UNIVERSITY OF NEWCASTLE

Wrapping the Future

Tom Anderson, Brian Randell,
Sascha Romanovsky

University of Newcastle upon Tyne, UK

WCC2004-TD3, Toulouse, August 2004

OTS Components

- OTS (Off The Shelf) software components:
  - Are relatively cheap, because of price amortisation
  - Can come with extensive records of use
  - Are Immediately available
- But:
  - Previous use might not be representative
  - Information about their development process may be unavailable
  - They may even have to be viewed as black boxes
- So:
  - How can they be used safely in a critical bespoke software system?

The DOTS Project

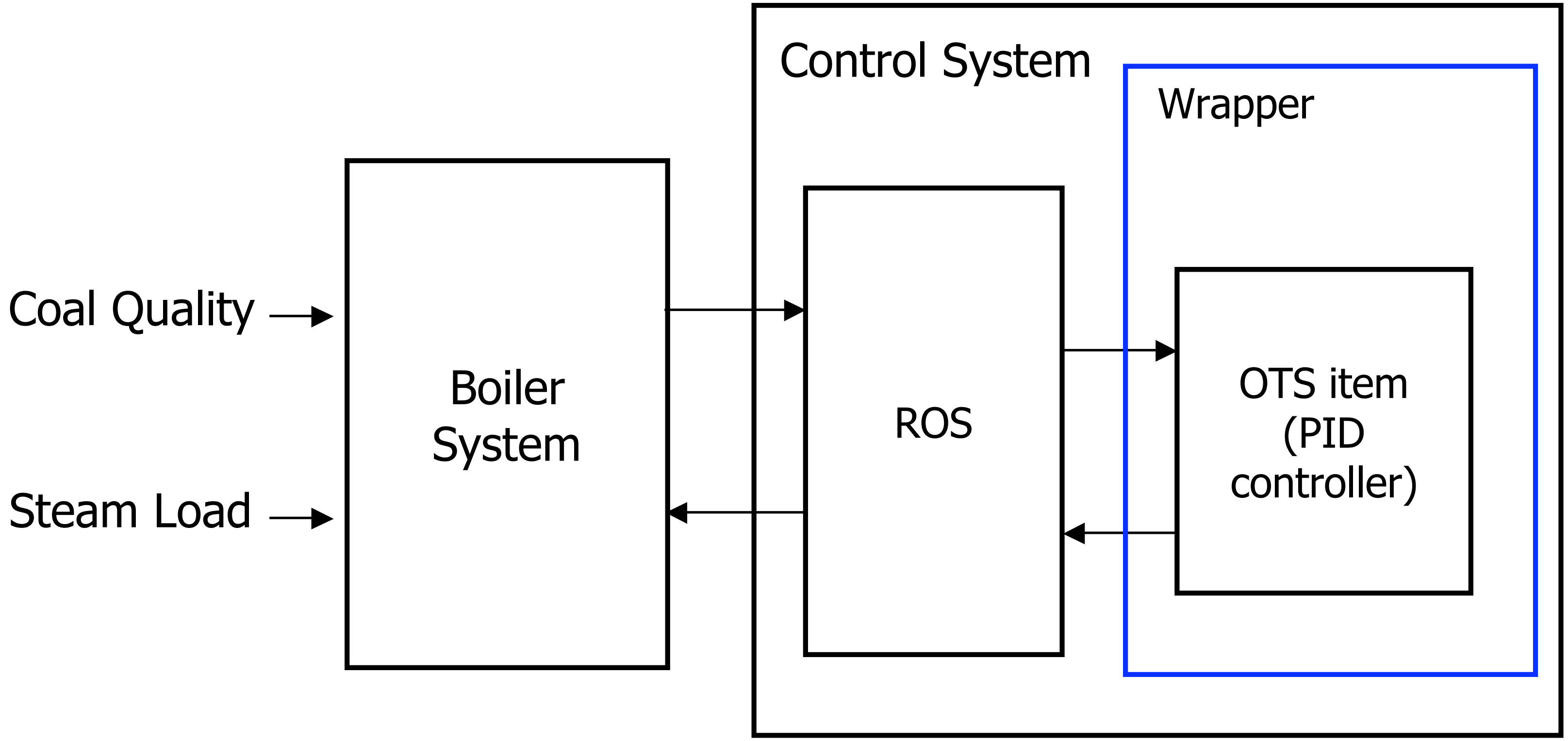
UNIVERSITY OF NEWCASTLE

- “Diversity with OTS components”:
  - A joint City/Newcastle project, sponsored by EPSRC
  - Explored an architectural approach to using untrustworthy OTS components in critical applications
  - The approach - enclosing the OTS component in a purpose-built “protective wrapper”
- Such a wrapper:
  - Intercepts all the OTS component’s inputs and outputs
  - Attempts to mask any faults resulting in errors in this I/O
- DOTS addressed the question:
  - How effective is this (simplistic yet popular) technology?

The DOTS Investigation

- Aimed at realism
  - Though not daring to use a *real* critical application!
- Via an industrial model of a RT control system
  - A Honeywell-supplied industrial grade simulation of a steam boiler and its associated control system
  - Written in Simulink
  - Represents a real steam-raising system in which a coal-fired boiler responds to demands for steam
  - The automated control system incorporates a PID (i.e. Proportional, Integral and Derivative) controller
- The PID is the chosen OTS component to be wrapped

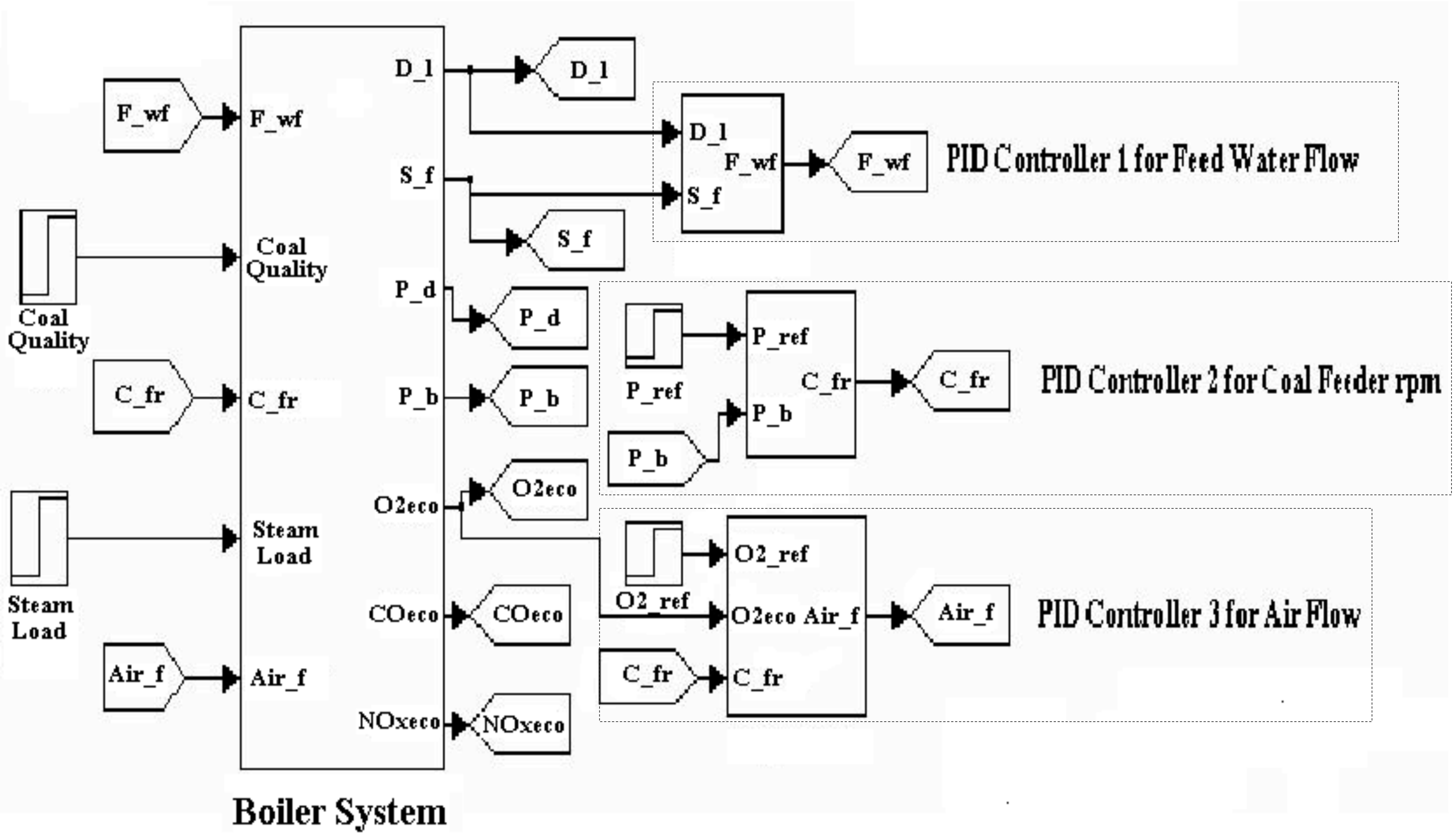
The Steam Boiler Model



PID (Proportional, Integral and Derivative) Controller
ROS - Rest of System (smart sensors, actuators, and configuration controls)

UNIVERSITY OF NEWCASTLE

The Simulink Model of Boiler System with PID Controller in MATLAB



UNIVERSITY OF NEWCASTLE

The PID Wrapper

- Monitored, and if necessary, altered all PID I/O
  - The aim - to detect and recover from errors
  - Using limited information about boiler and control system
  - Inner details of PID ignored
  - No access to full external specification of the PID
- Developed an approximate spec. for the PID
  - Based on *Acceptable Behaviour Constraints* (ABCs), of behaviour at PID interface
- Error detection
  - Systematic use of generic criteria

ABCs & Error Recovery

UNIVERSITY OF NEWCASTLE

- Inputs to PID
  - Missing, invalid, unacceptable, marginal or suspect values from the sensors or configuration variables
- Outputs from PID
  - Missing, invalid or unacceptable values intended for the actuators
- (Forward!) Error Recovery
  - Priority given to detected errors which concerned either the steam pressure or the quantity of water in the boiler
- The implemented strategy aimed at realism, but:
  - would need rigorous analysis for an actual boiler plant

UNIVERSITY OF NEWCASTLE

Wrapper-initiated Error Recovery

- Depending on circumstances the wrapper:
  - Shuts down the boiler to a safe state by sending appropriate commands to the actuators
  - Resets the PID or ROS or both to clear a supposed transient problem
  - Alerts operators by ringing alarm
  - Notes problem but takes no any action unless the problem appears to persist

UNIVERSITY OF NEWCASTLE

Some types of Error Cues, with Examples

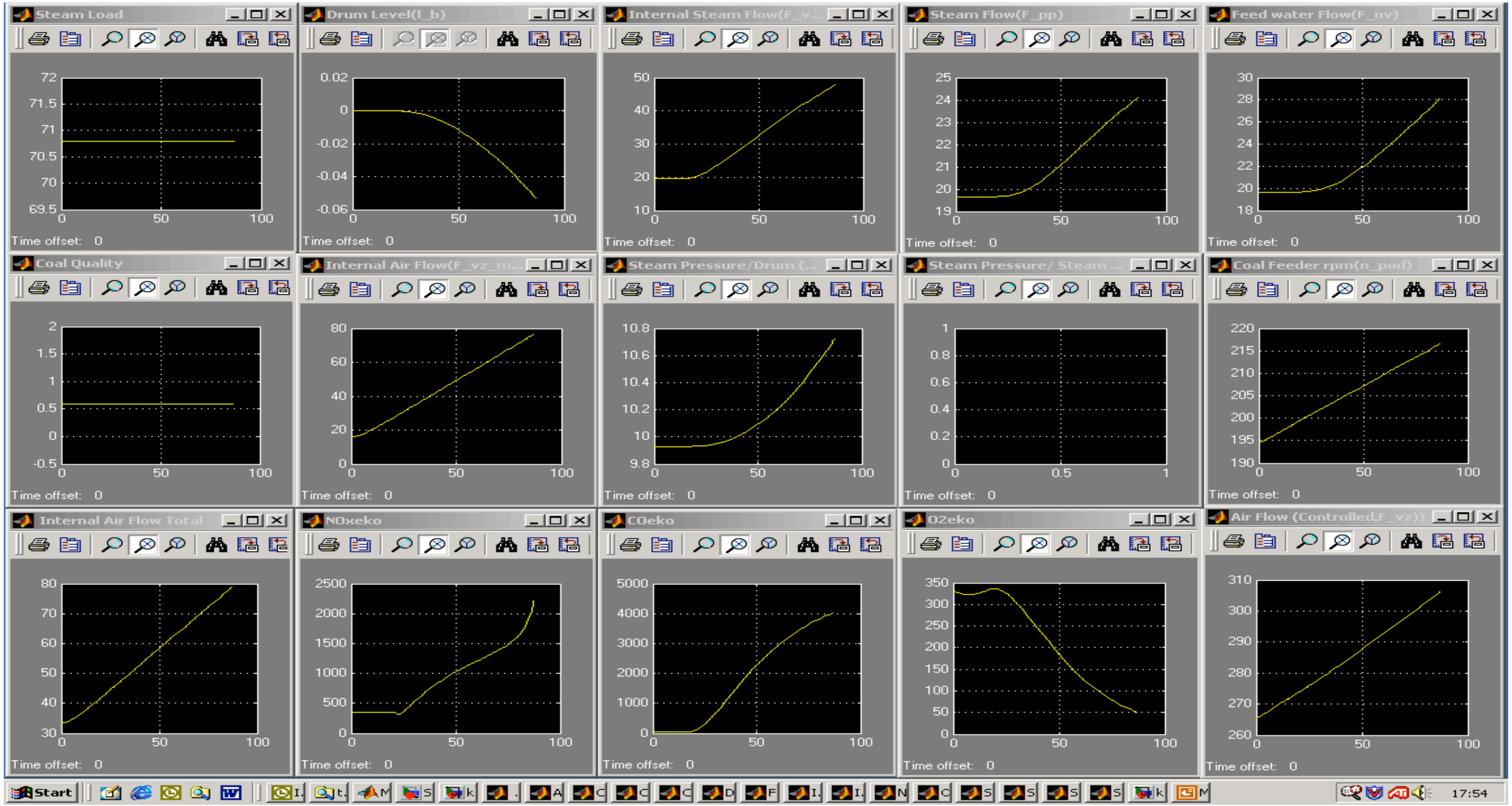
1. *Illegal output from ROS (according to ROS own specification)*: syntax errors in messages exchanged over the IEEE 488 bus.
2. *Output from ROS is detectably erroneous:* ROS sampling rate suddenly exceeds the specified rate.
3. *Output from ROS is illegal with respect to the system designer's specification of system operation:* PID inputs are outside the envelope of values anticipated by the system designer
4. *Illegal Input to the PID (according to PID's specification)*: syntax errors in messages exchanged over the IEEE 488 bus
5. *Input to the PID is illegal with respect to the PID's specification: S*et point values are mis-configured and violating the PID controller's specification.
6. *Input to the PID which is not fully trusted:* the measured derivative of PID inputs is beyond a certain value which is the maximum value the item has been tested for or lower than the normal performing value.

etc., etc.

UNIVERSITY OF NEWCASTLE

Example of Results

1. Illegal output from ROS (according to ROS own specification)



The process dies, with important variables continuing to increase (or decrease) dangerously.

UNIVERSITY OF NEWCASTLE

Initial Results

- First phase, using a range of fault injection scenarios now completed.
  - The scenarios involve signal communication faults (bias, random noise, stuck-at previous, stuck-at random) and faults affecting the PID’s control algorithms (transient zeros, control parameter overwrites).
- Preliminary examination of the results indicated the wrapper was very effective in reducing serious failures of the boiler system.
  - See:

    Protective Wrapper Development: A Case Study, Anderson, T., Feng, M., Riddle, S., Romanovsky, A. Second Int. Conf. On COTS-Based Software Systems (2003) Ottawa.

    http://www.cs.ncl.ac.uk/research/pubs/trs/papers/781.pdf

Protective wrapping – what next?

- Extend the evaluation of the steam boiler example, and use it to explore:
  - Formal development of wrappers - based on contracts derived from ABCs
  - Timing issues - deadlines and delays
  - Scoping issues - what access might a wrapper have to other variables elsewhere in the system
  - Modelling issues - how to gain confidence in accuracy of the model
  - Safety issues - standards, hazard analysis, safety cases, etc.

Protective Wrapping in Pervasive Systems

- The future is one of huge networked computer systems
  - These are likely to become pervasive, as IT is embedded into virtually everything
  - And to be required to function continuously.
- Our experiments have concerned a typical “closed” safety-critical control system
- But future pervasive systems will need to be “open”, and involve:
  - Online composition
  - Dynamic reconfiguration, evolution and upgrading
- Fault tolerance will be increasingly vital

UNIVERSITY OF NEWCASTLE

Pervasive Systems

- Even the best current development methods are insufficient for such systems
- *Some* promising research directions:
  - Dependability-explicit system development, from first design phases through into deployment
  - Cost-effective formal methods
  - Architecture theory, enabling reasoning at this level about systems and their dependability
  - Adaptivity - dynamic system integration, adjustment and evolution

  (From our recent contribution to the UK Foresight Report on Cyber Trust and Crime Prevention - see http://www.foresight.gov.uk/)
- Protective wrapping - could play a useful role in all the above research topics