Year 2 Review
Paris, November 8th and 9th, 2006

Achievements and Perspectives:

Adaptive Real-Time

Cluster leader: Giorgio Buttazzo
Scuola Superiore Sant'Anna
Pisa, Italy

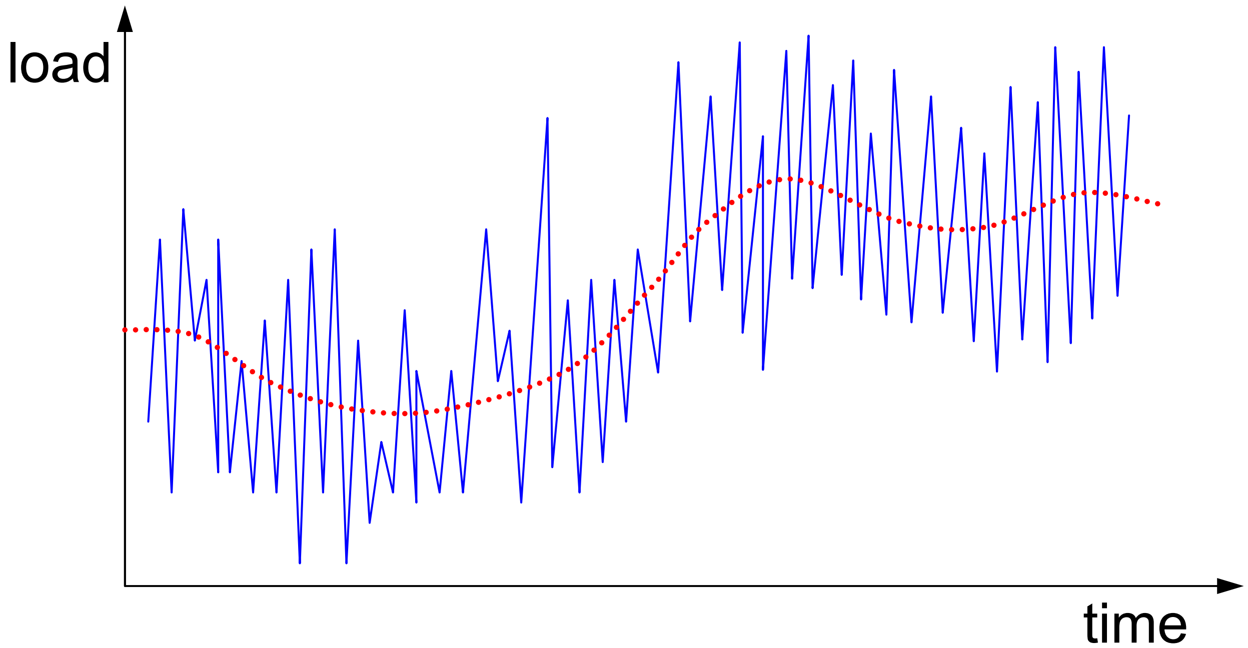
Outline of the Presentation

- **The ART cluster**
  - Goal
  - Research areas
  - Partners
- **Activities**
  - Aims
  - Achievements
  - Future evolution
- **Scientific Highlights**
  - Flexible scheduling
  - Real-time networks

The ART Cluster

Goal

Investigate novel methodologies to provide predictability and flexibility for systems where resources requirements are inherently unstable and difficult to predict in advance.



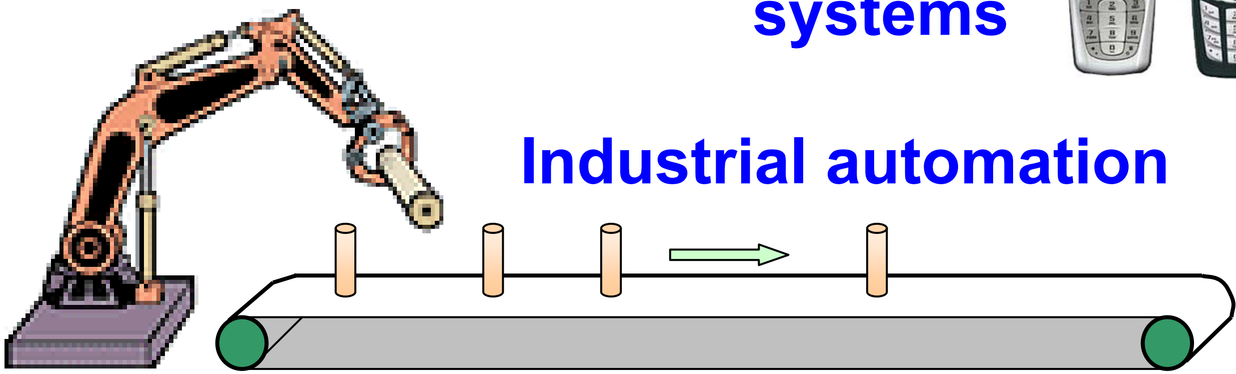
Application domains

Consumer Electronics

Multimedia systems

Telecommunication systems

Industrial automation



Research areas

- **Operating Systems**
  - Predictability, Portability, Standards, ...
- **Resource Management Algorithms**
  - Scheduling, Mutual Exclusion, Power-aware algorithms, ...
- **Networks**
  - Protocols, Synchronization, Co-ordination, ...
- **Middleware**
  - Adaptativity, QoS Mangement, ...
- **Languages**
  - Expressiveness, Reliability, ...

ART partners	Role
Cluster Leader	
SSSA, Pisa:	RT scheduling and RT kernels
Core Partners	
Univ. of Aveiro:	networking, distributed applications
TU Kaiserslautern:	video streaming, scheduling
Univ. of Cantabria:	schedulability analysis and OS standards
Univ. of York:	real-time languages
UP Madrid:	QoS and resource management
PI Porto:	heterogeneous networks
Affiliated Partners	
Univ. of Pavia:	RT scheduling and resource management
Evidence:	kernels and tools for RT systems
UP Catalonia:	control methodologies for RT systems
Univ. of Catania:	distributed systems
UC3 Madrid:	QoS and resource management

ART Cluster Activities

JPIA Platforms

- A common infrastructure for adaptive RT systems

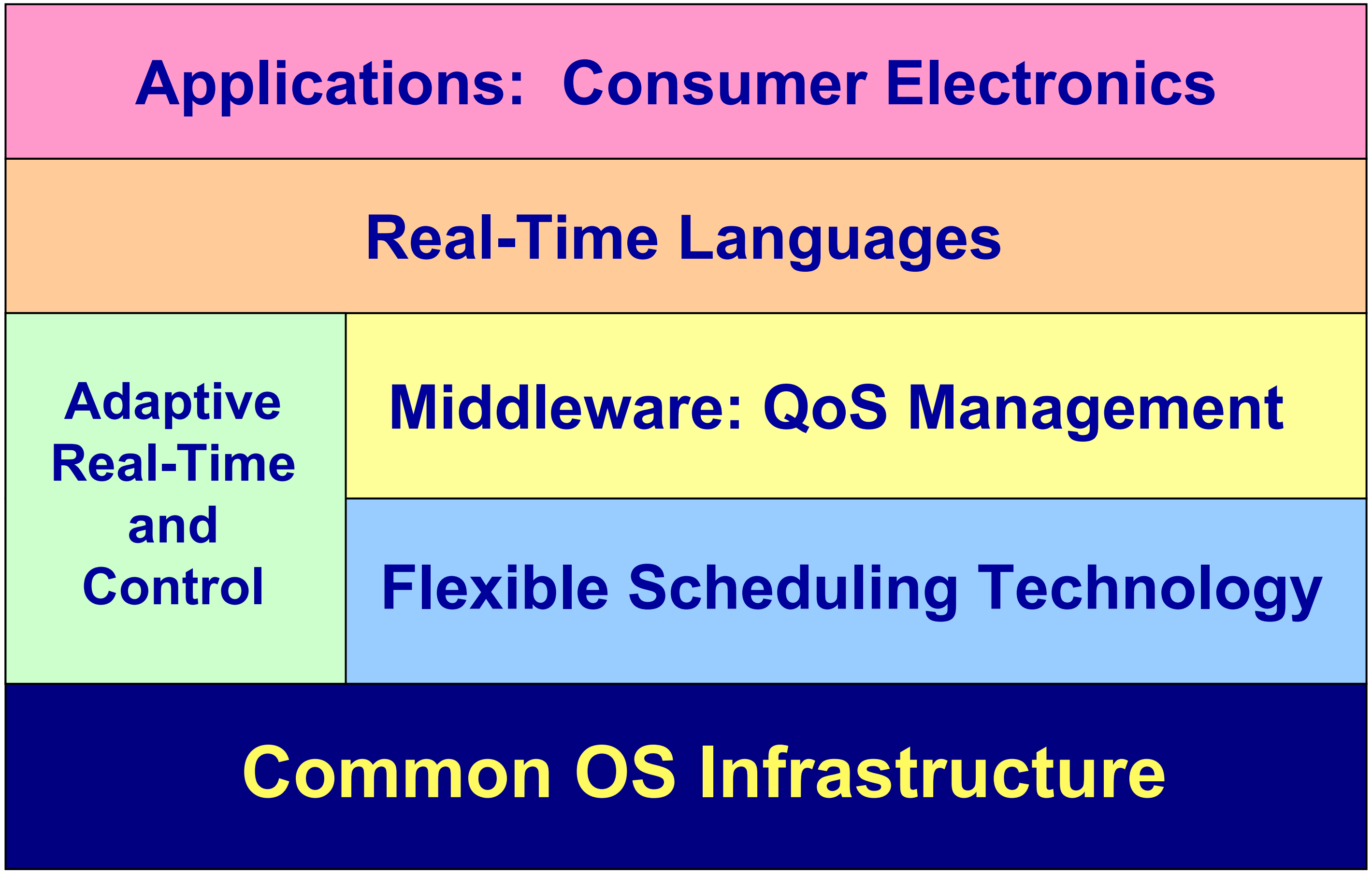
JPRA NoE Integration

- Adaptive real-time, HRT and Control
- QoS Aware Components

JPRA Cluster Integration

- Flexible Scheduling Technologies
- Adaptive Resource Management for Consumer Electronics
- Real-Time Languages ⟵ **New** (started on March 2006)

Overview of the activities of the ART cluster



Overview of the activities of the ART cluster

Applications: Consumer Electronics

Real-Time Languages

Adaptive Real-Time and Control

Middleware: QoS Management

Flexible Scheduling Technology

Common OS Infrastructure

Activity 1: JPIA Platforms

A Common Infrastructure for Adaptive RT Systems

Objectives

- Provide a shared OS platform for experimenting new real-time software technologies on real applications:
  - novel scheduling algorithms
  - resource management techniques
  - energy-aware policies
  - overload handling techniques for robustness and predictability
- Show how to extend current OSs and nets to support RT appl^{s} with highly dynamic behavior. Impact on standards (POSIX, OSEK, …)
- Speed up the process of transferring research results to industry

Activity 1: JPIA Platforms

A Common Infrastructure for Adaptive RT Systems

Approach

– Select a flexible, open-source RT kernel ⇒ **Shark**

– Kernel installation on partner sites and personnel training

– Support partners in developing RT applications and new kernel mechanisms

Activity 1: JPIA Platforms

A Common Infrastructure for Adaptive RT Systems

Achievements in Year 2

1. **New release: SHARK 1.5.1** (July 25, 2006)
   - bugs fixed
   - event tracer with log file
   - support for USB devices
   - support for Dynamic Voltage Scaling
   - new documentation + Quick Guide

2. **Repository for RT software**
   - Kernel modules (scheduling, resource protocols)
   - RT Applications (supported and unsupported)

Activity 1: JPIA Platforms

A Common Infrastructure for Adaptive RT Systems

Achievements in Year 2

3. SHARK for education

- Used in many university courses for teaching RTOS
- simulated control systems
- simple robotics applications
- verify theory in practice

First European Laboratory on Real-Time and Control for Embedded Systems

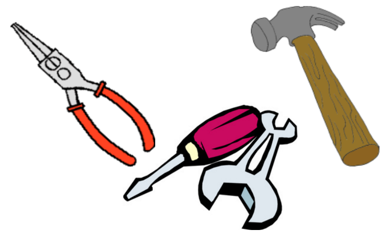
Scuola Superiore Sant'Anna, Pisa, Italy

July 10-14, 2006

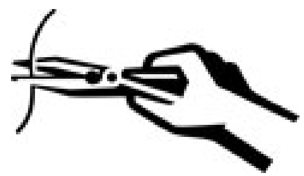
Organized by: ART & Control Clusters

Aims

1. Provide concepts, methodologies and tools used for developing RT embedded systems (scheduling, RTOS, networks, and control)



2. Show how to apply theory into practice, teaching students how to develop simple control applications using Shark.



First European Laboratory on Real-Time and Control for Embedded Systems

30 students from 10 different countries

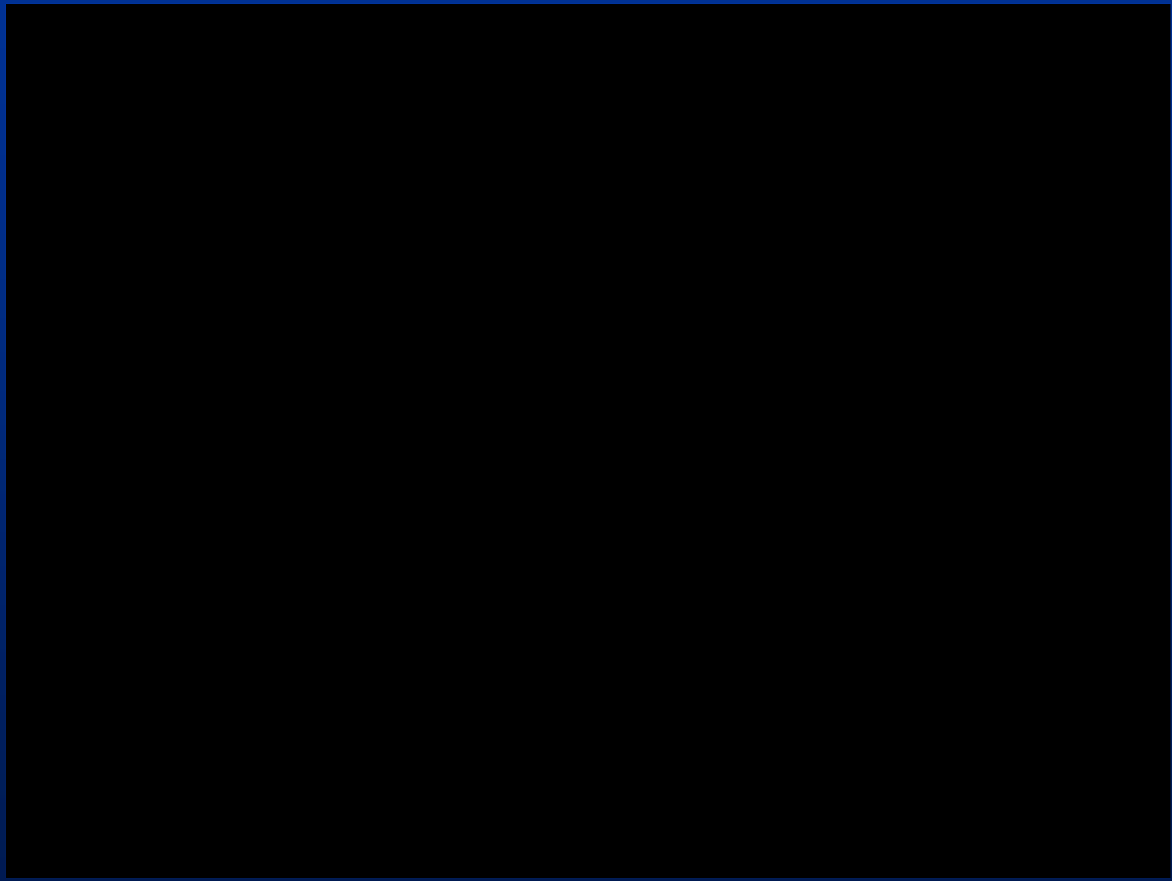
Morning	Afternoon
1. Scheduling	Getting familiar with SHARK
2. Resource management	Writing a simple RT demo
3. Control	Implementation on Shark
4. Networks	A distributed control application
5. Working hard on applications	Project presentation

Morning lectures

Afternoon lab

First European Laboratory on Real-Time and Control for Embedded Systems

Sample projects



1. 2 d.o.f. Laser pointer
2. Grasping by vision
3. Robot Goalkeeper
4. Inverted pendulum
5. Light gun
6. Data glove acquisition
7. Distributed ping pong
8. Distributed car race
9. Ball and plate balancing by vision
10. Vehicle following a target by vision

Activity 1: JPIA Platforms

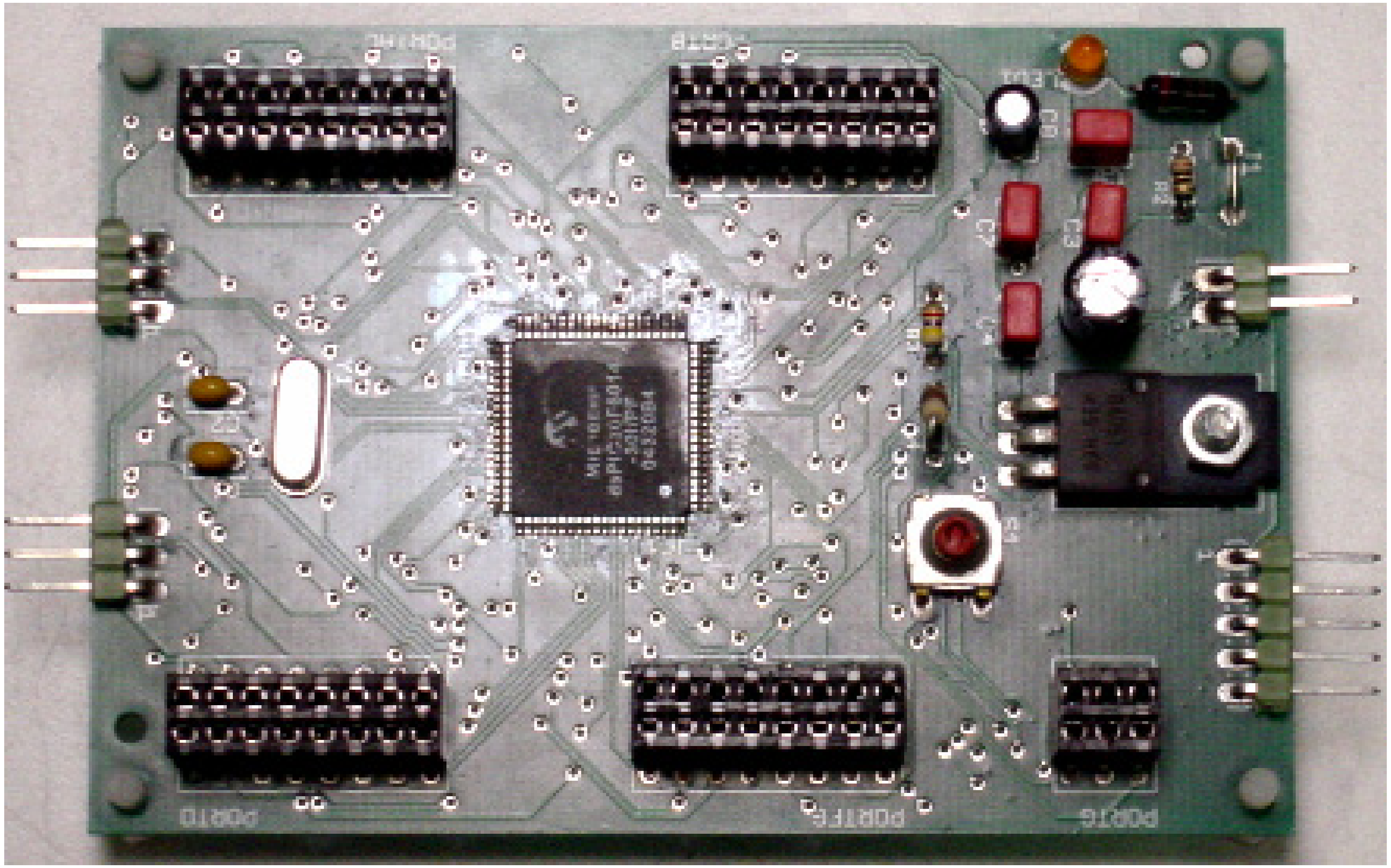
A Common Infrastructure for Adaptive RT Systems

Achievements in Year 2

4. Agreement with Microchip Technology

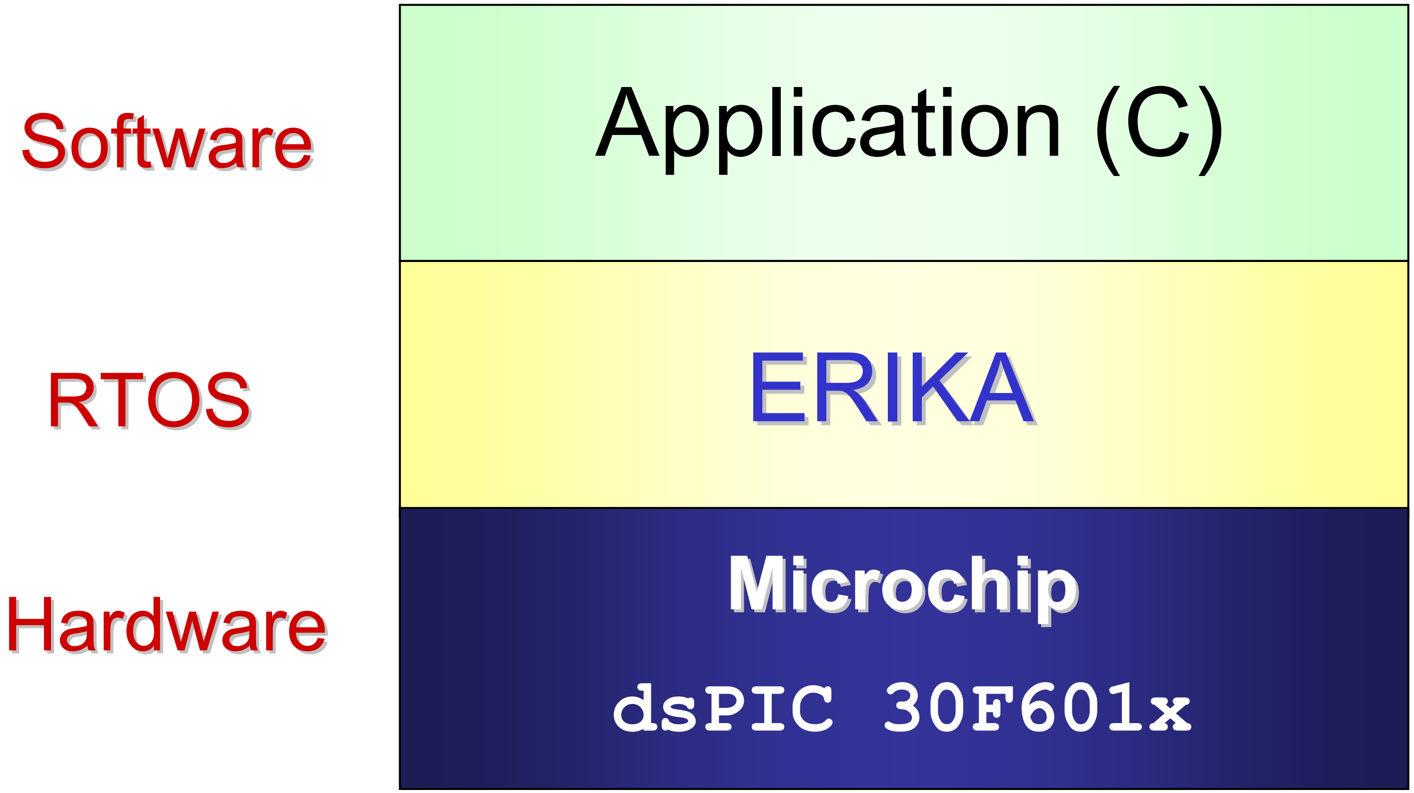
- Develop a general purpose board for embedded systems
- Provide RTOS technology with OSEK compatibility
- Develop a number of special purpose boards with related applications

Embedded Control Board



Dimensions: ***92 x 62 mm*** Weight: ***25 g***

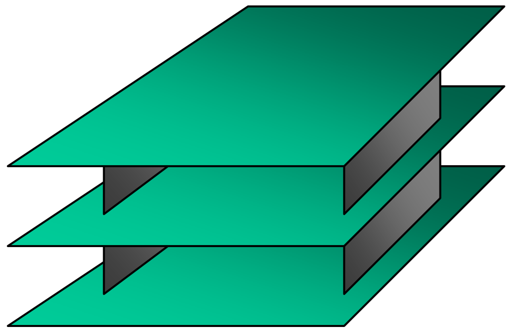
Architecture



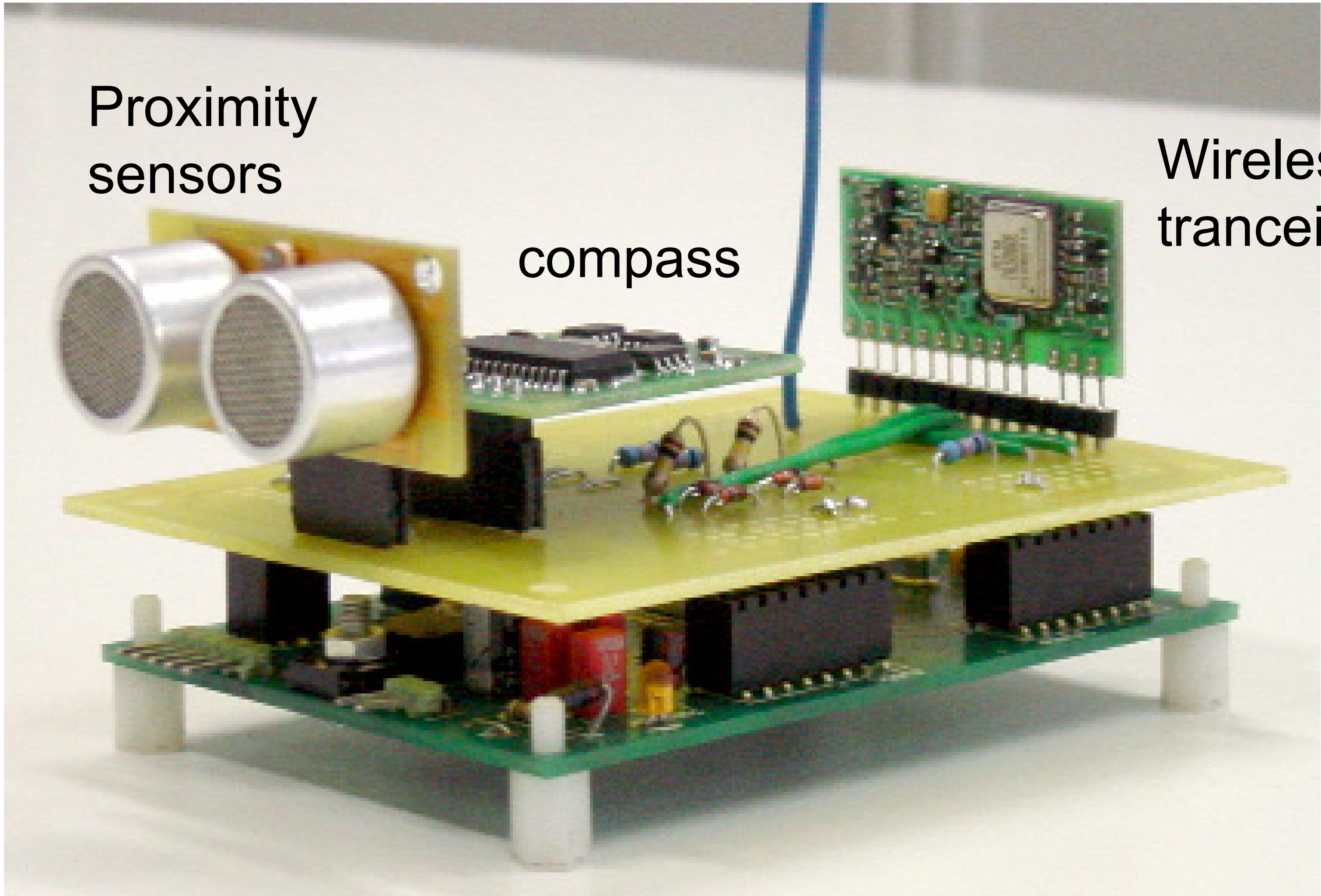
Embedded Control Board

Main Features:

- On-board power regulation
- On the fly and remote programming
- Expandible with piggy-back connections
  - Servomotor controller
  - Inertial platform
  - Connection bridge
  - Sensor network node



Example



Activity 1: JPIA Platforms

A Common Infrastructure for Adaptive RT Systems

Future Work and evolution

Y3	– Extensive testing to identify algorithms and tools to support adaptive RT systems – Provide advanced features in OSEK compliant kernels

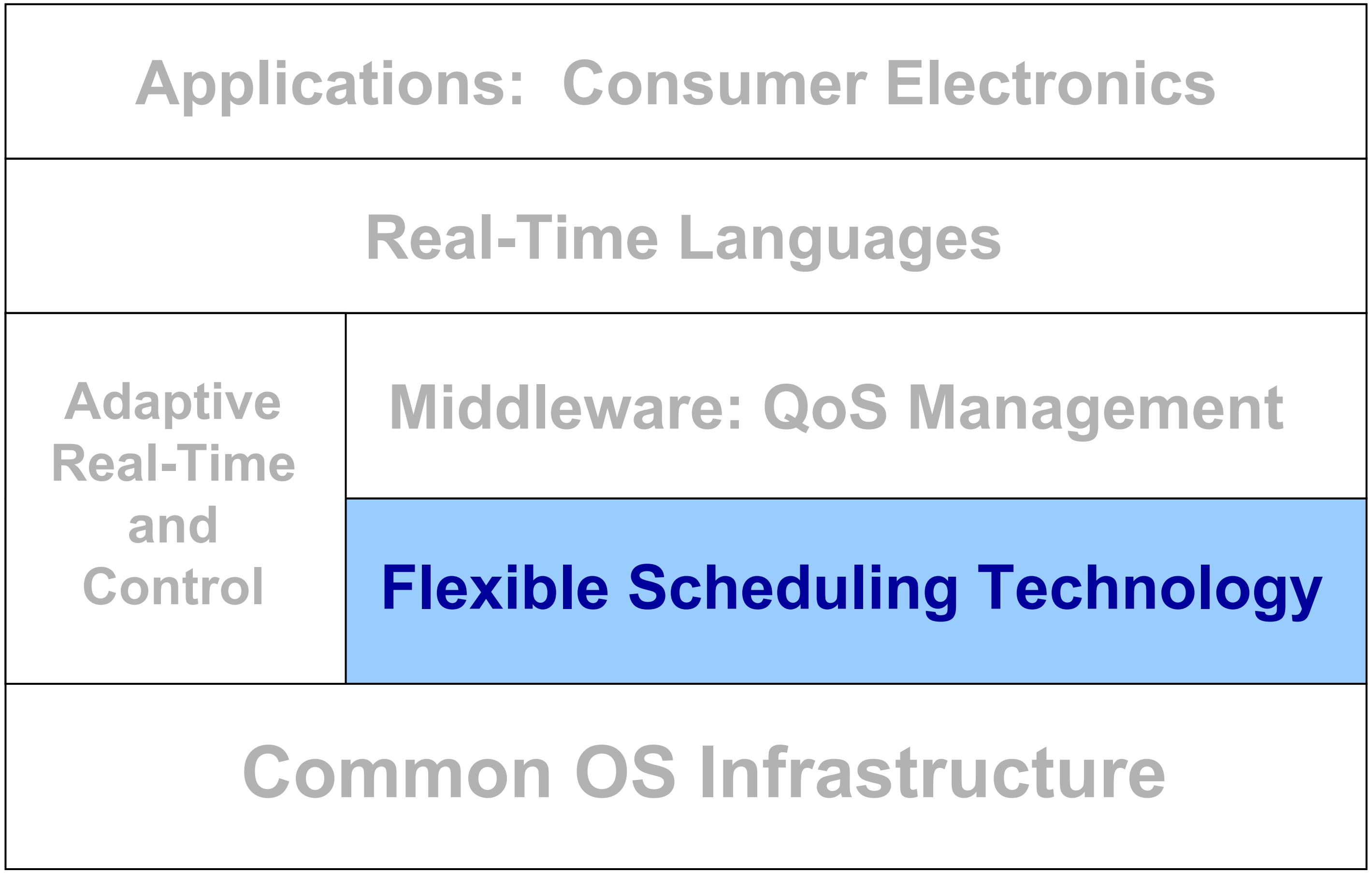
Y4	– Towards component-based RT operating systems – Identify problems and propose solutions

Note on Shark

Virtualization

- We started exploring the possibility of making Shark to coexist with Linux, porting it over the L4 microkernel.
- The port is working, but still lacks of a complete set of device drivers.
- Other approaches to virtualization should be considered, at least trying to use Xen as a hypervisor, but we had no direct experience with them.
- We still have to try the interaction between a running Shark over L4 and a running Linux over L4.

Overview of the activities of the ART cluster



Flexible Scheduling Technologies

Activity Leader: Michael Gonzalez Harbour (Univ. of Cantabria)

Objectives

- Provide predictability and adaptivity to systems where resource requirements are inherently unstable and difficult to predict.
- Make OS and networks able to support
  - resource reservation
  - different scheduling paradigms
  - energy-aware policies
  - overload handling techniques for graceful degradation
- Develop a real-time scheduling framework to perform adaptive QoS control of applications demanding various types of timing constraints with dynamic characteristics.

Flexible Scheduling Technologies

Achievements in Year 2

Requirements for integrated resource scheduling framework

- Workshop in Massy (Paris), June 2006 (20 participants, 14 institutions)
- Produced a collection of requirements for flexible scheduling framework
- Input will be used to design a framework for integrated resources

Baseline for integrated resource scheduling framework

- FIRST scheduling framework (IST project finished in 2005)
- Multiple network protocols, RT-EP, FTT ethernet, FTT CAN, wireless,…

New theoretical developments:

- Server-based scheduling, network scheduling, energy aware scheduling

Flexible networked architectures and communication protocols

- Field buses, wireless, general-purpose networks

Flexible Scheduling Technologies

Plan for the next 18 months

- Requirements analysis for flexible scheduling framework
- Architectural model of the framework, based on contract-based scheduling, integrating multiple resources: processors, networks, memory, energy, dynamically reconfigurable modules, shared resources, interrupts
- Design of the QoS resource manager performing adaptive management of the contract-based framework
- Start implementation and evaluation work
- Continue theoretical developments

Overview of the activities of the ART cluster

Applications: Consumer Electronics

Real-Time Languages

Adaptive Real-Time and Control

Middleware: QoS Management

Flexible Scheduling Technology

Common OS Infrastructure

QoS Aware Components

Activity Leader: Alejandro Alonso (UP Madrid)

Objective: Improve support for development and execution of QoS-aware components

Expected results:

- Techniques for specification of QoS character.
- Include QoS characteristics in interfaces and infrastructures
- Automatic generation of analyzable models
- Run-time support for QoS-aware components

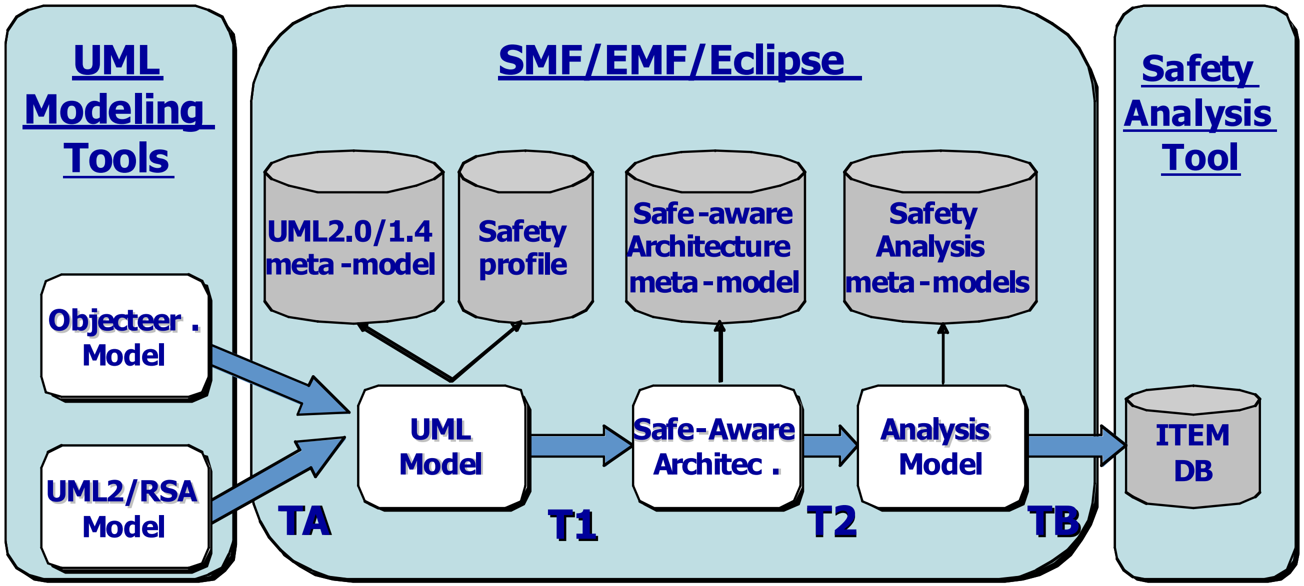
QoS Aware Components
Achievements in year 2

- Partners interaction: Paris meeting, OSERTS'06 wshop
- Specification of QoS characteristics:
  aspect-based, UML profiles MARTE and QoS & FT
  - Exercise with a use case (SCADA)
- QoS characteristics: time and safety
- Generation of analyzable models: time and safety
  - Based on UML profile
- Run-Time QoS support in components frameworks:
  CCM & Robocop
- Standardization: UML MARTE, QoS & FT, M3W, QoS4CCM

QoS Aware Components

Generation of Analysable Models

- Definition of a strategy for modelling and evaluation of QoS and time requirements
- Creation of transformation tools to generate analysable models for **safety** (QoS property)



QoS Aware Components

18-month Plan & Future Evolution

- Continuation of the on-going works
  - Specification of QoS characteristics

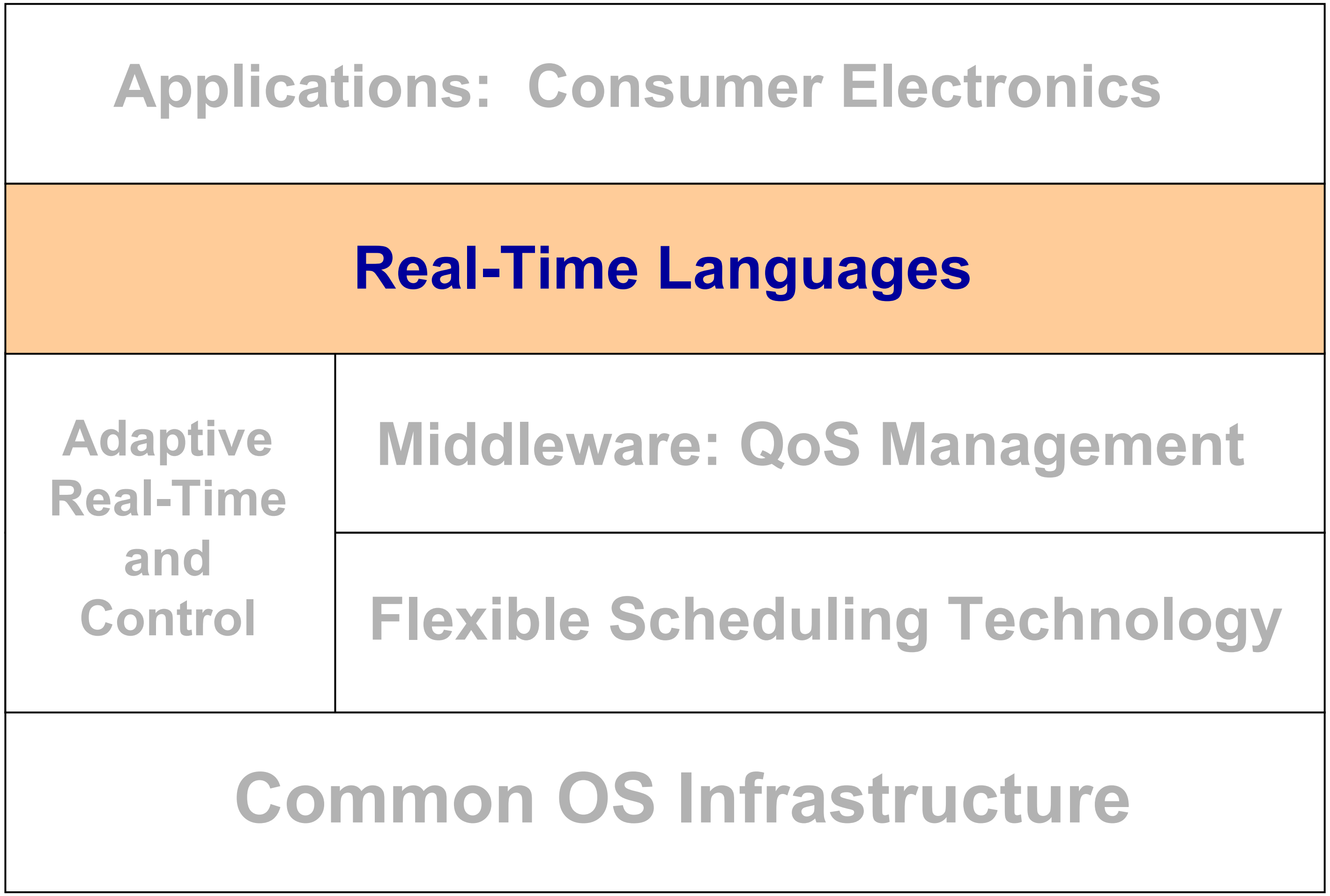
    Use different approaches in the use case

    Compare approaches and derive use guidelines
  - Generation of analysable models

    Support for time and safety characteristics
  - Components framework

    Compare approaches and common definition of required facilities
  - Standardization efforts

New activity



Real-Time Languages

Activity Leader: Alan Burns (Univ. of York)

Objective

- Support RT functionality via language constructs rather than OS calls and hence eases the programmer's task in writing complex flexible real-time applications.
- Combine the efforts and skills of the leading groups to define flexible RT technologies that support multi-language development.
- Initial focus will be on **Ada**, **Java**, and **POSIX** standards.

Expected Results

- Modifications to the standards (where appropriate)
- Development of patterns for the use of language abstractions
- Development of guidelines for using RT programming languages

Real-Time Languages

Activity Leader: Alan Burns (Univ. of York)

Achievements

- Activity only started in March 2006
- Workshop on Ada 2005
  - 22/23 March (York)
- Major workshop on Scoop
  - CORDIE 2006
  - Considered concurrent, real-time and distributed programming in an Eiffle-like language
  - 4/5 July (York)
- Initial work to build a library of reusable facilities to ease the programming of flexible real-time systems (in Ada 2005)

Real-Time Languages

Activity Leader: Alan Burns (Univ. of York)

Plans for next 18 Months

- Major workshop on Java Technologies for Real-Time and Embedded Systems (JTRES)
  - Has actually now taken place – 10-13 Oct 2006 (Paris)
- Plan and participate in the 13th IRTAW
  - International Real-Time Ada Workshop
  - Will take place in US in April 2007
  - Will focus on the new functionality defined in Ada 2005
- Develop and publish via a web site an initial set of patterns (repository) for use by Ada 2005 application programmers.

Overview of the activities of the ART cluster

Applications: Consumer Electronics

Real-Time Languages

Adaptive Real-Time and Control

Middleware: QoS Management

Flexible Scheduling Technology

Common OS Infrastructure