SRG AutoHAN

AutoHAN/Oxygen/Pebbles Project

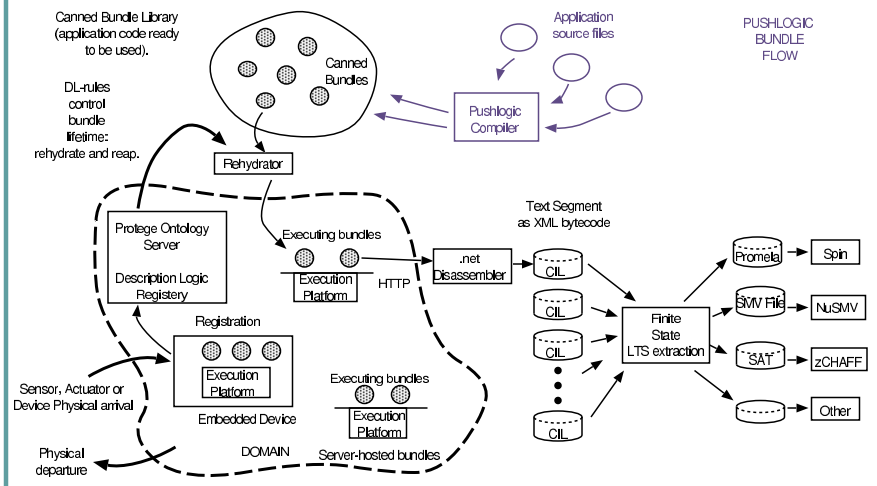
A Toolchain for Avoiding Feature Interaction

Rapid Proof (using Model Checking) as Systems Combine and Split.
System = Collection of Controllers, Actuators, Sensors and Plant Model.
Applications = Written in an experimental language called Pushlogic.

The Cambridge-MIT Institute

UNIVERSITY OF CAMBRIDGE

Computer Laboratory
David Greaves et al.



- o DOMAIN contains controllers, actuators, sensors and plant models.
- o Domain contents are dynamic, as applications are triggered and devices come and go.
- o Domains may be merged (two sets of train carriages become a longer train, but this button still opens ALL the doors)
- o Wish to avoid lengthy binding and proof sessions as domain structure evolves : precompute as much as possible.
- o Description-logic ontology for triggering changes in amount of provides a two-level system, using different tools to handle dynamic and finite-state systems.

General Research Areas

- o Feature Interaction and Scripting for Ubiquitous Devices
- o Enhanced toolchain for automotive and industrial control.
- o Eternal System Middleware
- o Language Primitives for Passive Tuple Space Data Models
- o Application Digests for rapid formal checking on system composition.

Work In Progress

- o Defining an architecture for interaction of ubiquitous computing devices or embedded controllers that includes REFLECTION for both control APIs and proactive users of those APIs (application programs).
- o Exploring the practical limitations of a scripting language (called Pushlogic) that is constrained to fully support automated reasoning.
- o Developing application digests that facilitate fast automatic reasoning for temporal logic, saftey and liveness properties.

- o A number of software and hardware demonstrators have been built.
- o The CD/DVD player conforms to our AutoHAN component architecture.
- o There are separate PEBBLES for mechanism, display, keypad, timer, ...
- o Ethernet allows full remote description, control and inspection of each Pebble.

- o There is a pre-bound Pushlogic code bundle in the device ROM for out of the box operation using conventional rear-panel connectors.
- o A dynamically-bound code bundle directs media to output devices over the network. This must be checked for feature interaction before being allowed to run.
- o The code bundles are exported in .net or pushlogic byte code in XML format over HTTP.
- o Next step is support of application digests for rapid model checking of larger domains.