# Perspectives on COSMIC

Michael Schulze and Sebastian Zug

*mschulze,zug@ivs.cs.uni-magdeburg.de*

Department of Distributed Systems (IVS)
Otto-von-Guericke-University Magdeburg
Universitätsplatz 2
D-39106 Magdeburg

Minema Workshop, Trento, 08.06.2007

# Outline

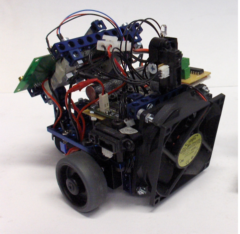
# Presentation of our working group EOS

## Chair Prof. Dr. Jörg Kaiser

Otto-von-Guerike Universität Magdeburg
Department of Distributed Systems
Embedded Systems and Operating Systems (EOS)

## Phd students

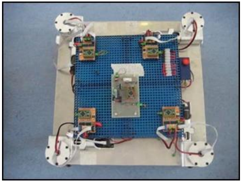
- Michael Schulze
- Sebastian Zug
- Thomas Kiebel

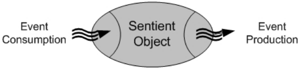


# Idea

## Applications as cooperation of sentient objects

- The system consists of distributed heterogeneous hardware connected by different networks.
- The components are autonomous subsystems and exhibit spontaneous behaviour.
- Message transmission and receive is safety critical sometimes.
- The connections are variable.





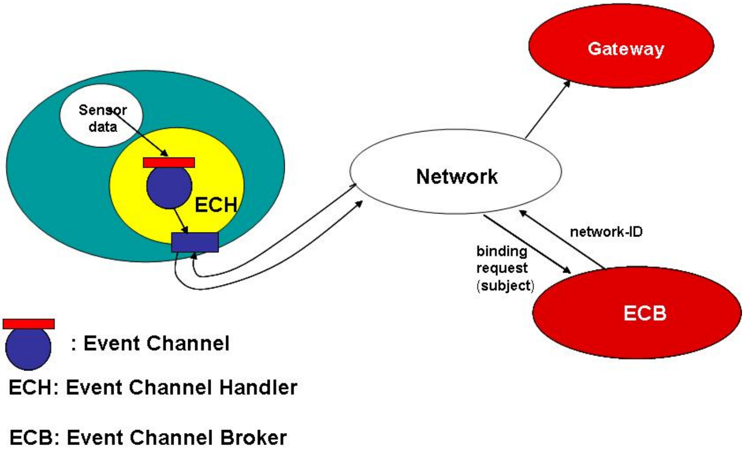

A middleware is necessary !

# COSMIC Middleware

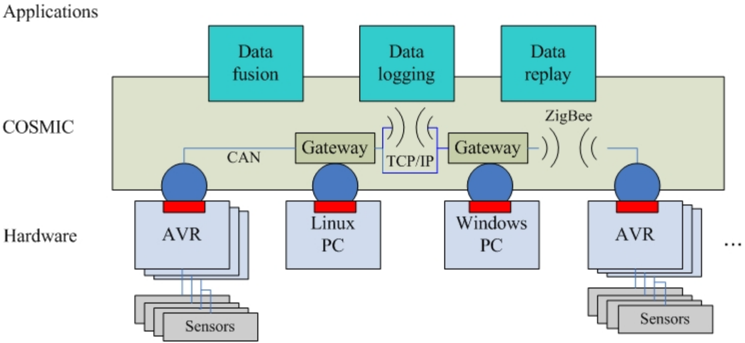
## Cooperating smart Devices

- Event based communication
- Event channels as abstraction of different networks
- Publish / Subscribe mechanism
- Different Real-time Levels

# COSMIC communication architecture



# Current structure



# Application view

## Applications want/require . . .

- . . . always the same communication interface, but often different . . .
  - . . . controller types (8-32Bit)
  - . . . communication media (field bus, ethernet, wireless)
  - . . . operating system software
- quality of service
  - robustness
  - fault-tolerance
  - soft/hard real-time

# Implication of application requirements



## The approach

- adaptable system software
- configuration and composition to application needs
- tailorable to system requirements

# Requirements on/features of the middleware I

## Functional requirements

- adaptation to communication media/driver and OS
- address binding
- subject binding
- if more than one com. media ...
  ... additional demands
  - Routing
  - Scoping
  - Filtering
- time synchronisation (rt)
- scheduling (srt)

# Requirements on/features of the middleware II

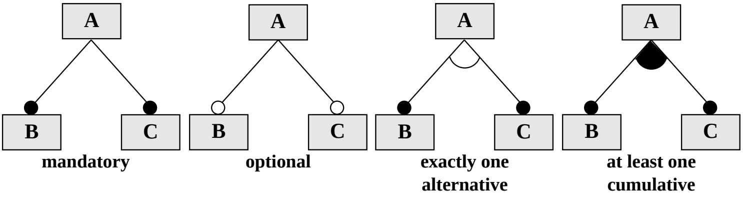
## Non-Functional requirements

- as small as possible, due to some $\mu C$
- real-time
  - effect on functional demands
- robustness
- fault-tolerance

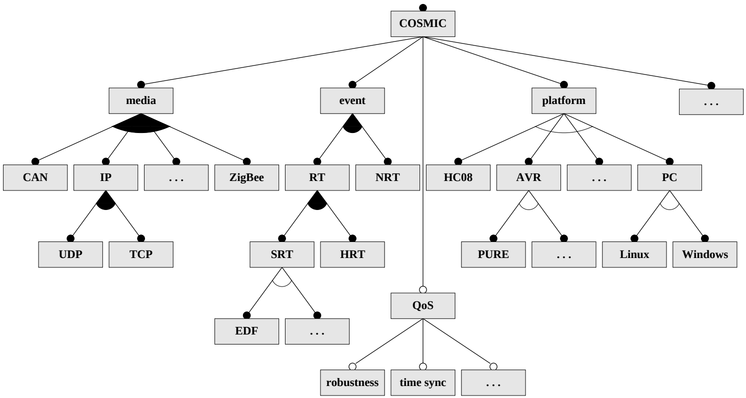
# Structuring the possibilities

## Feature Models[5, 2]

- property description on an abstract level
- functional and non-functional features representable
- common vs variable features
  - common feature are nodes
  - variable feature are leaves



# Feature Model of COSMIC



# Separation of concerns



- Things which don't related with each other have to be treated and to be implemented separately.
- However, some demands can be separated difficultly because of their cross-cutting character.
  - global system strategies
  - non-functional requirements

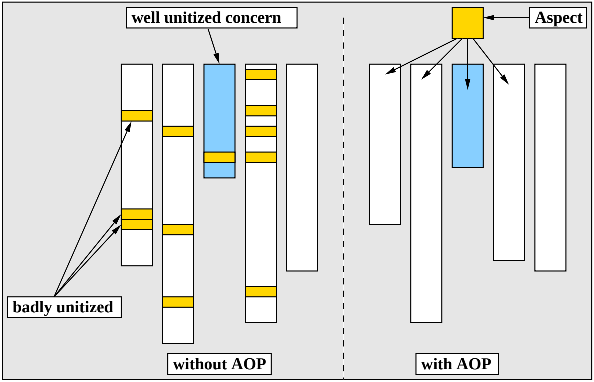
## Solution $\Longrightarrow$ AOP

- separation of cross-cutting concerns

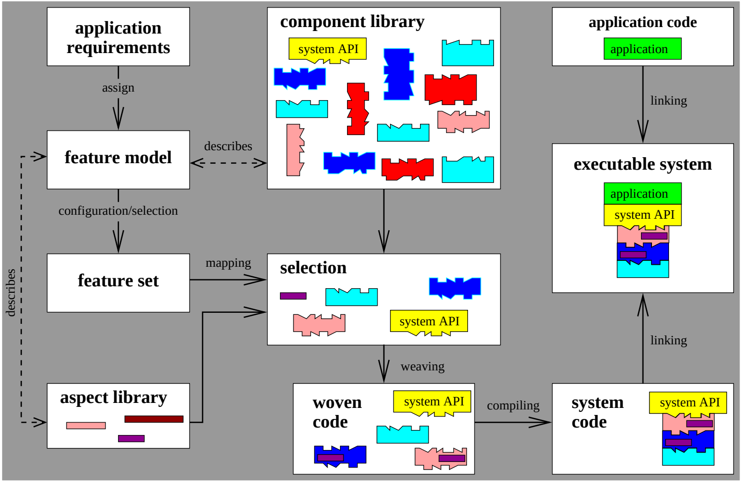
# Aspect-Oriented Programming [6]

## Aspect

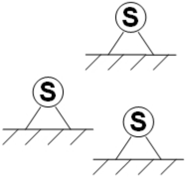
- modular implementation of a cross-cutting concern
- can act on/in functions or other aspects
- can have logical dependencies



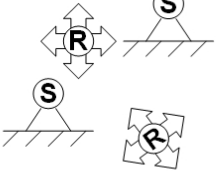
# Construction of the system



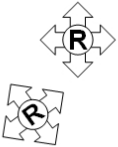
# Applications




Sensor networks




Combined systems




Robotic swarms

challenge: fusion of time shifted information based on varying kind and number of sensors

## Main Objectivs

- Definition of the problem related data
- Analysis of available information
- Tracing of sensor data
- Adaptable fusion strategies

# Steps

## On design time

- Definition of output events / results
- Specification of the required information
- Time and performance elastic computation

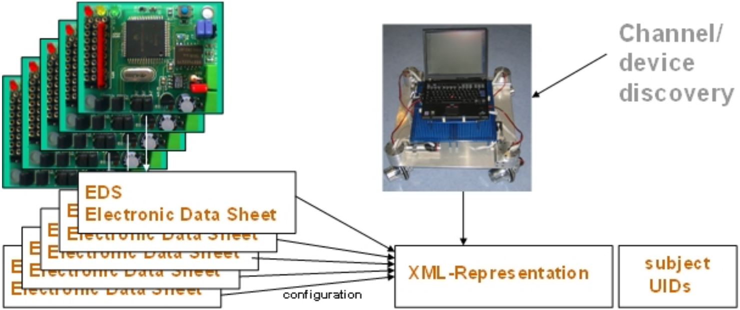
## On runtime

- Discovery of available data sources
- Selection of the events
- Evaluation of the temporal validity
- Estimation of current states
- Adaption of a fusion model
- Fusion

# Available Information

## Self describing Components

- Electronic data sheets (TEDS)
- Toolchain to integrate sensors




[]

# Definition of the required information

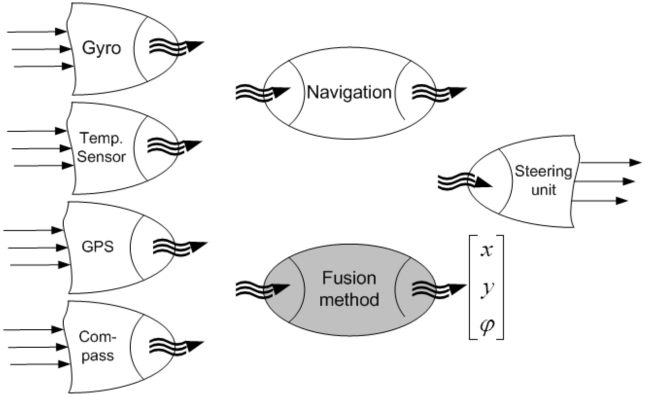
## Method

- a lot of possible sensors
- different positions/directions
- varying quality of information elements

Description of „usefull“ events

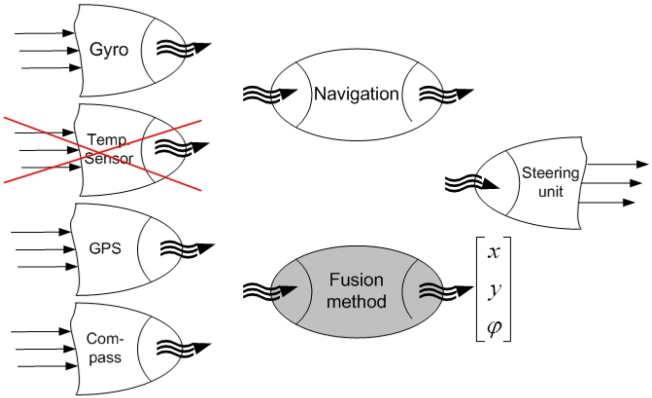
```
event:= <subject, attributes, data>
<temperature senors, [position quality]>
<compass sensor, [timestamp]>
```

# Example - position estimation for mobile system



# Example - position estimation for mobile system

```
<[compass sensor, gps, gyro], [timestamp]>
<acceleration sensor, [timestamp, orientation, position]>
```



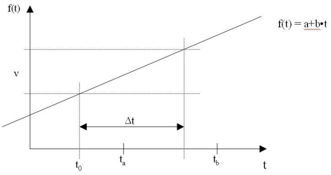
# Temporal Validation and Tracing of sensor data



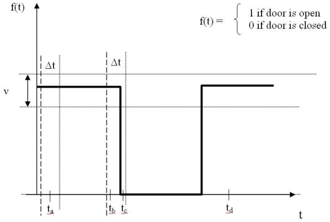
## Defined on compile time

$$f(t_c) = v(t_a) + \delta \text{ with } t_c > t_a$$

- Temporal consistency
- Temporal validity




$\delta < \nu$

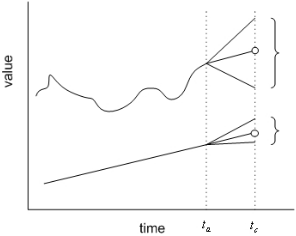



[4]

# Temporal Validation and Tracking of sensor data

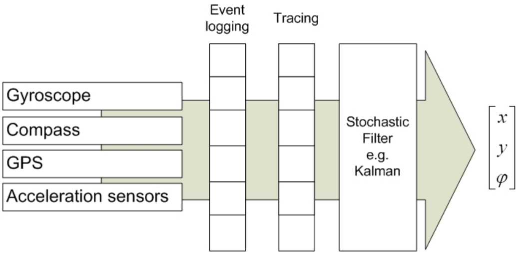
## Adaptive

- continuous analysis of the signal dynamic
- determination of a current variance as bases of temporal validity



# Adaptable Fusion

## One Possibility - Kalman filtering




$$x = Ax + Bu$$
$$y = Cx + Dv$$

# Adaptable Fusion

## Positioning Example

$$
\begin{array}{r}
GPS \rightarrow \\
\\
acceleration \rightarrow \\
\\
compass \rightarrow \\
gyro \rightarrow
\end{array}
\begin{pmatrix} x \\ y \\ \ddot{x} \\ \ddot{y} \\ \varphi \\ \dot{\varphi} \end{pmatrix}
=
\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & dt^2 \\ 0 & 1 & dt^2 \\ 0 & 0 & 1 \\ 0 & dt & 1 \end{pmatrix}
\cdot
\begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \\ \varphi \\ \dot{\varphi} \end{pmatrix}
+ Dv
$$

$$
\begin{array}{r}
GPS \rightarrow \\
\\
compass \rightarrow \\
gyro \rightarrow
\end{array}
\begin{pmatrix} x \\ y \\ \varphi \\ \dot{\varphi} \end{pmatrix}
=
\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & dt & 1 \end{pmatrix}
\cdot
\begin{pmatrix} x \\ y \\ \phi \end{pmatrix}
+ Dv
$$

# Future Work

- Implementation of Matlab routines for
  - Comfortable access to COSMIC
  - Service discovery based on a TEDS description
- Design of methods for adaptable combination of varing sensordata
- Choosing a fusion concept
- further development on the COSMIC component and aspect library
- Implementing mechanisms to tailor down the system
- Providing a framework for configuration of COSMIC
- Development of a decentralized controll for Q based on a data fusion in each node

# Practical Demonstration

# References I


Danilo Beuche, Abdelaziz Guerrouat, Holger Papajewski, Wolfgang Schröder-Preikschat, Olaf Spinczyk, and Ute Spinczyk.
On the Development of Object-Oriented Operating Systems for Deeply Embedded Systems—The PURE Project.
In *Proceedings of the 2nd ECOOP Workshop on Object-Orientation and Operating Systems (ECOOP-OOOSWS'99)*, pages 27–31, Lisboa, Portugal, June 14–18 1999.

Danilo Beuche, Holger Papajewski, and Wolfgang Schröder-Preikschat.
Variability Management with Feature Models.
In *Proceedings of the Software Variability Management Workshop*, pages 72–83, University of Groningen, The Netherlands, February 2003.
Technical Report IWI 2003-7-01, Research Institute of Mathematics and Computing Science.

A. Casimiro, J. Kaiser, and P. Verissimo.
An architectural framework and a middleware for cooperating smart components.
In *ACM Computing Frontiers conference (CF'04), ISCIA*, Italy, April 14–16 2004.

A. Casimiro and P. Verissimo.
Generic timing fault tolerance using a timely computing base.
June 2002.

# References II


Kyo C. Kang, Sholom G. Cohen, James A. Hess, William E. Novak, and A. Spencer Peterson.
Feature-Oriented Domain Analysis (FODA) Feasibility Study.
Technical report, November 1990.

G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin.
Aspect-Oriented Programming.
In M. Aksit and S. Matsuoka, editors, *Proceedings of the 11th European Conference on Object-Oriented Programming (ECOOP '97)*, volume 1241 of *Lecture Notes in Computer Science*, pages 220–242. Springer-Verlag, June 1997.