Seminar and Hands-On on Orocos

Peter Soetens, FMTC
Flandres’ Mechatronics Technology Centre
Leuven

5th July 2006
V Jornades de Programari Lliure
Barcelona
Outline

1. Orocos Applications
   - Machine Control
   - Adaptive Control
   - Shared Control
   - Vision Integration
   - Distributed Control

2. Orocos Introduction

3. Use Case: “RoboCatcher”

4. Getting Started with Orocos
Outline

1. Orocos Applications
   - Machine Control
     - Adaptive Control
     - Shared Control
     - Vision Integration
     - Distributed Control

2. Orocos Introduction

3. Use Case: “RoboCatcher”

4. Getting Started with Orocos
Constraint Based Control
Outline

1. Orocos Applications
   - Machine Control
   - Adaptive Control
   - Shared Control
   - Vision Integration
   - Distributed Control

2. Orocos Introduction

3. Use Case: “RoboCatcher”

4. Getting Started with Orocos
1. Orocos Applications
   - Machine Control
   - Adaptive Control
   - Shared Control
   - Vision Integration
   - Distributed Control

2. Orocos Introduction

3. Use Case: “RoboCatcher”

4. Getting Started with Orocos
Placing a Car Window
1. Orocos Applications
   - Machine Control
   - Adaptive Control
   - Shared Control
   - Vision Integration
   - Distributed Control

2. Orocos Introduction

3. Use Case: “RoboCatcher”

4. Getting Started with Orocos
Milling a Human Bone
Outline

1. Orocos Applications
   • Machine Control
   • Adaptive Control
   • Shared Control
   • Vision Integration
   • Distributed Control

2. Orocos Introduction

3. Use Case: “RoboCatcher”

4. Getting Started with Orocos
Distribution in progress...

Middleware for Machine Control

OS

Device
Orocos Introduction Presentation

Freely available on:
http://www.orocos.org
The Orocos Component: “TaskContext”
RoboCatcher: Research Application
RoboCatcher: Requirements

- **Software Framework**
  - Modular - Component Based
  - Online reconfiguration
  - User interactivity
  - Real-Time

- **Application Code**
  - Camera capturing
  - Car image recognition
  - Car state estimation
  - Online trajectory generation
  - Robot kinematics algorithm
  - Robot/gripper interfacing
RoboCatcher: Without Orocos

"Classical" Component Application Setup

User Application

Camera Driver

Image Recognition

Kalman Filter

Trajectory Generator

Robot

Gripper

Kinematics

Target Frame Input

Joint Position Output

Tool Frame

Joint Vel. Output

Gripper Control
RoboCatcher: With Orocos

"Orocos" Component Application Setup

Schedule execution

Camera Driver

Image Recognition

Kalman Filter

Image

Car Location

Trajectory Generator

Extra-polated Location Velocity

Joint Vel. Output

Robot

Provide Services

React to Events

Execute Application Logic <<Real-Time>>

Kinematics

Exchange configurations

Gripper
RoboCatcher: Camera

Data Flow:
Image Connection

Execution Flow:
'methods' and 'commands'

bool fetchImage()
command

Image Data Port

Image Recognition

Image Data Port

Image* getImage()
method
Methods: Synchronously \textit{call}

- Recognition Component (a)
- Call Method: `getImage()`
- Camera Component (b)

Component Activity
Camera Commands

Commands: Asynchronous send

- Component Activity
- Check
- Completion Condition: imageFetched()
- Send Command: "fetchImage"
- Execute Command: fetchImage()
- Recognition Component (a)
- Camera Component (b)
- Command Queue
Methods: Synchronously *call*

- Component Activity
- Recognition Component (a)
- Call Method: getimage()
- Camera Component (b)

Commands: Asynchronous *send*

- Component Activity
- Recognition Component (a)
- Send Command: "fetchimage"
- Check Queue
- Check Completion Condition: imageFetched()
- Execute Command: fetchimage()
- Command Queue
RoboCatcher: With Orocos

"Orocos" Component Application Setup

[Diagram showing the RoboCatcher system with components and connections]

1. Camera Driver
2. Image Recognition
3. Kalman Filter
4. Trajectory Generator
5. Robot
6. Gripper
7. Kinematics
8. Exchange configurations
9. Schedule execution
10. Execute Application Logic
11. React to Events
12. Provide Services
13. Extra-polated Location Velocity
14. Joint Vel. Output
RoboCatcher: With Orocos

"Orocos" Component Application Setup
RoboCatcher: Application Logic

Application Component

Off

Startup

Shutdown

Position Robot

MoveToCar

SafeStop

Real-Time State Machines

Track& Grasp

Real-Time Communication
Diving Into the Application Code

- Task Browsing
- State Machines
- Interactive Scripting
- Creating your first Component
Conclusion

Orocos offers

- a software toolkit for building real-time components
- rich online browsable component interface
- user defined real-time state machines

Further Reference:

http://www.orocos.org