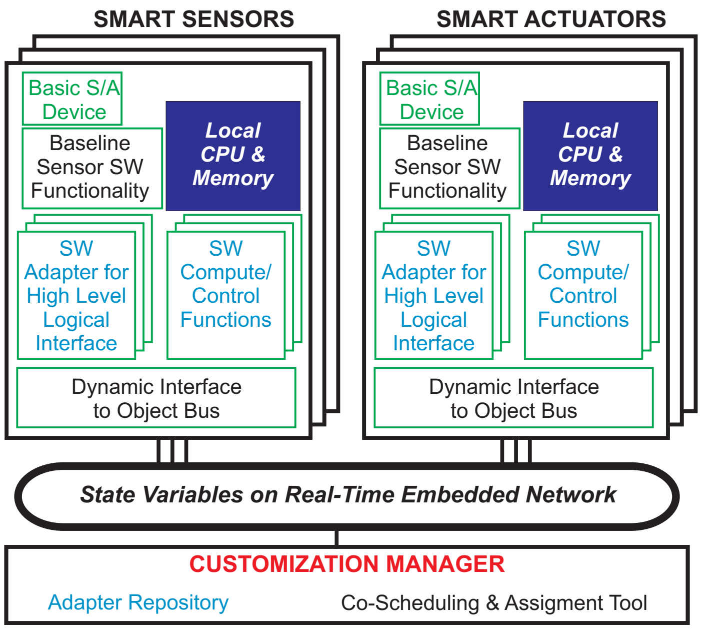
The RoSES architecture performs self-configuration using a customization manager that can be built into the system or provided as a detachable service tool. The customization manager installs desired system features by matching software adapters and algorithms to the available hardware resources. Each smart sensor or actuator is provided with software adapters that transform a basic component interface to one customized to the application (for example, providing shaft RPM instead of raw pulse counts from an optical shaft encoder). The customization manager ensures that the maximum possible functionality is provided, consistent with meeting real-time deadlines on processors and the embedded network.

SYSTEM-LEVEL BENEFITS

Every time a hardware component is added, fails, is upgraded, or is replaced by a slightly different component during maintenance, the customization manager is used to reconfigure the system for optimal performance. This use of software adapters and customization is a powerful approach that brings many benefits, including:

- *Graceful upgrade by adding components.* The system automatically recustomizes itself to exploit newly added component capabilities. Thus new hardware technology or algorithms can be exploited via new component installation.
- *Graceful degrade when components fail.* A component failure is equivalent to "downgrade" via component removal.
- *Graceful repair with approximately correct spares* in lieu of exact replacement components. The system automatically recustomizes to accommodate similar, but inexact, repair parts. Thus, repairs are simply field upgrades or downgrades depending on the capabilities provided by the repair part.
- *Reduced need for legacy spares.* As technology improves, new components can be installed in place of old ones during maintenance, reducing legacy spare inventory costs.

A prototype configuration manager has been constructed. We are currently working on architectural approaches and a run-time infrastructure for our testbed.


SMART SENSORS
SMART ACTUATORS
Basic S/A Device
Baseline Sensor SW Functionality
Local CPU & Memory
SW Adapter for High Level Logical Interface
SW Compute/ Control Functions
Dynamic Interface to Object Bus
State Variables on Real-Time Embedded Network
CUSTOMIZATION MANAGER
Adapter Repository
Co-Scheduling & Assigment Tool

Oct. 2002