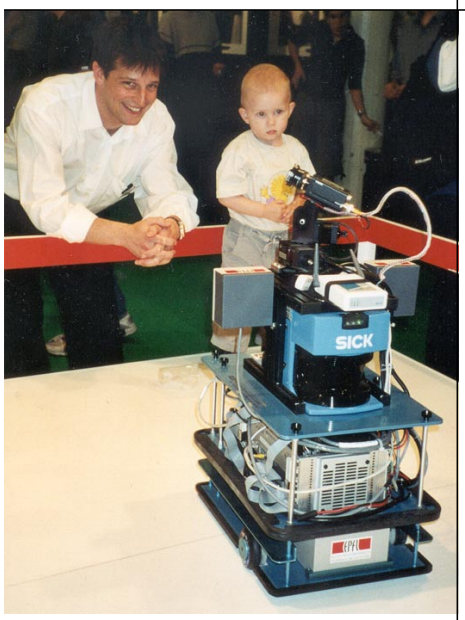
Reliability in Mobile Robotics

Dr. Nicola Tomatis

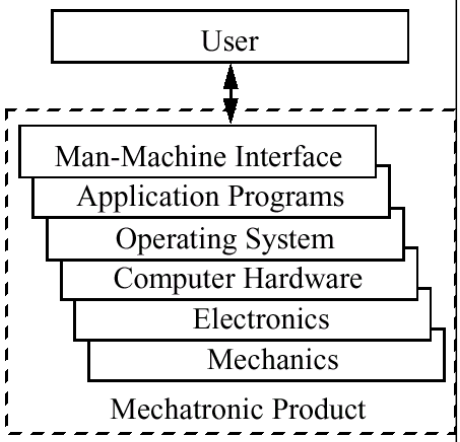
Autonomous Systems Lab

Contents



Introduction

Mobile Robotics



- Mechatronics:
  - Mechanics
  - Electronics
  - Software
- Quality robot product requires quality COTS embedded products:
  - √ Mechanics
  - √ Electronics
  - × Software

Motivation

Embedded Software

- Lack of **safety**
- Lack of **tools**
- Slow **development cycles**
- Difficult **integration**
- Lack of portability, abstraction & reuse
- **Errors** (especially memory):
  - Spurious accesses
  - Memory leaks
  - Dangling pointers
  - Un-typed memory handling

Operating System

ETH Zürich Institute of Robotics IfR

The XO/2 Hard Real-Time OS

XO/2 REAL-TIME OS

- Safety
- Memory management
- Real-time process scheduling
- Exception handling
- Ergonomics

Operating System

Safety Definition

XO/2 REAL-TIME OS

- **Safety**:
  - Nothing bad happens
- **Progress**:
  - The right things do (eventually) happen
- **Security**:
  - Things happen under proper authorization (potentially bad things happen under proper supervision)

Operating System

Memory Management

- Static *safety*:
  - **Strong-typing**
  - Index-checks
  - …
- Dynamic *safety*:
  - Dynamic type-systems
  - Real-time compatible mark-and-sweep garbage collector (**automatic memory reclamation**)

Operating System

Processor Scheduling

- Hard real-time (*progress*):
  - Producing **correct result** while meeting pre-defined **deadlines**
  - Earliest-deadline-first scheduling algorithm
- Two types of tasks:
  - Real-time
  - Non real-time (threads)

Operating System

Exception Handling

- Supervision (*security*):
  Each **exception** appears as a “TRAP” **under control** of the operating system
- Exception handling (*security*):
  Each task has a **HandleExeption method** which allow to run code for exception handling as soon as a TRAP appears

Operating System

Ergonomics

- Static *safety*:
  - **Interface checking** at compile-, linking- and loading-time
  - **Safe unloading**
- Very **short** edit-compile-test **cycles**

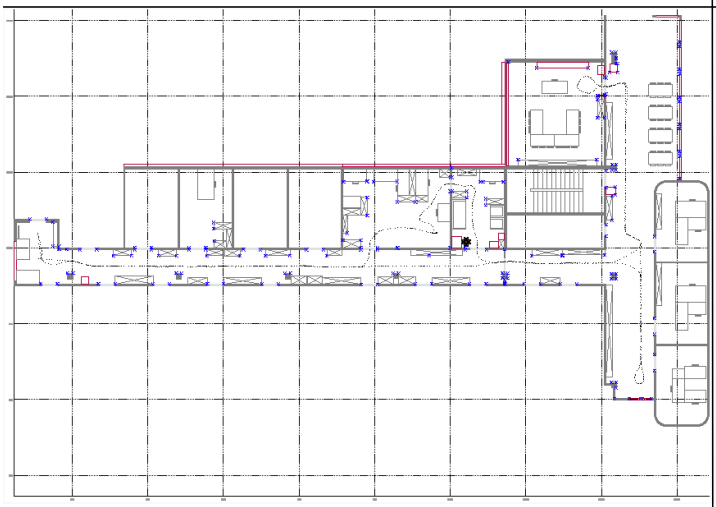
Programming Language

"The safer the tool, the more reliable the system."

Oberon-2

- **Strong-typing**: each allocated (statically or dynamically) object is bound to a type
- **No** memory **reclamation** (left to the system-wide garbage collector)
- Compatibility by name (not by structure)
- Object-orientation
- Modularization

Example: ASL Research

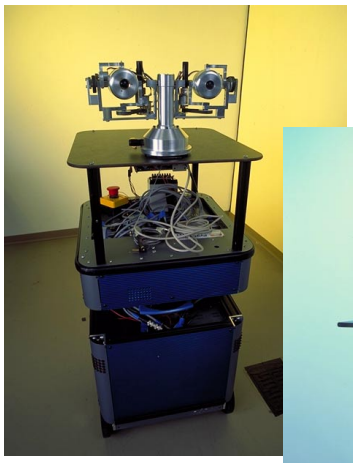


Navigation

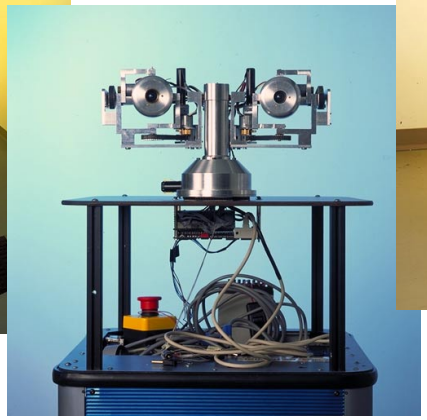
- More than 100 km traveled in a large, complex and dynamic environment
- More than 10 km experiments with no lost situations
- Errors of less than 1 cm

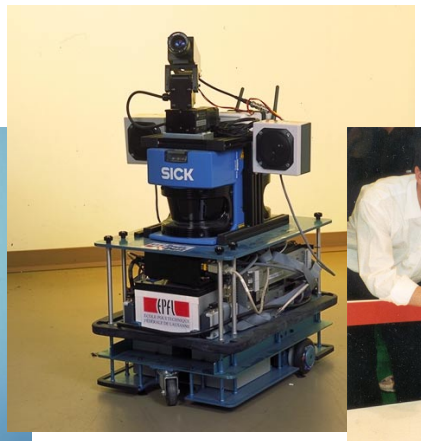
Example: ASL Research

Physical Interaction

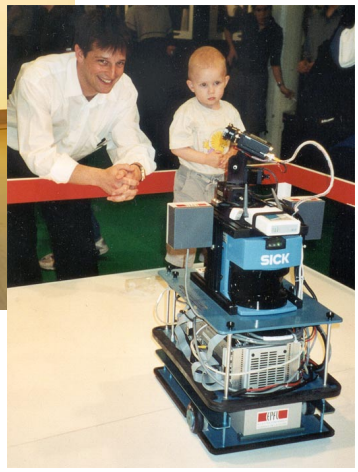


Pygmalion



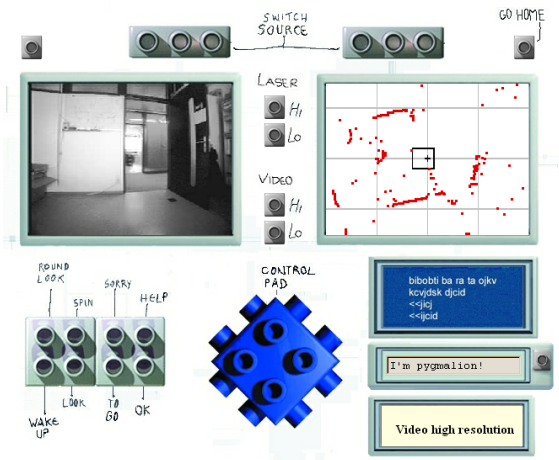


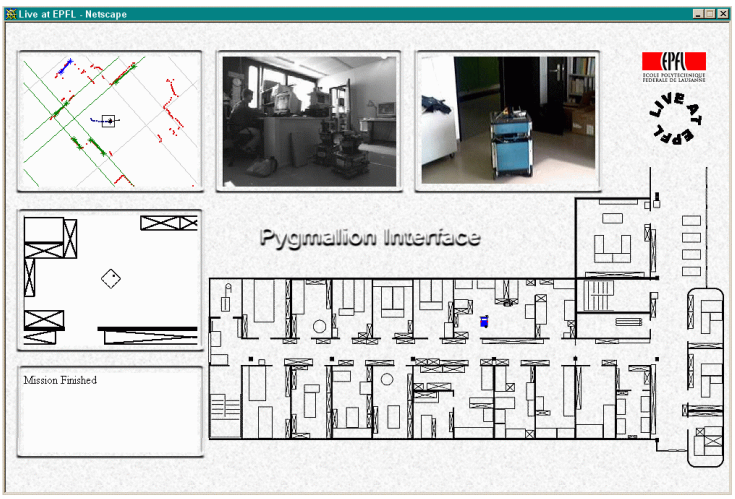
Daffy-Duck



Example: ASL Research

Web Interfacing





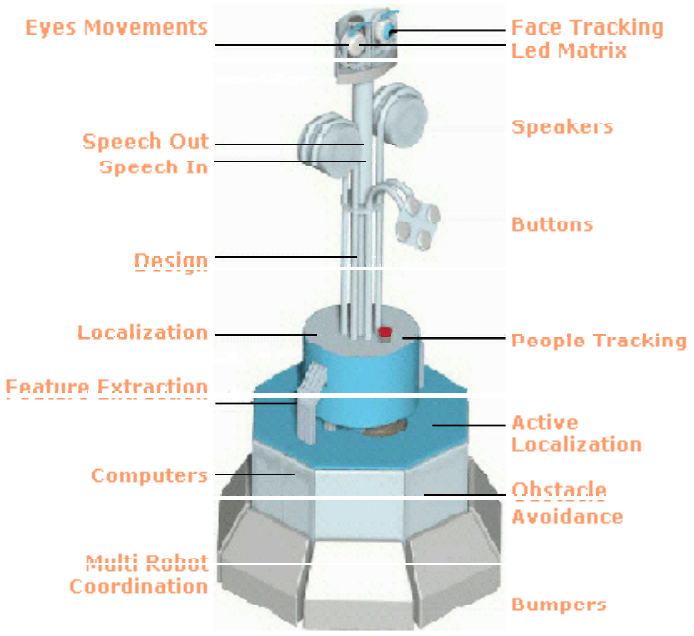
Conclusion

- Mobile robots are complex mechatronic products needing:
  - Safety
  - Reliability
  - Robustness
- This is achieved by employing:
  - Industry-proof products for mechanics and electronics
  - Experience in designing and building
  - XO/2 as operating system and development environment

Outlook

Autonomous Systems Lab

BLUEBOTICS
Mobile Robots at your Service



The **Robotics** project:
- 6 months
- 10 hours each day
- 10 robots

Development status:
- Two prototypes
- > 140 hours operation
- > 11 km navigation
- MTBF: 4h15min

http://robotics.epfl.ch/