Protective Wrapper Development: Error Recovery Handling

Mei Feng

Newcastle University

July 21, 2003

Overview

- **Background**
- **Protective wrapper**
- **Case study**
- **Further discussion**

Background

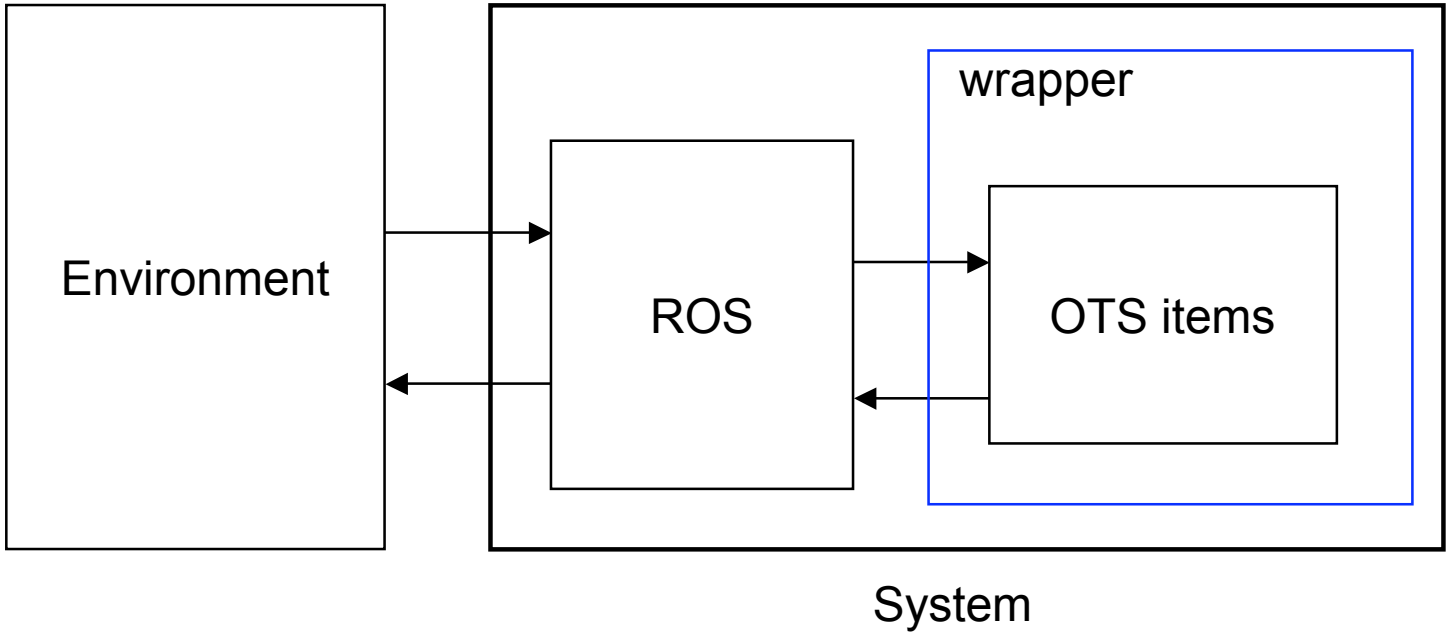
- **OTS (Off-the-Shelf) components are typically aimed at a mass-market and are often of a lower quality than bespoke components**
- **OTS components are seldom intended for the specific setting and environment in which they are employed - consequently a system in which an OTS component is integrated may misuse or misinterpret it**
- **Information about the COTS item which the system integrator has at his/her disposal is often incomplete, ambiguous or, even, erroneous**

DOTS Project

- **Use of OTS components is a source of failure**
- **So employ fault-tolerant techniques, such as protective wrapper during system integration**
- **DOTS project**

General Approach

- **OTS item**
- **Rest of System (ROS)**
- **Environment of the System**



Protective Wrapper

- **A piece of redundant, bespoke software intercepting all information going to and from the OTS item. It consists of the following functions:**
  - **Detecting errors or suspicious activities**
  - **Initiating appropriate recovery**

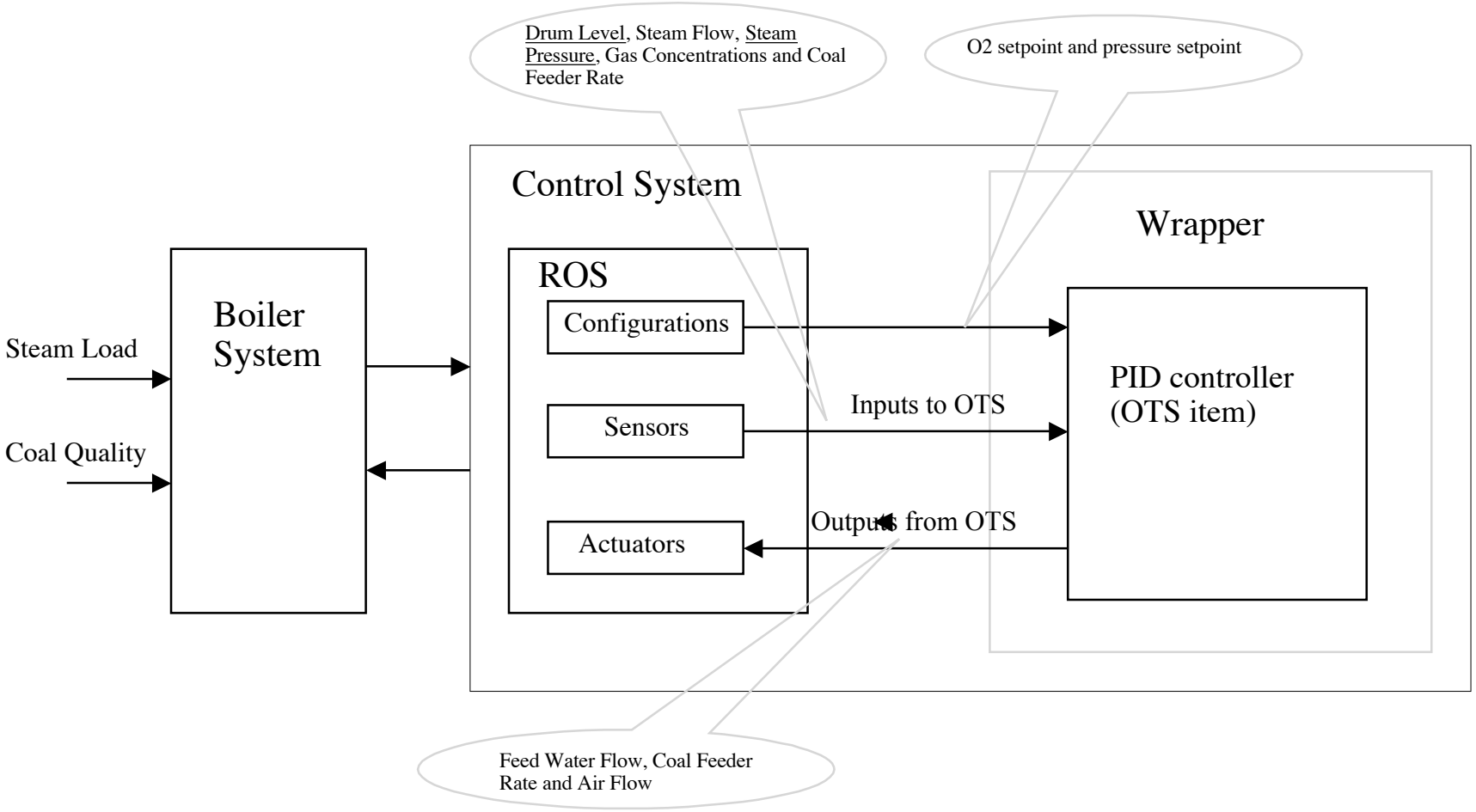
Information Required for Wrapper Development

- **Specification of the OTS item behaviour, as provided by both the item designers and the integrated system designers**
- **“Erroneous” behaviour of the OTS item, for example a known failure to react to stimuli as specified by the item designers, or behaviour which the system designer especially wants to protect against**
- **Specification of the correct behaviour of the ROS with respect to the OTS item**

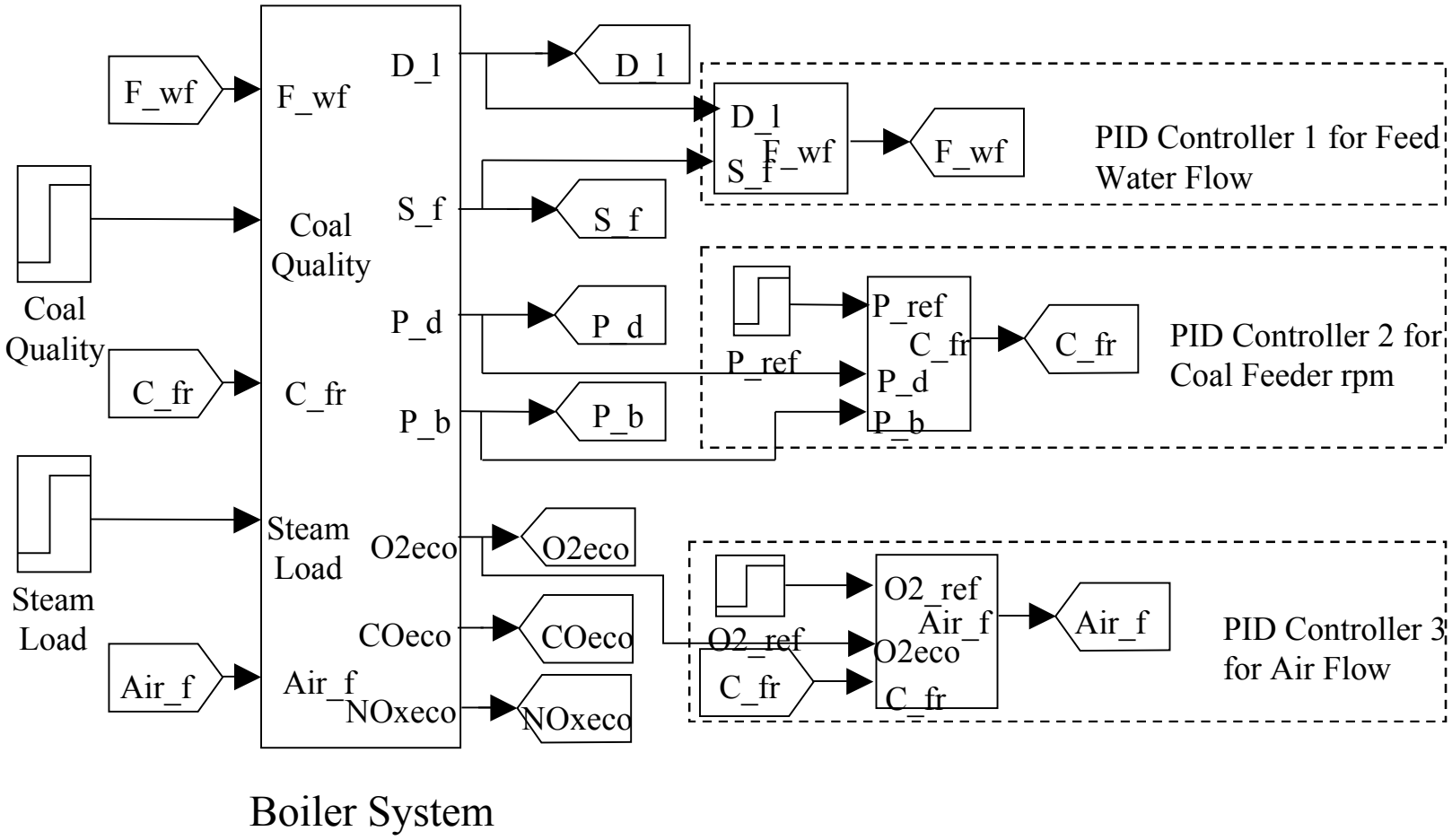
Case Study

- **Aim**
  - **Simulate erroneous behaviours**
  - **Use to design wrapper**
  - **Simulate behaviours of wrapper**
- **Platform used:**
  - **MATLAB**
  - **Simulink**
  - **Original model of boiler system and PID controller from Honeywell**

Integration of Boiler System, PID Controller and Wrapper



Simulink Model of Boiler System with PID Controller in MATLAB



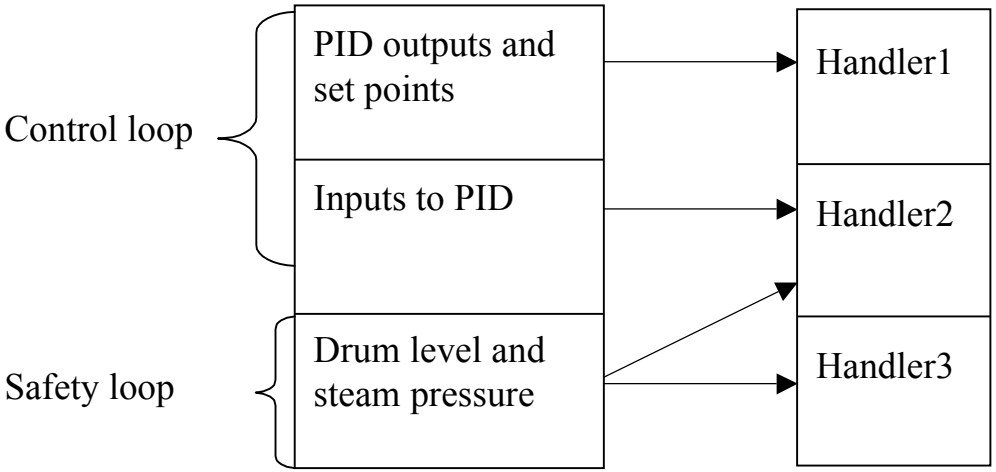
Model Variable Categories

- **Input/output:**
  - **Inputs from sensors**
  - **Set point inputs**
  - **Outputs to actuators**
- **Impact on safety:**
  - **Safety loop**
  - **Control loop**

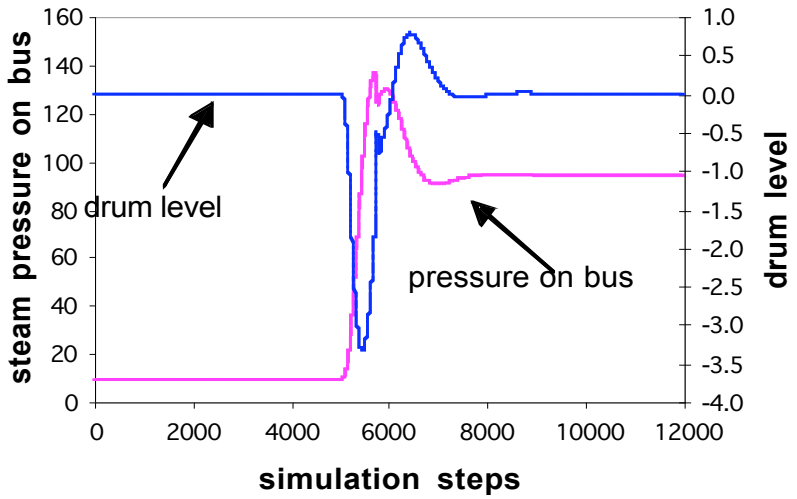
Response to Detected Errors

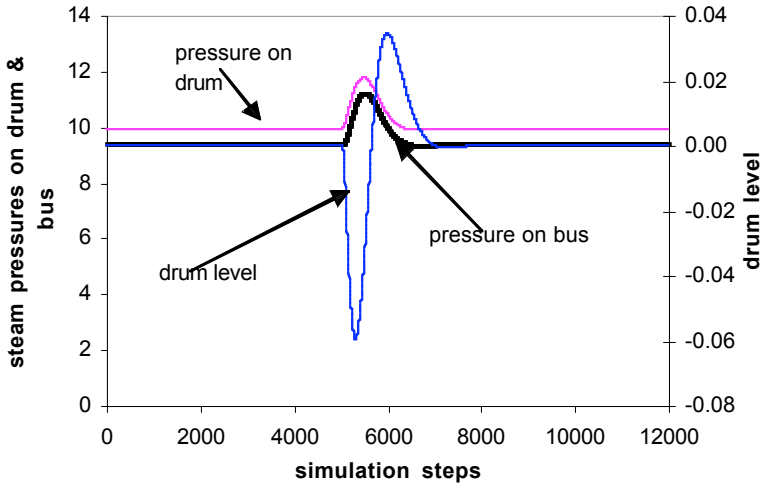
- **Forward and backward recovery**
- **Exception handling – needs to be designed**
- **Wrapper incorporating handlers**
- **Three specific handlers considered**
  - **H1: reset with alert**
  - **H2: wait ? alarm, wait ? H3**
  - **H3: shutdown and alarm**

Recovery Strategy

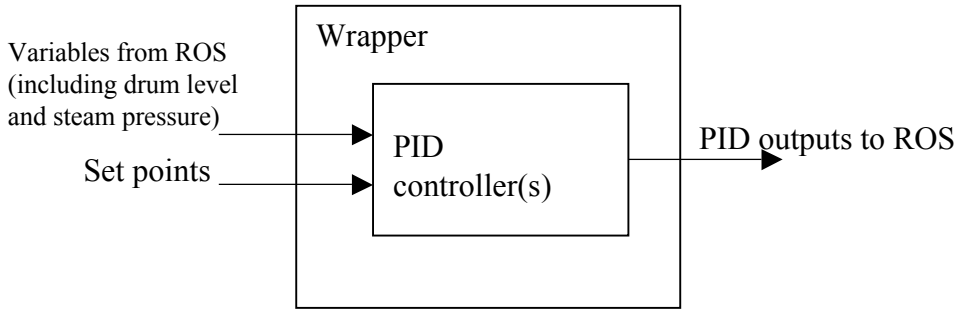


Operational Recovery – An Example





Further Discussion



- **Generic response by a wrapper for three different variable categories**
- **Injecting multiple faults and exploring the effects on the wrapper and the integrated system**