Safety in Robotics:
The Bremen Autonomous Wheelchair

Axel Lankenau, Oliver Meyer and Bernd Krieg-Brückner

Bremen Institute of Safe Systems

TZI, University of Bremen, Germany

Outline

Safety-Critical Systems and Robotics

- Definition of safety
- Rehabilitation robots as safety-critical systems

The Bremen Autonomous Wheelchair

- Hardware
- System architecture
- Formal design approach
  - *Fault tree based hazard analysis*
  - *Safety requirements and safety mechanisms*
- Verification by model-checking

Application Module: The DrivingWizard

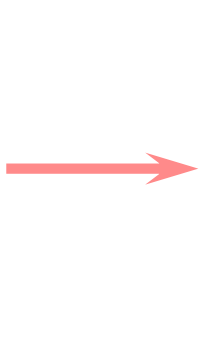
Future Work

Safety-Critical Systems

Safety-Critical Systems

- Ensure: no *catastrophic consequences* on the environment occur
- Examples: power plant, railway interlocking system

Rehabilitation Robots as Safety-Critical Systems

- Safety requirements:
  - *No collisions*
  - *User commands must be executed*
- Classical safety + availability + reliability

The Autonomous Wheelchair “Rolland”

Technical Data

- Meyra “Genius 1.522”
- 6 km/h top speed
- Control: serial interface

Sensors

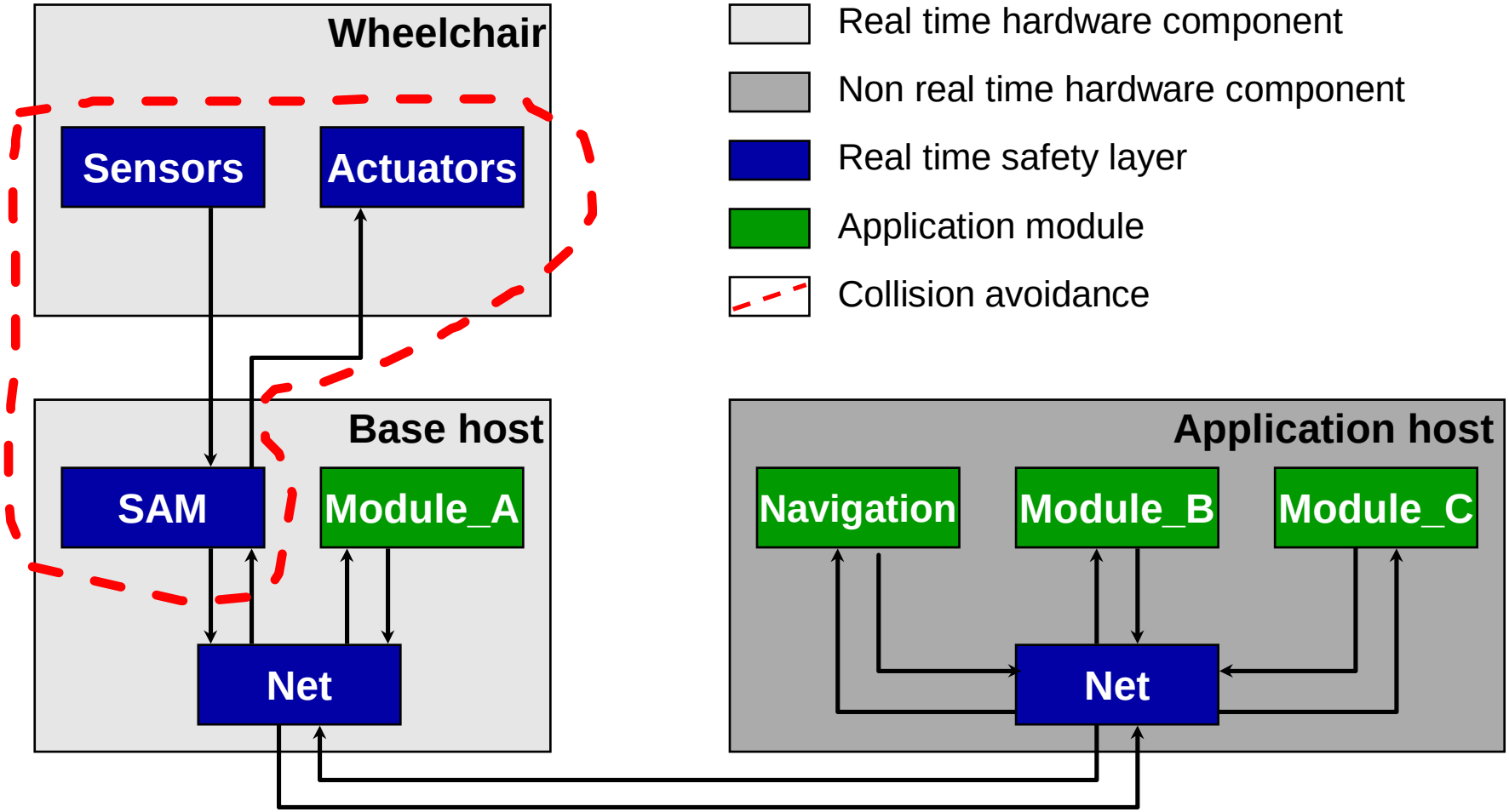
- 27 Nomad sonar sensors
- internal sensors
- [up to 16 Nomad IR-sensors]
- [laser scanner]
- [video camera]

Computing

- Industrial-PC (Pentium 133)
- QNX (real time OS)
- network for additional PCs

Universität Bremen TZI

System Architecture *Safe Wheelchair*



Formal Design Approach

Hazard Analysis

- Fault tree analysis
- Specification of undesired system behaviour

Derivation of Safety Requirements

- Specification of the environment
- Specification of safety properties

Definition of Safety Mechanisms

- Controller ensuring system safety
- Potential introduction of new hazards caused by the controller

Verification of Safety Properties

Fault Tree Based Hazard Analysis

Fault Tree Segment (Problems of External Sensors)

```
X Failure of external sensors
  | X.1 Measuring error that may cause a collision
       & X.1.1 Too large values measured by sensors
            | X.1.1.1 Too large values measured up
                      to n consecutive times.
            | X.1.1.2 Too large values measured more
                      than n consecutive times #
       & X.1.2 Obstacle distances overestimated.
  | X.2 Disastrous breakdown of external sensors
       & X.2.1 No distances measured #
       & X.2.2 Breakdown not detected.
  | X.3 Obstacle not detectable by external sensors #
```

Specification of the Environment

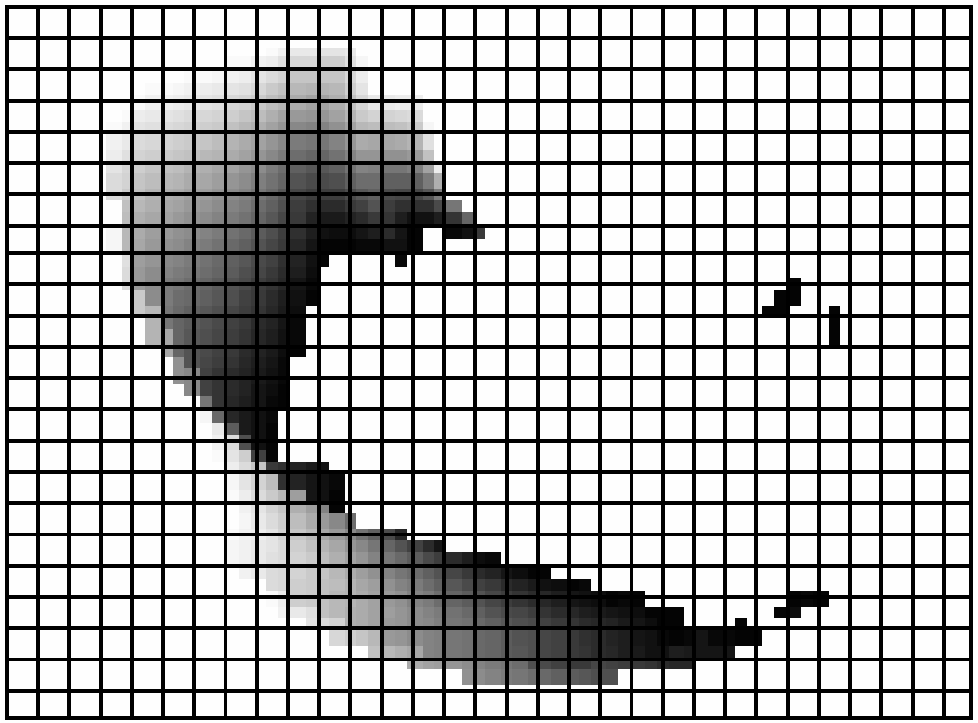
Fault Tree Leaves:

```
Safety Requirements of the system not satisfied
| 1 Collision (at sensor level)
    | 1.1 Passive collision #
    | 1.2 Active collision
    ...
| 2 Crash (not at sensor level) #
| 3 Downfall #
...
```

Requirements Imposed on the Environment:

- No “active” obstacles
- Maximum horizontal extent of every obstacle at sensor level
- No stairs etc.

Virtual Sensors



Storing Measurements

- Immune to temporary measuring error/failure
- Transformation of map

Virtual Sensors

- Depending on rotation, direction and steering
- Anticipation of collisions with respect to the ***map***
- Calculation of distance considering movement

“Aging” of Measurements

- Dynamic obstacles are “forgotten” after a while
- Maximum “lifetime” of obstacles in the map specified in the fault tree

Safety Requirements & Safety Mechanisms

Derivation of Safety Requirements

- Negation of the leaves of the fault tree
- Depending on the logical relation of hazards

Safety Mechanisms: Fault Tolerance vs. Prevention

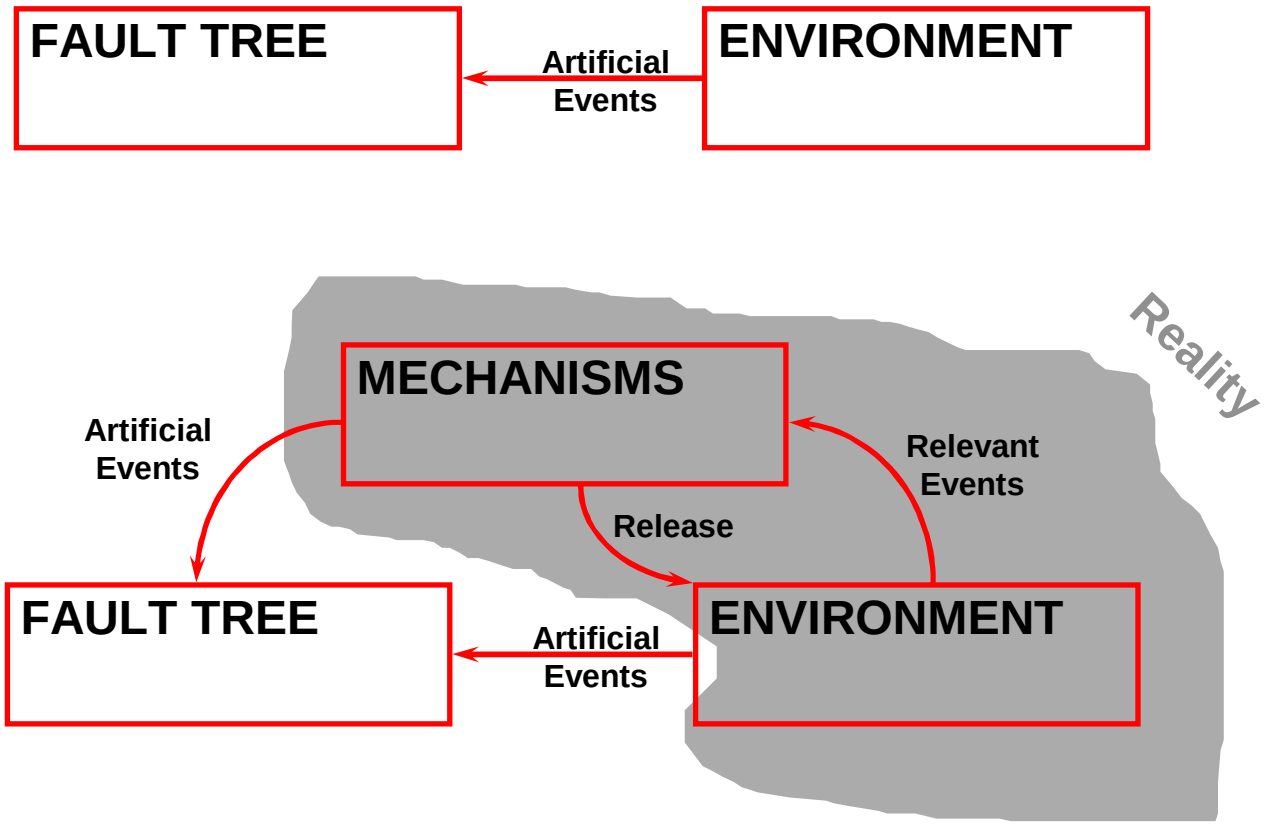
- Virtual sensors operating on a local occupancy grid map
- No collisions on the network due to the TDMA frame-protocol

Controller Implements Safety Mechanisms

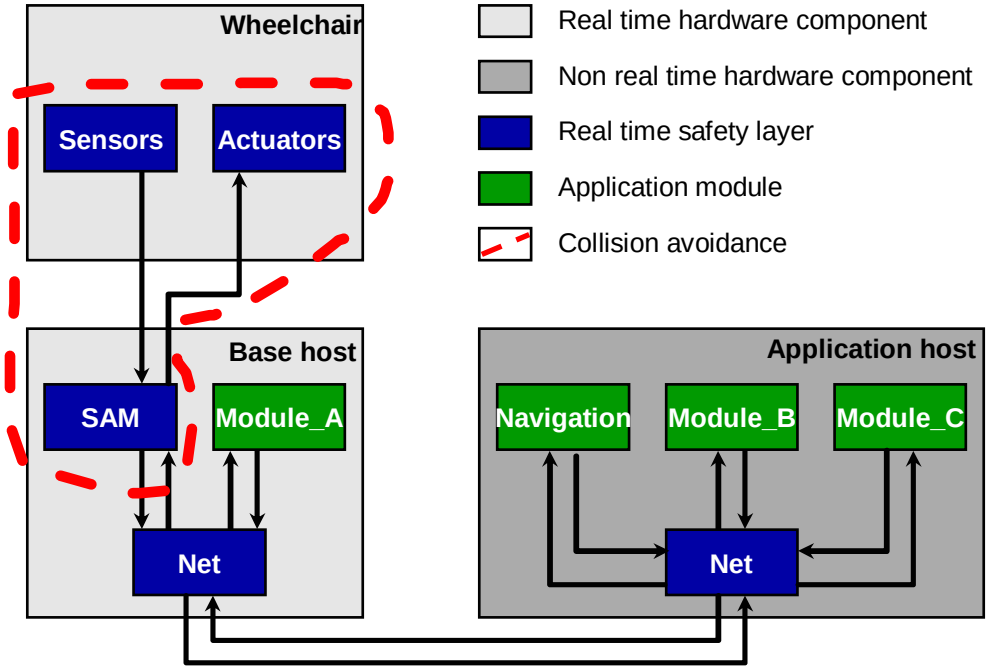
- Sensor parameters as inputs
- Influences physical system via actuator parameters
- Transition of physical system into a hazardous state prevented

Formal Verification by Model-Checking

Implemented Fault Tree as Arbiter



The DrivingWizard (Speed & DoorWizard)



SAM

- Danger: STOP!
- Acts at latest point in time
- Small high-resolution map

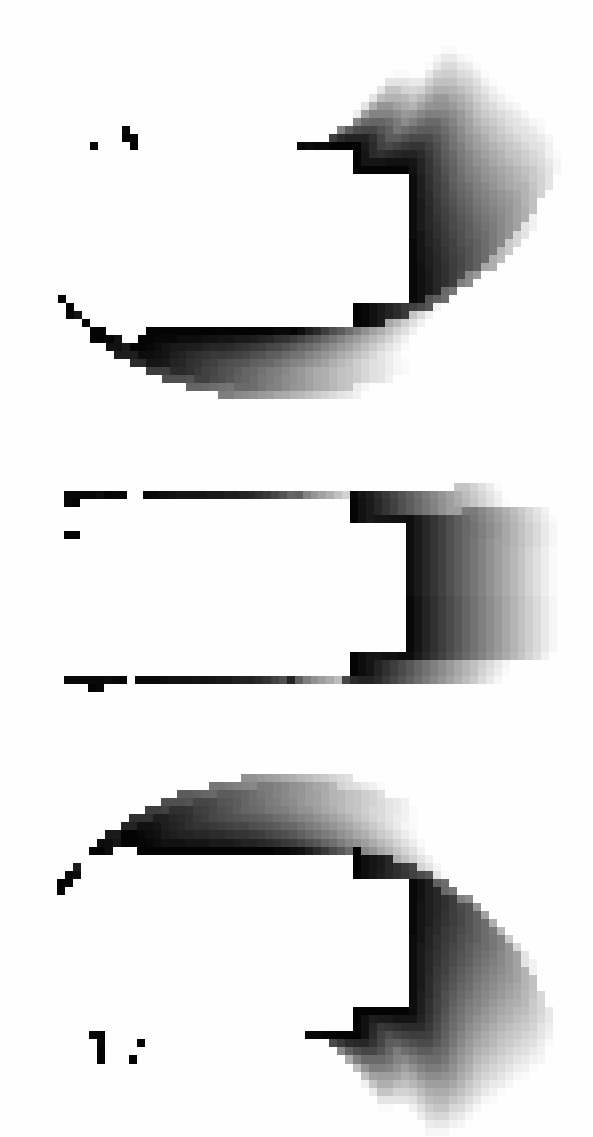
SpeedWizard

- Danger: Slow down!
- Acts as early as possible
- Large low-resolution map

Basic Idea (SpeedWizard)

- Set speed to the largest possible value below the user‘s command
- After decelerating, accelerate comfortably to the target speed if no obstacles are present

The DoorWizard



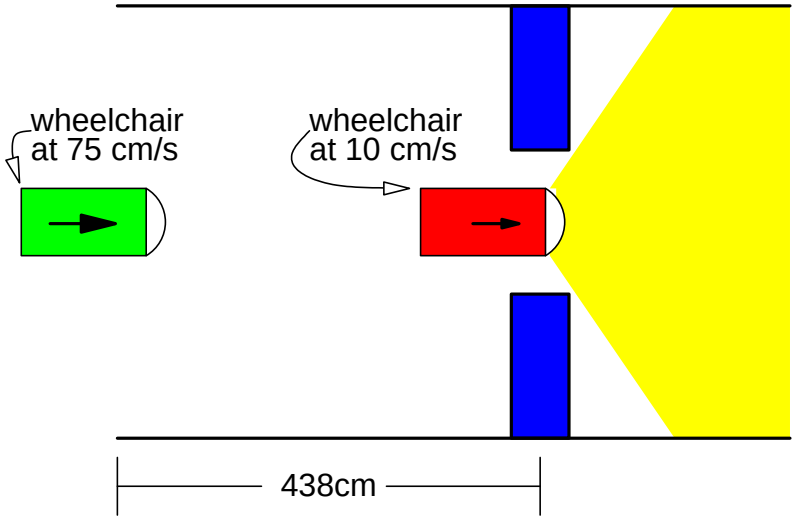
Basic Idea

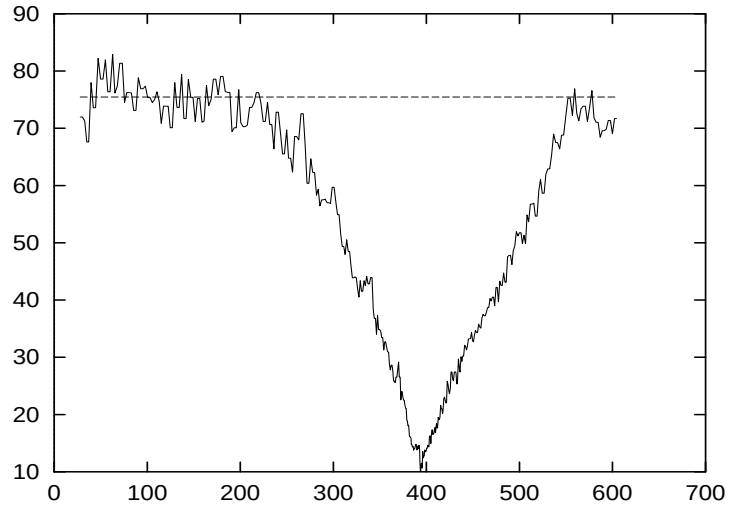
- Takes over control when approaching a door
- Uses SAM-knowledge (virtual sensors)
- Chooses first “promising” steering angle

Extensions

- Permanent use
- Obstacle avoidance possible
- Planning of trajectories

Experimental Results (SpeedWizard)





Collision Avoidance

- SAM ensures that no collision occurs as long as the requirements imposed on the environment hold

Door Passing

- Eased by SpeedWizard
- Automated by DoorWizard

Future Work

Formal Verification of “Robotic Issues”

- Problem: Modelling of the environment is very complex
- Automated Testing

Various Modules for Handicapped Users

- DrivingWizard
- Commands by speech recognition
- Special automated manoeuvres: turns, docking to a table etc.

Route Navigation

- DFG Priority Program Spatial Cognition
- Video camera, laser scanner