Objectives

Reliable and cost-effective engineering of Feedback Control Systems is difficult:

1. Finding the right control model

2. Implementing/Integrating this model into the target system

Model Driven Approach

Find Control Model

Model Architecture

- Identify relevant touchpoints: sensors and effectors
- Reuse existing elements or create new ones
- Add required controllers
- Model data and control flow

Generate Code

- Generate skeletons for the missing elements
- Generate necessary plumbing code
- Generate deployment descriptors - blueprints

Model and runtime support for engineering FCS
Uniform reusable architecture with good system level support for rapid prototyping
Adaptable, reflective and explicit elements of feedback loop
Explicit data and control flow

Case Study - Overload Control of the Workload Management System (WMS)

WMS is responsible for distributing and managing jobs across grid resources. It is a gateway for the gLite middleware. This scenario aims at its overload control by maintaining a certain number of jobs inside its task queue.

Every element is an Adaptive Element

- Sensor - system context observing
- Controller - decision making logic
- Effector - system alteration
- Link - connection between elements

Can be observed (state, meta-data)

Can be modified

Model Notation and Examples

Structural Model (excerpt)

Filter

Sensor

Controller

Collector

DataLink

Link

ControlLink

Simple Feedback Control Loop

Filter

Controller

Host A

Active Collectors

Passive Collectors

Effectors

Remote Data Link

Remote Control Link

Host B

Remote Links

Controller Coordination

Model Notation - examples of loops

Future Work

- Distribution aspect of the system - deployment model
- Composition and compact representation with merged elements
- Different runtimes - C++ based, SCA components
- Experiments in Condor and gLite middleware
- In longer term use the model as a base for a library of loop elements and patterns