AARHUS
UNIVERSITY
DEPARTMENT OF ENGINEERING

Architecture & Design of Embedded Real-Time Systems (TI-AREM)

SW Architecture Styles and Two-Part Architecture Model

Version: 2-2-2015

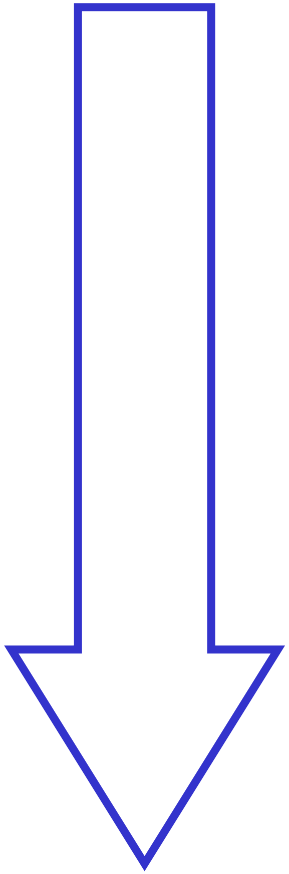
Agenda

State of the Art for SW architecture

- Architectural Styles
  - Pipes and Filters
  - Two-part architecture model

State of the Art for SW Architecture

High level



Low level

- **Architectural Styles**
  - Styles dominates a given architecture
  - Examples: Pipes and Filters, Layered Architectural Structure, Two-part architecture model
- **Architectural Patterns**
  - Address “System-wide” design problems
  - Are not dominating and are often combined with other design patterns
  - Examples: concurrency, persistency and memory p.
- **Design Patterns (GoF) – Mechanistic Design**
  - Design patterns have often a more local effect
  - Examples: Strategy pattern, State Pattern
- **Idioms**
  - Language near patterns and mechanism
  - Examples: Counted pointer for C++

Architectural Styles (Shaw&Garlan)(1)

- Five categories of ***Architectural Styles***:
  - Dataflow systems
    - Batch sequential, **Pipes and filters**
  - Call-and-return systems
    - **OO systems**, Main program and subroutine, Hierarchical layers
  - Independent components
    - Event systems, Communicating processes
  - Virtual machines
    - Interpreters, Rule-based systems
  - Data-centered systems (repositories)
    - Databases, Hypertext systems, Blackboards

Ref. [Shaw&Garlan96]

Architectural Styles (Shaw&Garlan)(2)

- Examples of ***Architectural Styles***:
  - **Pipes and filters**
  - Data abstraction and Object-Oriented organization
  - **Event-based**, implicit invocation
  - Layered systems
    - Examples:
      - OSI model
      - Business oriented systems: presentation, business logic and model layer
  - Repositories
  - Interpreters
  - **Process control**

Ref. [Shaw&Garlan96]

Buschmann’s three Pattern Categories

Architecture patterns:

- Layers
- Pipes & Filters
- Blackboard
- Broker
- Model-View-Controller
- Presentation-Abstraction-Control (PAC)
- Microkernel
- Reflection

Shaw& Garlan Styles

Design Patterns:

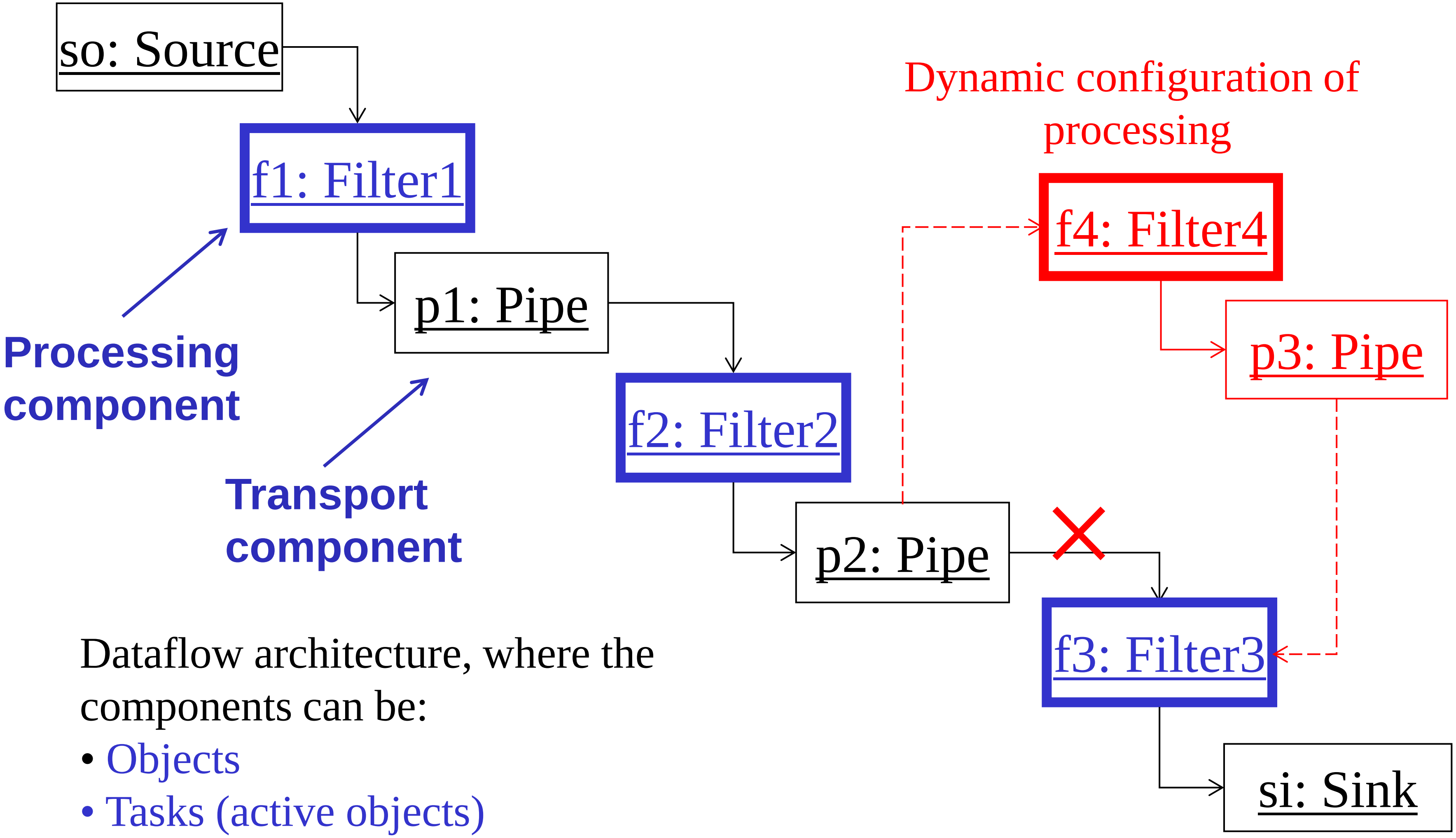
- Observer (GoF)
- Publisher-subscriber
- Strategy (GoF)
- Composite (GoF)
- Abstract Factory (GoF)
- Bridge (GoF)
- Proxy (GoF)
- Command Processor
- View Handler
- Master-slave

Idioms:

- Singleton (GoF)
- Factory Method (GoF)
- Counted pointer, Handle-Body
- Envelope-Letter

Ref. [Bushmann 96 – POSA1]

Pipes & Filters Architecture Style



Processing component

Transport component

Dynamic configuration of processing

Dataflow architecture, where the components can be:

- Objects
- Tasks (active objects)
- Processors (HW)

Pipes and Filters Structure (CRC)

CRC= Class, Responsibility and Collaboration

Class **Filter**	***Collaborators***
Responsibility Get input Perform function Set output	Pipe

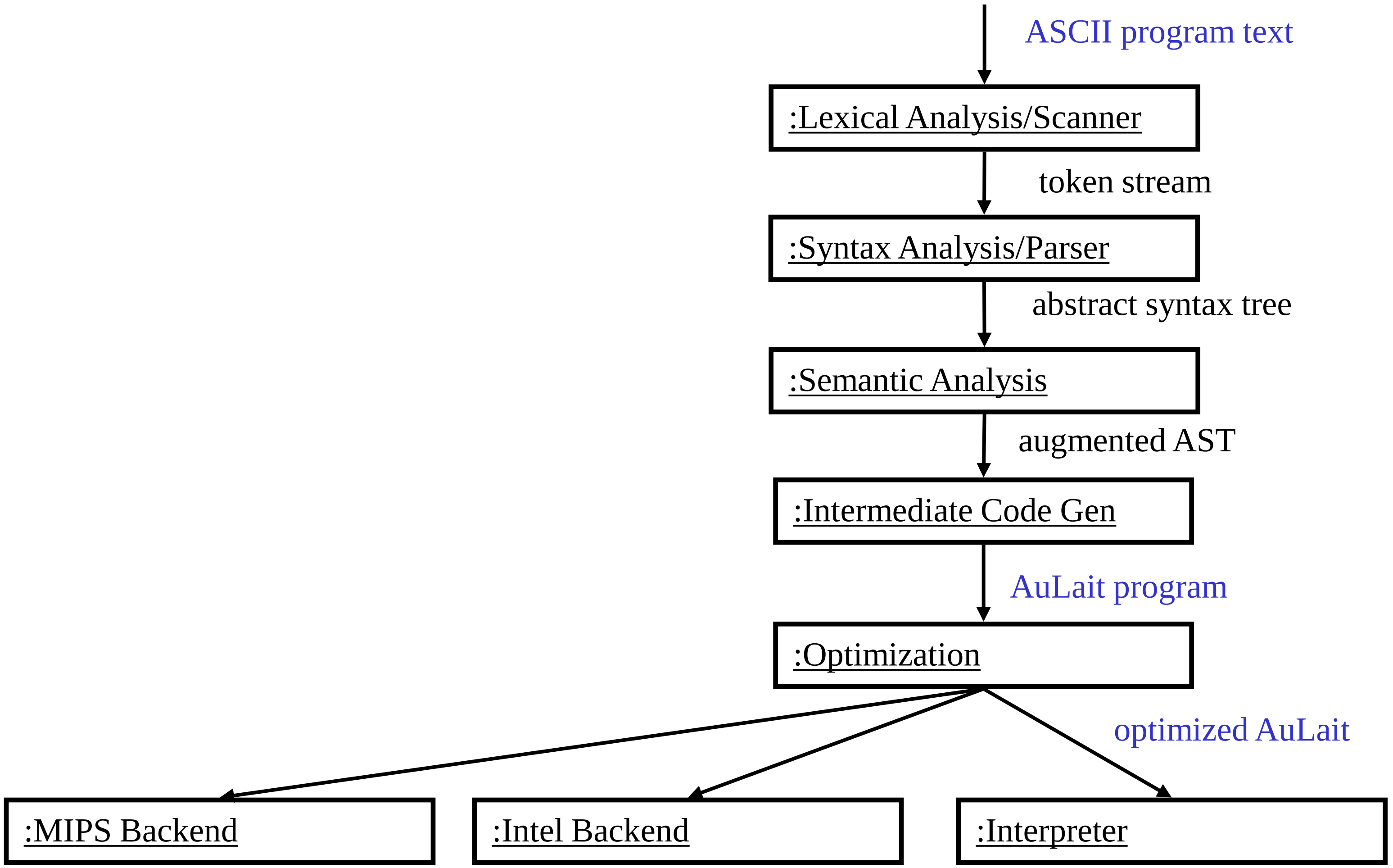
Class **Pipe**	***Collaborators***
Responsibility Transfer data Buffer data and synchronize filters	Data Source Data Sink Filter

Class **Data Source**	***Collaborators***
Responsibility Deliver input to processing pipeline	Pipe

Class **Data Sink**	***Collaborators***
Responsibility Consumes output	Pipe

Ref. [Bushmann 96 – POSA1]

Pipes and Filters: Mocha Compiler Example



ASCII program text

:Lexical Analysis/Scanner

token stream

:Syntax Analysis/Parser

abstract syntax tree

:Semantic Analysis

augmented AST

:Intermediate Code Gen

AuLait program

:Optimization

optimized AuLait

:MIPS Backend

:Intel Backend

:Interpreter

Ref. [Bushmann 96 – POSA1]

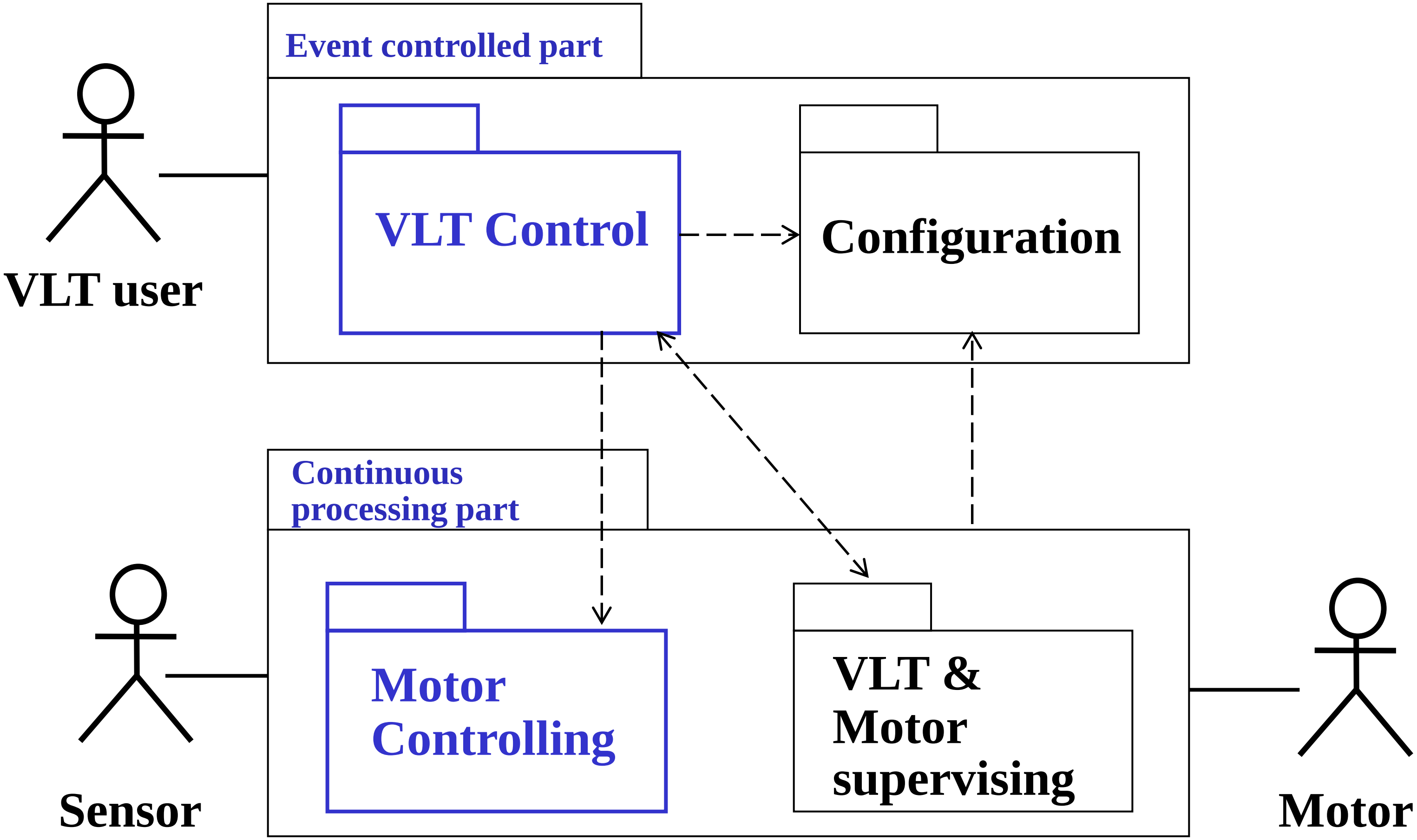
Two-Part Architecture Model Style

- Combines the following two ***Shaw&Garlan Architectural Styles***:
  - Event-based
  - Process Control (plus *pipes & filters* internal)
- This combination can be used for many embedded systems
- Examples:
    - control and regulation e.g. a frequency converter for controlling a motor
    - measuring instruments e.g. an oscilloscope or a noise-meter
    - audio/video: e.g. a CD player

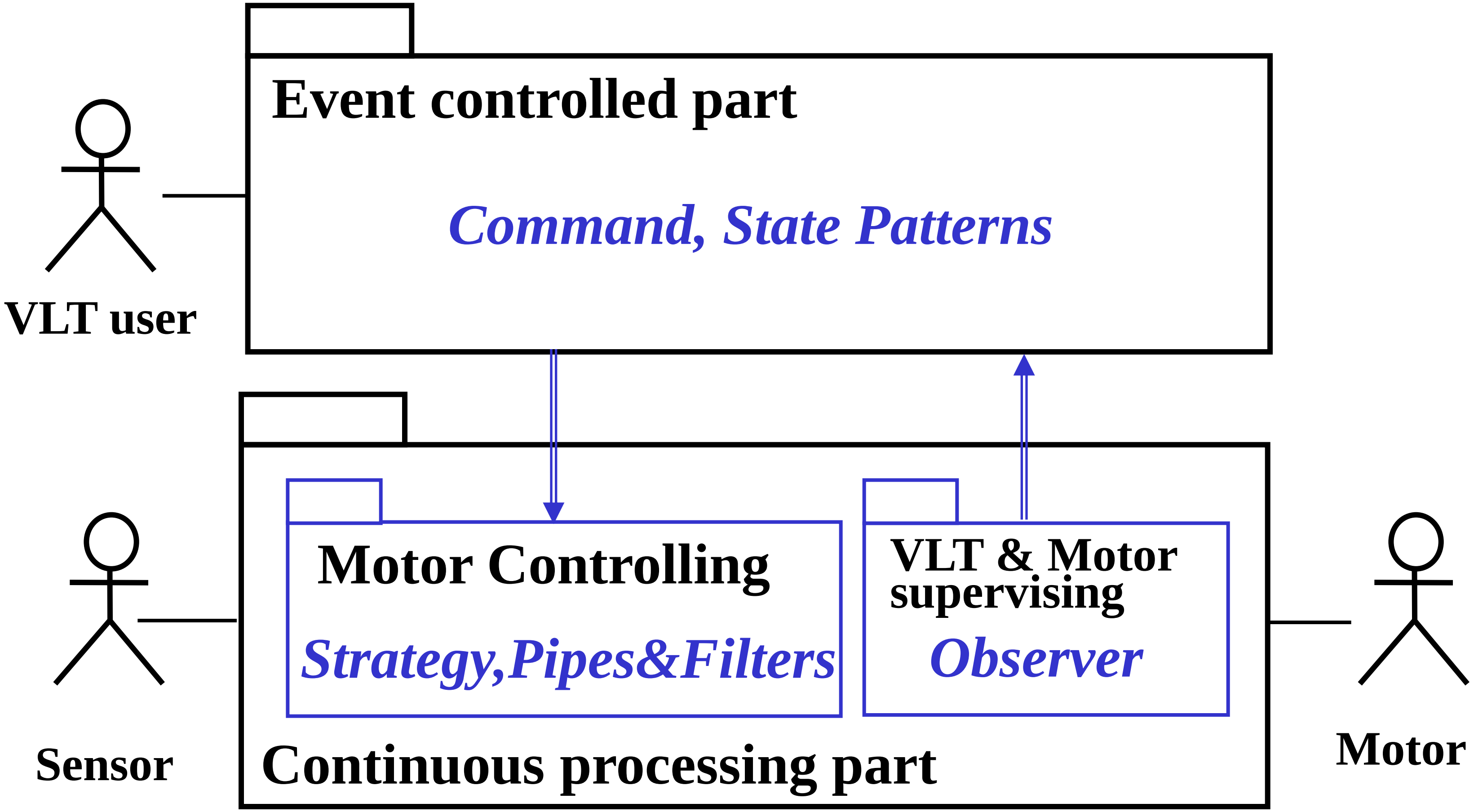
Ref: [COT] Center for Object Technology project (Case 2)

Two Part Architectural Model Example

Danfoss frequency converter (called a VLT)

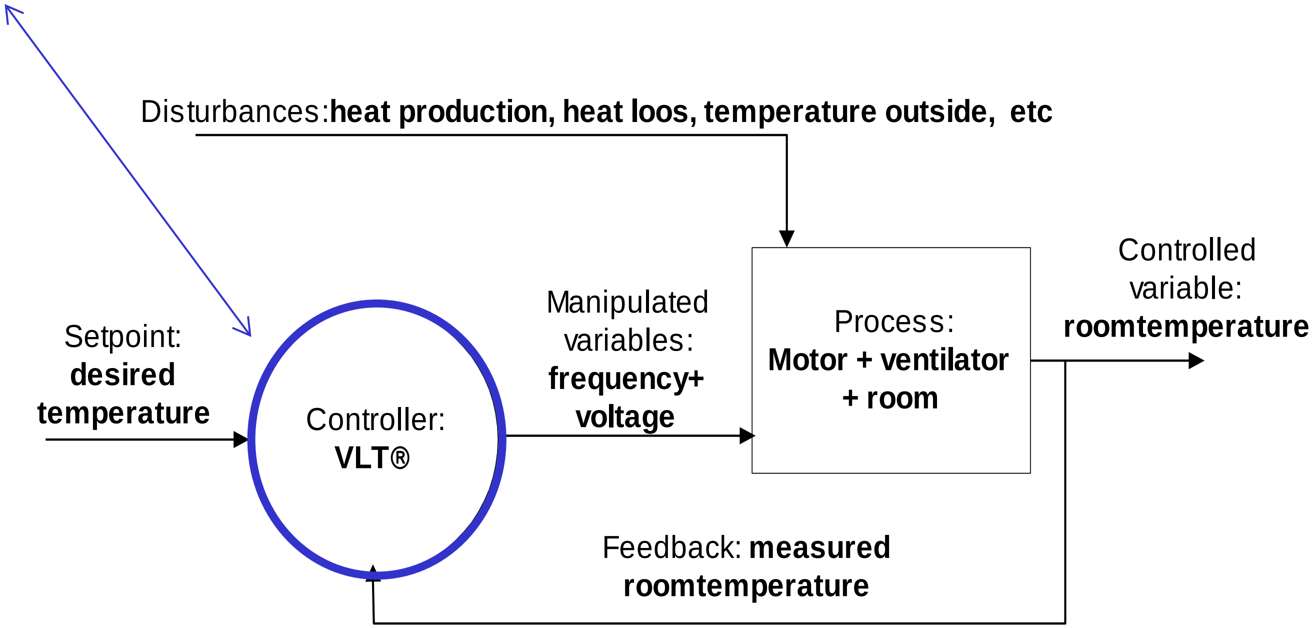


Used Design Patterns in the OO-VLT



Ventilator Control as Process Control Style

Danfoss frequency converter (VLT)



Disturbances:**heat production, heat loos, temperature outside, etc**

Setpoint: **desired temperature**

Controller: **VLT®**

Manipulated variables: **frequency+ voltage**

Process: **Motor + ventilator + room**

Controlled variable: **roomtemperature**

Feedback: **measured roomtemperature**

Example of “*closed-loop feedback*” control

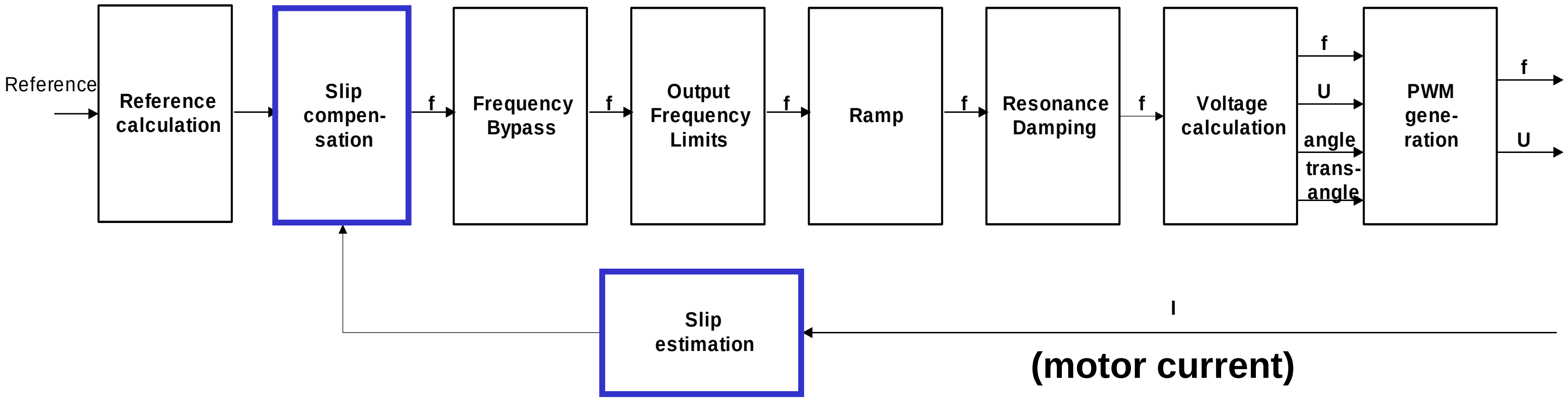
Shaw & Garlan: Process Control Style

Ref. [Shaw&Garlan96]

Frequency Converter Example

Block Diagrams with two different operating modes

Speed Open loop controller:



Reference

Reference calculation

Slip compen-sation

f

Frequency Bypass

f

Output Frequency Limits

f

Ramp

f

Resonance Damping

f

Voltage calculation

f

U

angle

trans-angle

PWM gene-ration

f

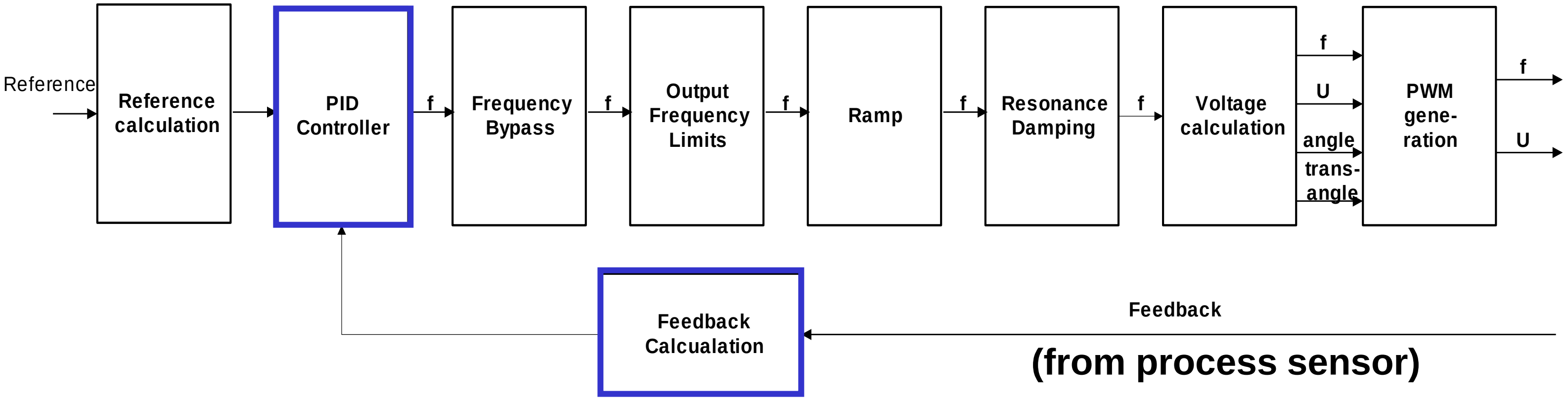
U

Slip estimation

I

(motor current)

Process Closed Loop controller:



Reference

Reference calculation

PID Controller

f

Frequency Bypass

f

Output Frequency Limits

f

Ramp

f

Resonance Damping

f

Voltage calculation

f

U

angle

trans-angle

PWM gene-ration

f

U

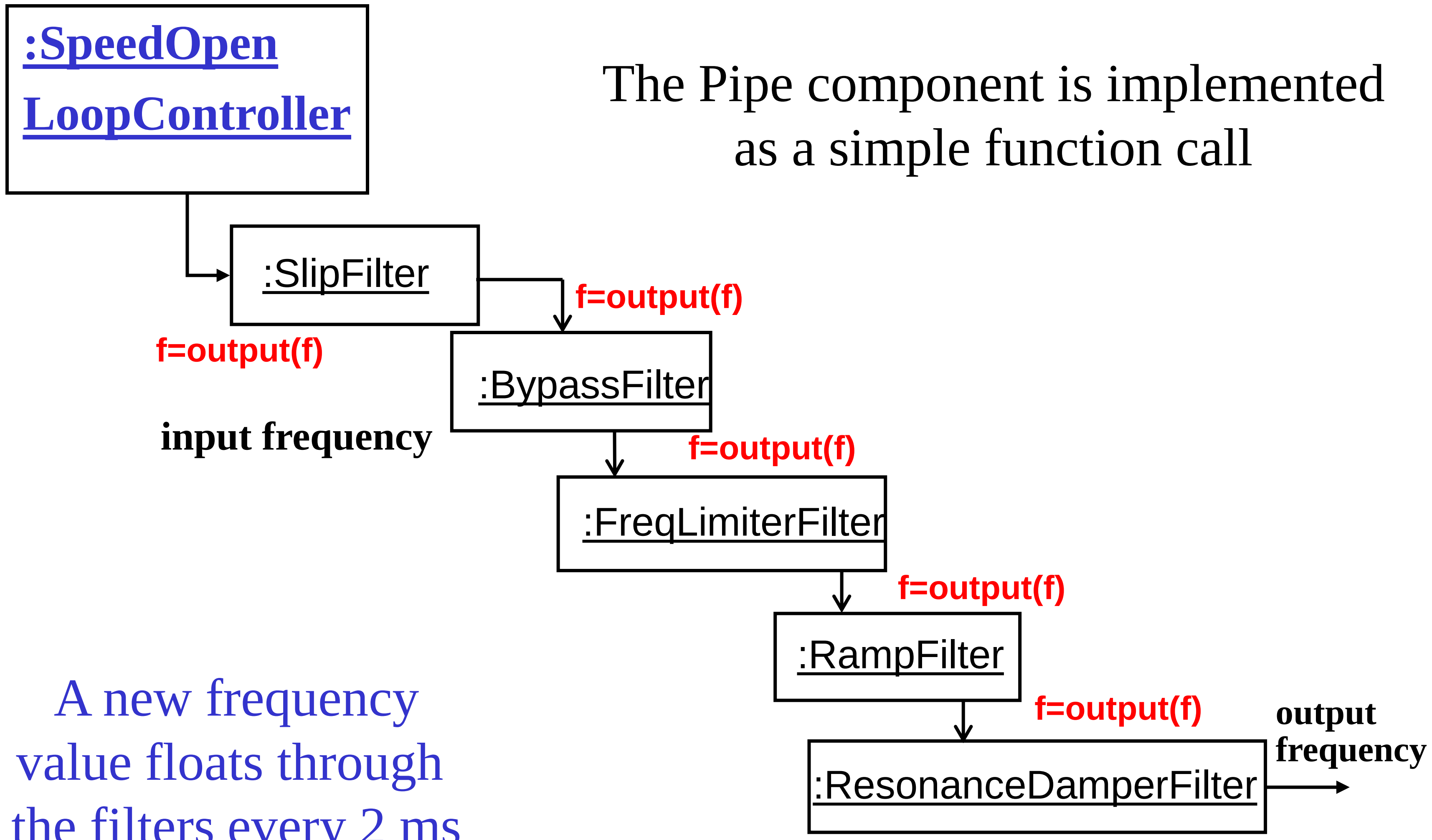
Feedback Calcualation

Feedback

(from process sensor)

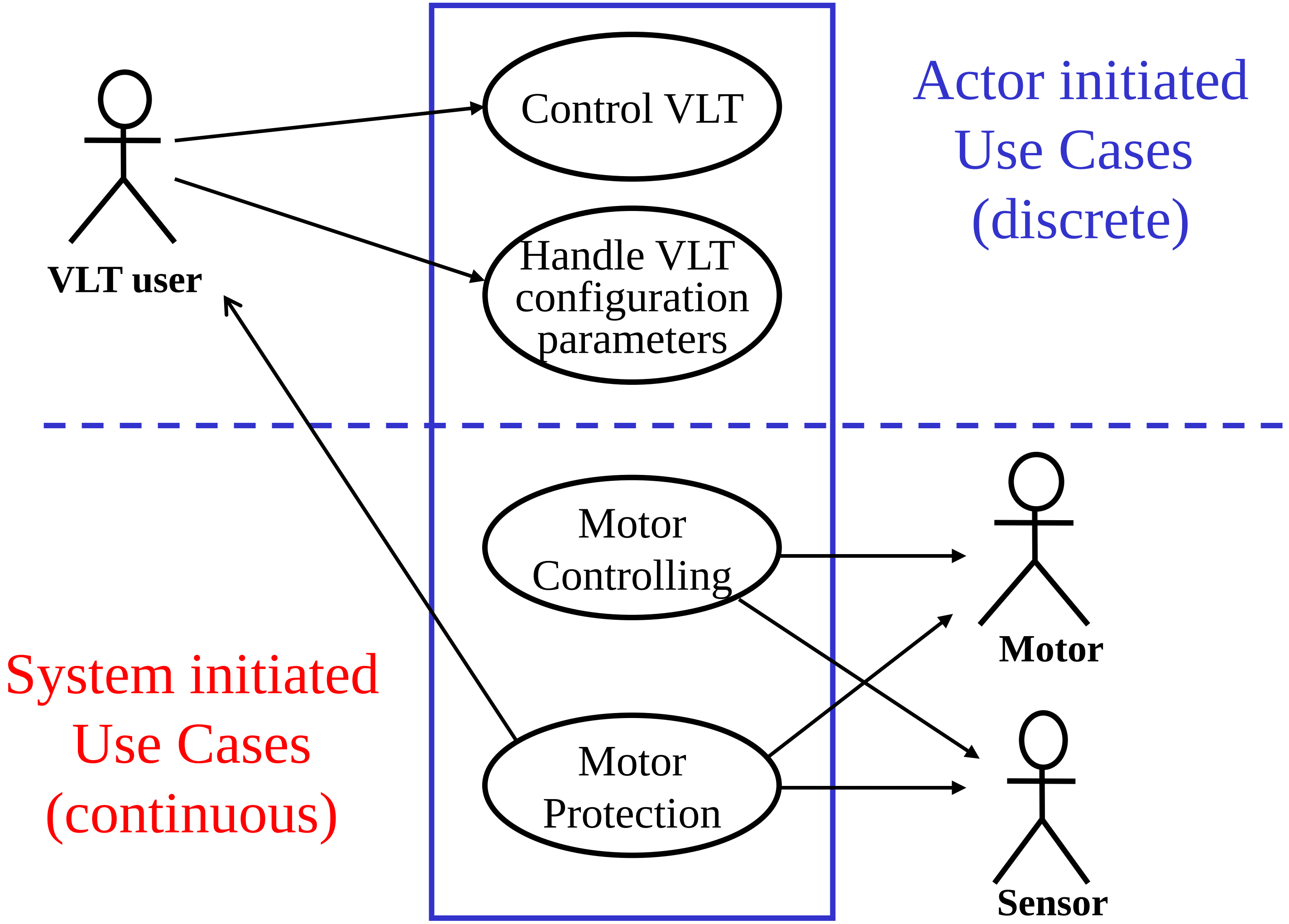
Object Diagram for SpeedOpenLoopController

The Pipe component is implemented as a simple function call

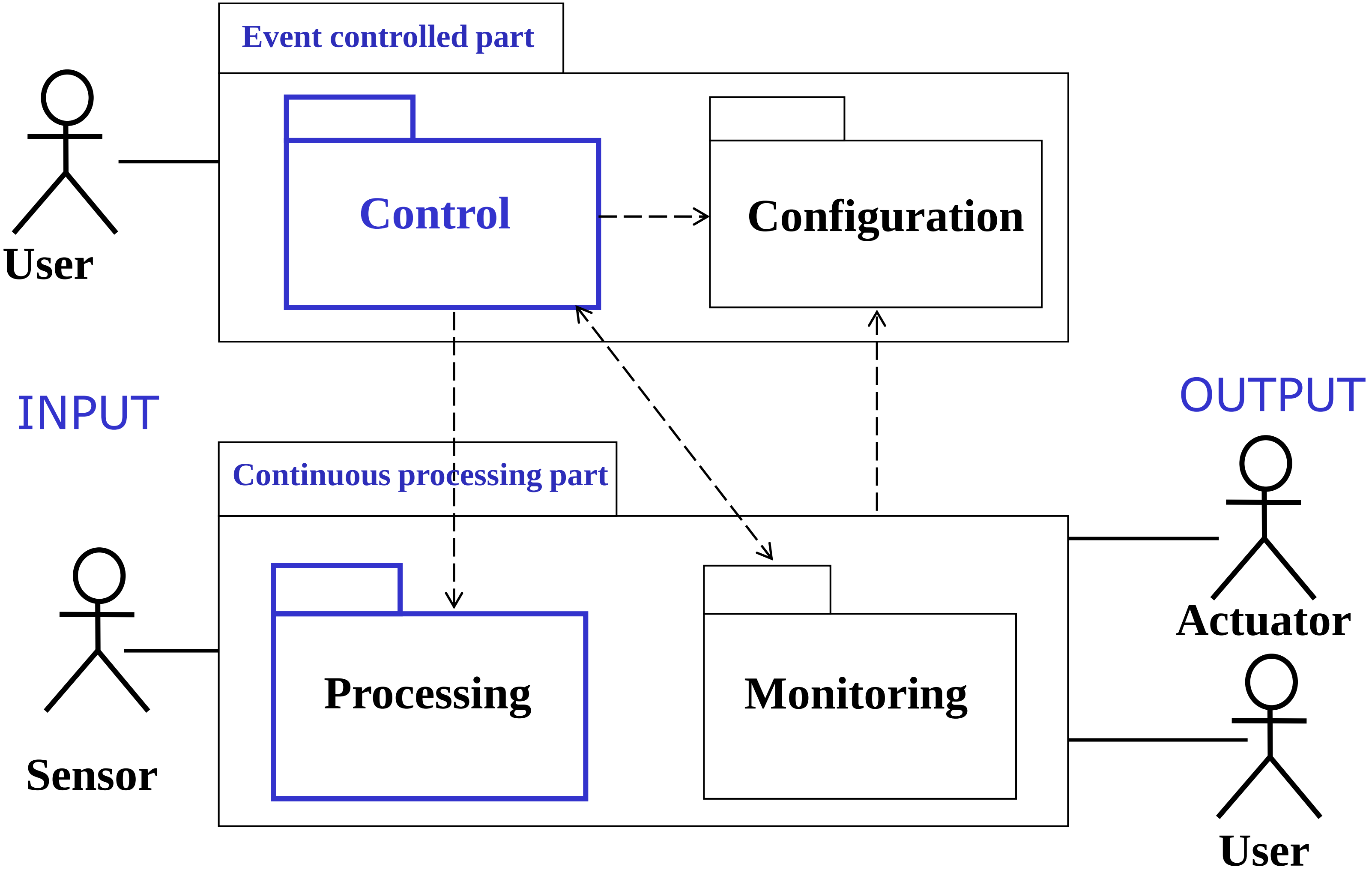


A new frequency value floats through the filters every 2 ms

Two types of VLT Use Cases


VLT user
Control VLT
Handle VLT configuration parameters
Actor initiated Use Cases (discrete)
Motor Controlling
Motor Protection
System initiated Use Cases (continuous)
Motor
Sensor

Architectural Style: Two-Part Architecture Model



Development towards a Framework

- The **first version** of the frequency converter software was developed for one specific type of frequency converters, where motor signals where generated by an ASIC (Application Specific Integrated Circuit)
- The SW architecture was defined by an OO model and implemented in C++
- A lot of existing C-code (i.e. ~80%) was wrapped in OO Wrapper classes
- The **second version** was modified for another frequency converter type, where the motor signals where generated by a DSP (Digital Signal Processor)
- Part of the architecture was modified, but the main architecture ideas, based on the two-part architecture model, was kept as the basic architecture
- The software has been developed towards a framework for motor controlling software, to be used for a hole family of frequency converters

Summary

- Architecture Styles and examples
  - Pipes and Filtes
  - Two-part architecture model for embedded systems

References

- [Shaw&Garlan96]
Software Architecture, Perspectives on an Emerging Discipline
by Mary Shaw, David Garlan, Prentice-Hall, 1996
- [COT1997-2000]
Center for Object Technology (COT),
  - a Danish research project financed by Center for IT Research. The COT project has a number of reports on SW architecture,
  - http://www.cit.dk/COT/ - see under Report Series
- [Jeppesen&Hansen2001]
Designing Event-Controlled Continuous Processing Systems
by Hans Peter Jepsen and Finn Overgaard Hansen, Proceedings Embedded Systems Conference, San Francisco April 2001.