EVENT DRIVEN PROGRAMMING INTRODUCTION

- Embedded Real Time Systems
- Ron Barker

HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN · FH
MÜNCHEN

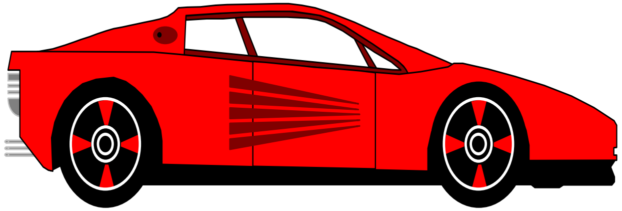
EVENT DRIVEN PROGRAMMING

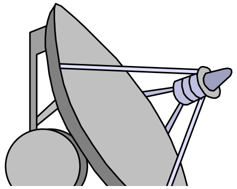
Introduction to Event Driven Real Time Systems

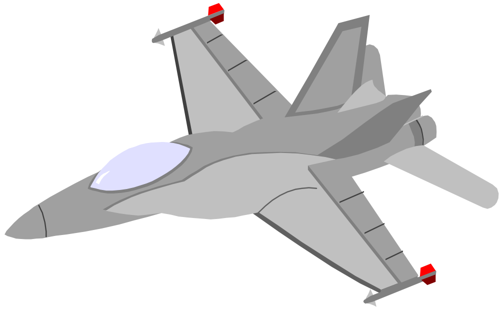
Introduction

- What are embedded event driven systems?
- What makes them different?
- Real time operation
- Many sets of constraints on designs
- Challenges in embedded computing system design.
- Design methodologies.

Where do we find embedded systems?







EVERY
WHERE

Examples

Office systems and mobile equipment	Building systems	Manufacturing and Process Control
Answering machines Copiers Faxes Laptops and notebooks Mobile Telephones PDAs, Personal organisers Still and video cameras Telephone systems Time recording systems Printer Microwave	Air conditioning Backup lighting and generators Building management systems CTV systems Fire Control systems Heating and ventilating systems Lifts, elevators, escalators Lighting systems Security systems Security cameras Sprinkler systems	Automated factories Bottling plants Energy control systems Manufacturing plants Nuclear power stations Oil refineries and related storage facilities Power grid systems Power stations Robots Switching systems Water and sewage systems

How do we define an embedded reactive system

- Embedded system: any device that includes a programmable computer but is not itself a general-purpose computer.
- Computer purchased as part of some other piece of equipment
  - Typically dedicated software (may be user- customizable)
  - Often replaces previously electromechanical components
  - Often no “real” keyboard
  - Often limited display or no general- purpose display device: don’t need all the general-purpose bells and whistles.

Characteristics of an embedded system

Real-Time Operation

- Reactive: computations must occur in response to external events
- Correctness is partially a function of time

Small Size, Low Weight

- Hand- held electronics and Transportation applications -- weight costs money

Low Power

- Battery power for 8+ hours (laptops often last only 2 hours)

Harsh environment

- Heat, vibration, shock, power fluctuations, RF interference, lightning, corrosion

Safety- critical operation

- Must function correctly and Must *not* function *in* correctly

Extreme cost sensitivity

- $. 05 adds up over 1,000, 000 units

Embedded – Reactive Systems

FUTURE OF EMBEDDED SYSTEMS

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

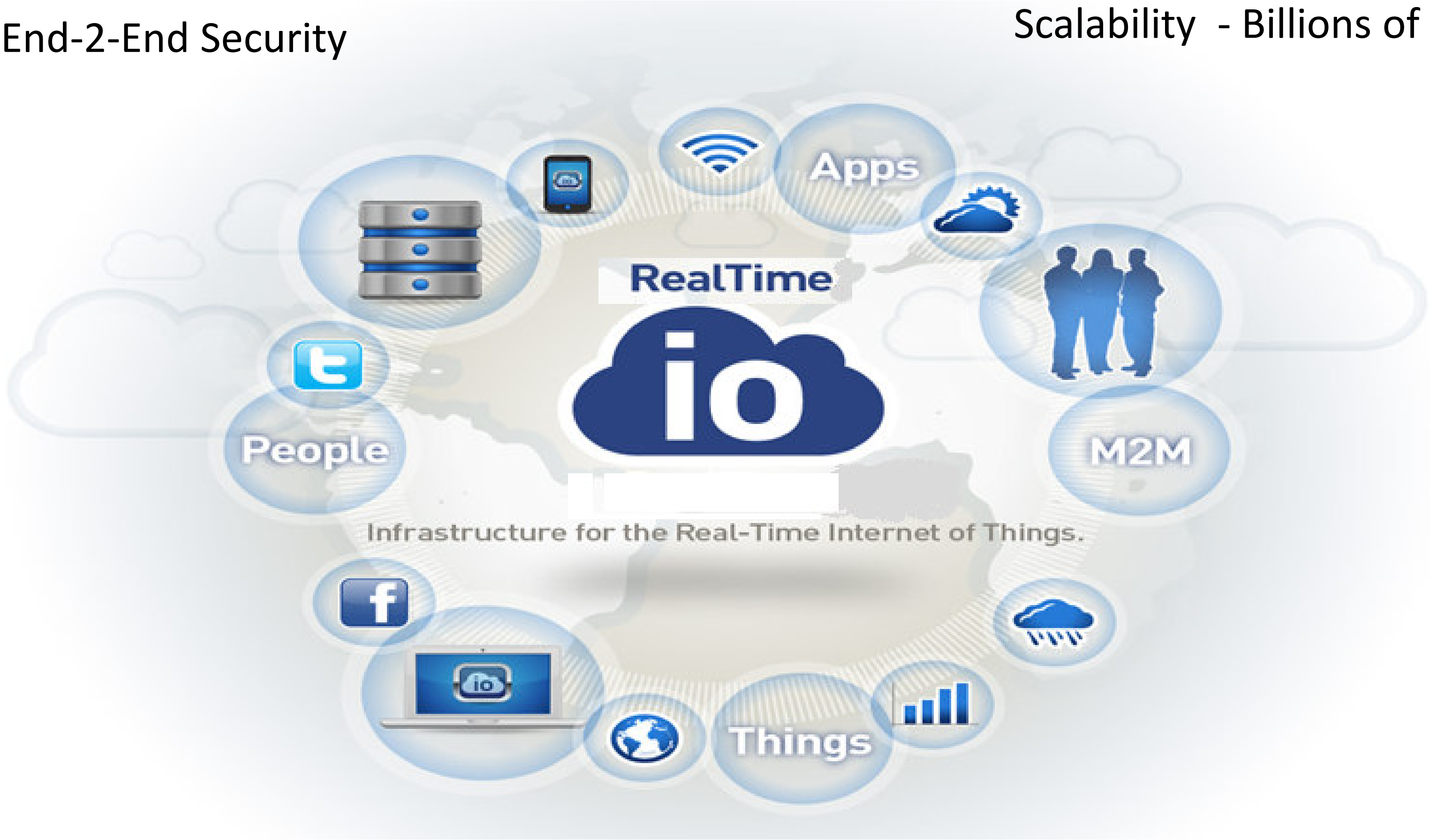
Networking

- **Networks of Embedded Systems – aka as:**
- **Internet of Things**

Internet of Things

End-2-End Security

Scalability - Billions of devices



Stability – Long Product Life Cycles

Affordability– Pay for what you use

HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN · FH
MÜNCHEN

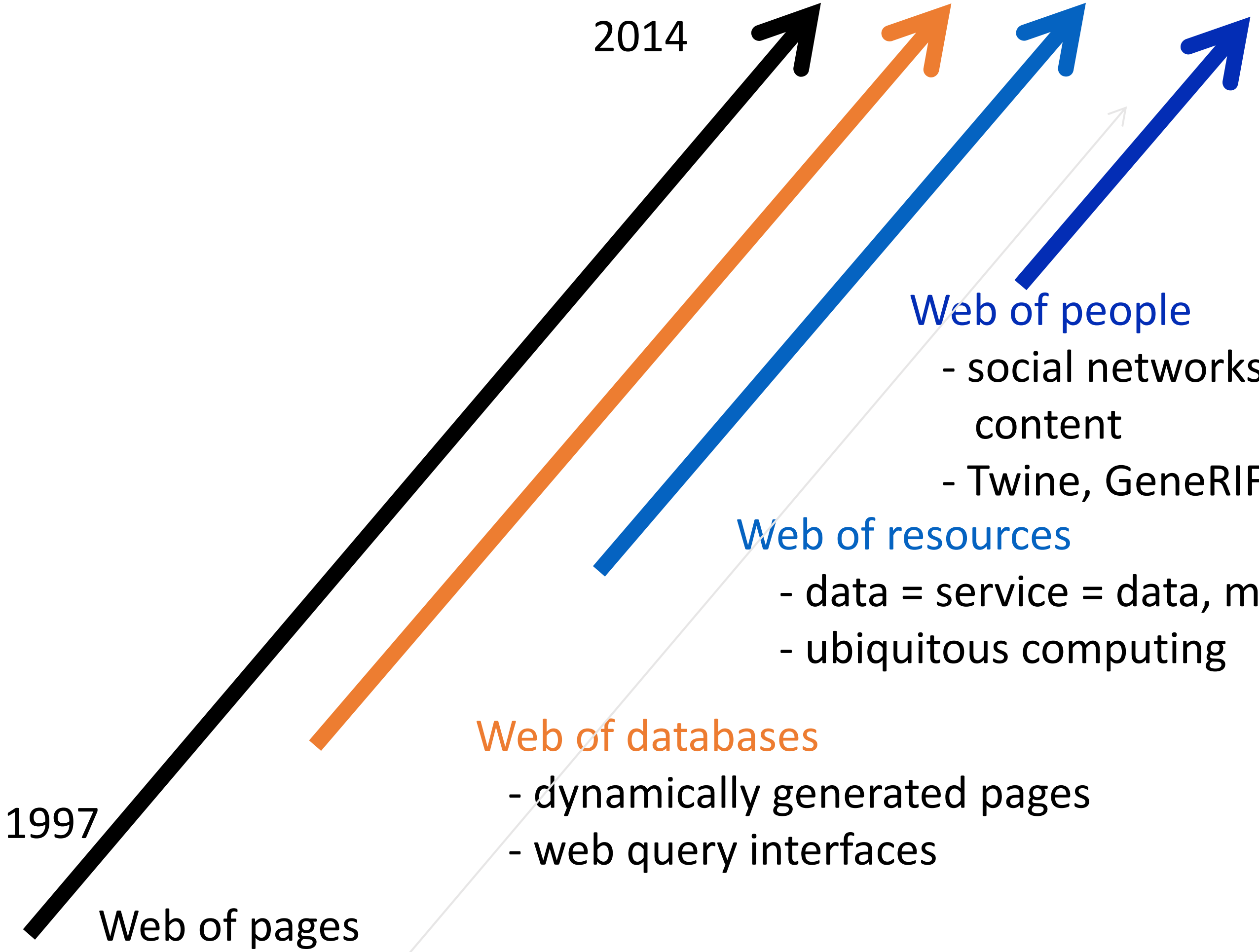
The Programming Paradox

- How to program ?
  - A simple device
  - A complex (SOC)
  - A Network
- How to program for?
  - Low cost
  - High Security
  - Long Term Sustainability
  - Dynamic Product Life Cycles
  - Low Latency – Real Time

The Networking Paradox

The IT Hammer is?

THE Web



Web of Things
-Smart Home
-Smart Grid

Web of people
- social networks, user-created casual content
- Twine, GeneRIF, Connotea

Web of resources
- data = service = data, mashups
- ubiquitous computing

Web of databases
- dynamically generated pages
- web query interfaces

Web of pages
- text, manually created links
- extensive navigation

Today‘s Internet of the WEB

- Asymetric Architecture
- Best-effort - QoS Issues
- Security Issues
  - Heart Bleed
  - Shell Shock
- Stability decaying
  - Buffer Bloating
- Obsolete Client Server Model
- Focus on Content Delivery
- IPv4 Address Exhaustion
- Flat-Rate Cost Pressures

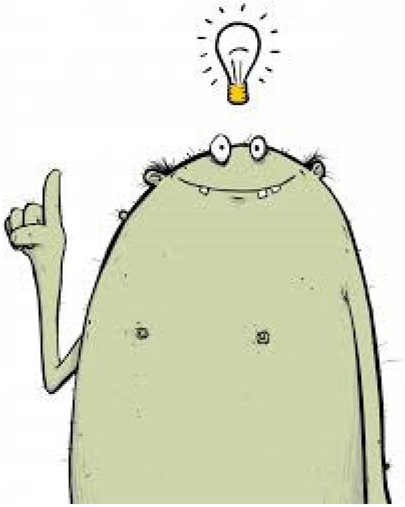
HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN·FH
MÜNCHEN

IP==IT==WWW Syndrome

- IP undocumented „assumed“ communication path
- IP==IT undocumented „assumed“ implemention path
- Dogmatic conviction: WEB Services==Best Practice / Only Solution

WEB of Things vs. Internet of Things

- IOT specific demands on IP Networks
  - Real Time / low latency
  - Demand Response Interaction
  - High E-2-E Security

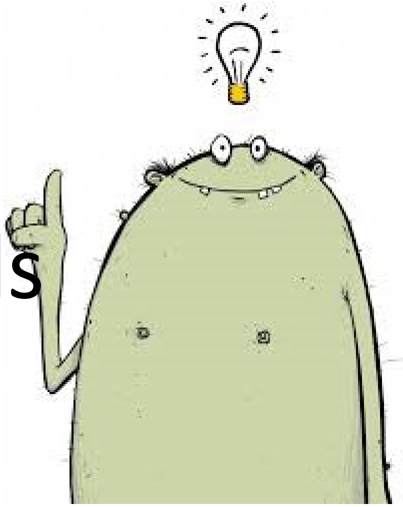


Apps
People
M2M
Things
io
RealTime.io
Infrastructure for the Real-Time Internet of Things.
SECURE – END-TO-END SECURITY
SCALABLE – OVER 100K DEVICES PER INSTANCE
STABLE – OVER 4 YEARS OF DEVELOPMENT, PATENTED
POWERED BY
ioBridge

The Programming Paradox

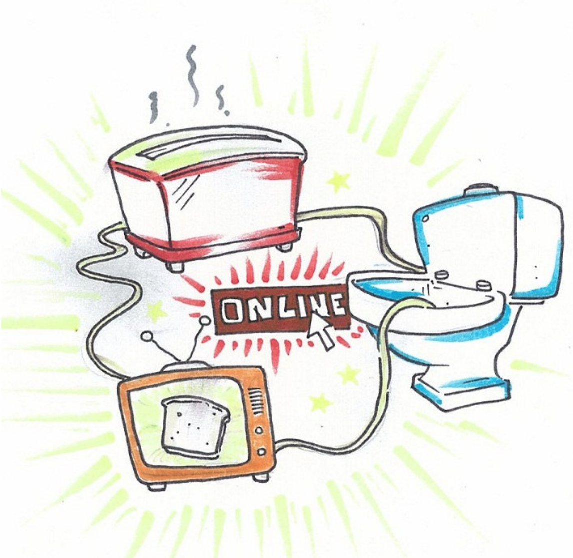
- How to program ?
  - A simple device
  - A complex (SOC)
  - A Network
- How to program for?
  - Low cost
  - High Security
  - Long Term Sustainability
  - Dynamic Product Life Cycles
  - Low Latency – Real Time

Conclusion….



Event Driven Systems:

1. Sustainable Program Models
2. High Security
3. High Relability
4. ??

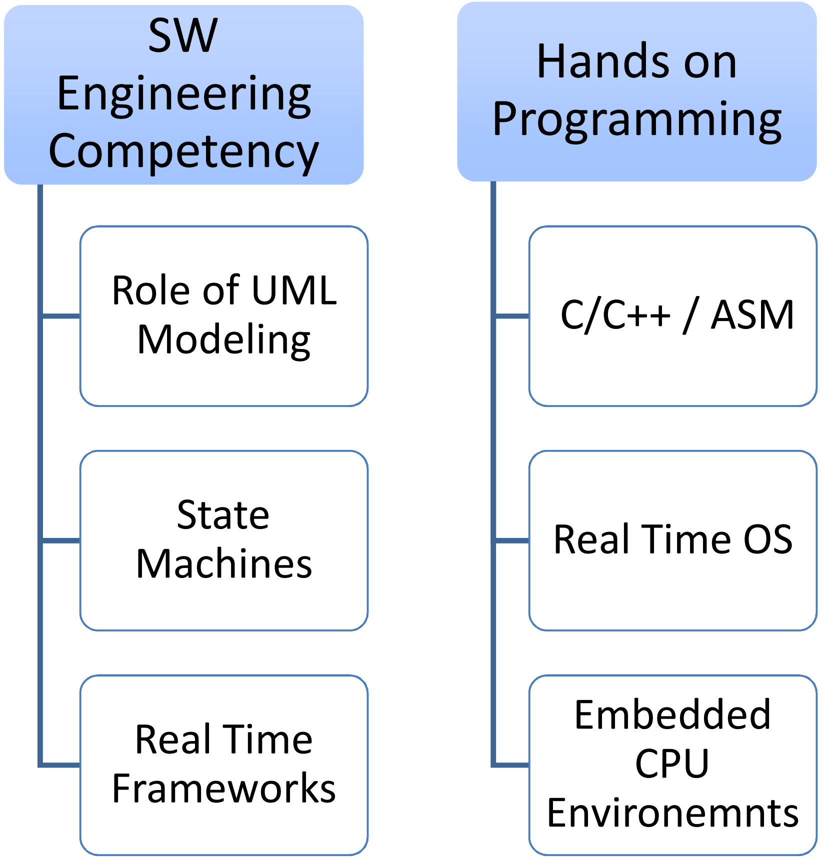


Embedded – Real Time - Reactive Systems

OVERVIEW OF COURSE DESIGN AND GOALS

HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN · FH
MÜNCHEN

Areas of Focus in this Seminar



SW Engineering Competency

Role of UML Modeling

State Machines

Real Time Frameworks

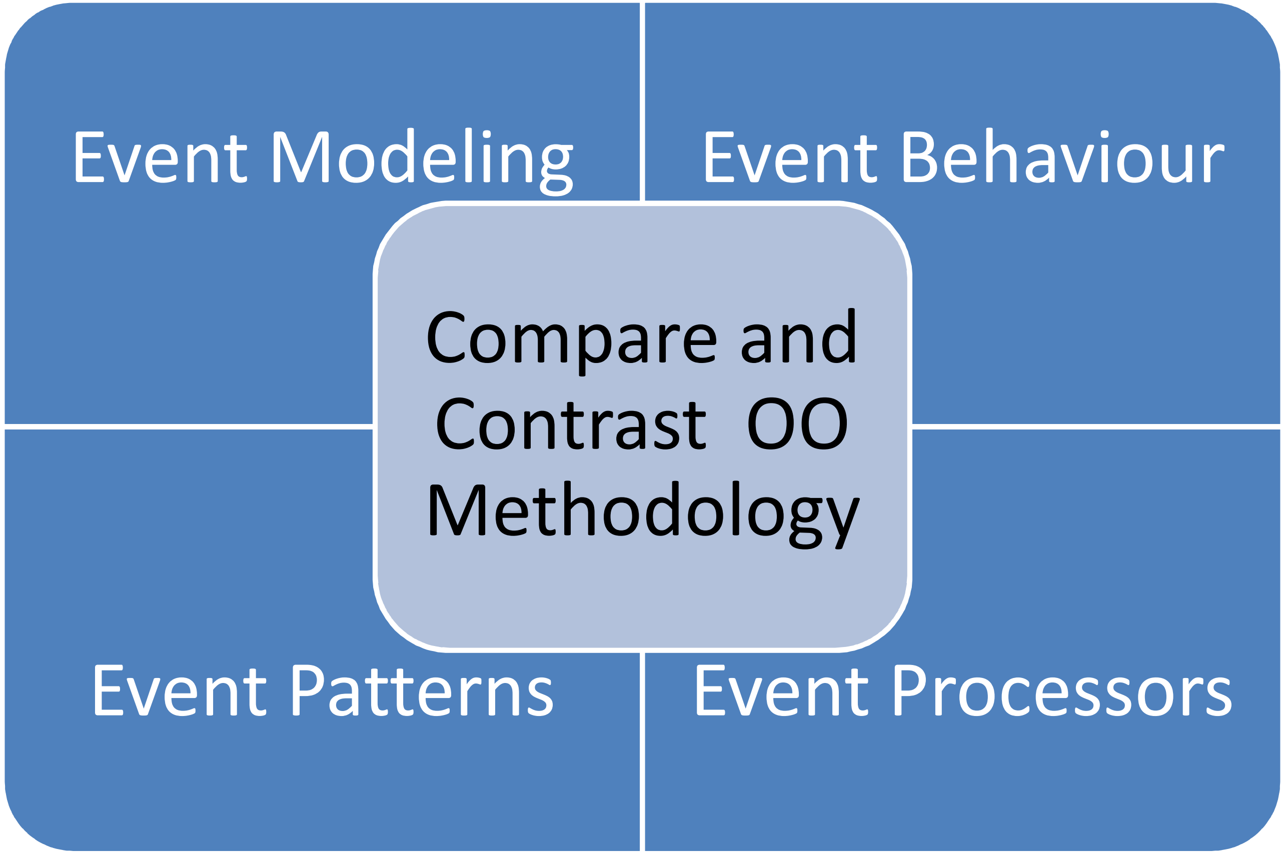
Hands on Programming

C/C++ / ASM

Real Time OS

Embedded CPU Environemnts

Software Engineering



Event Modeling

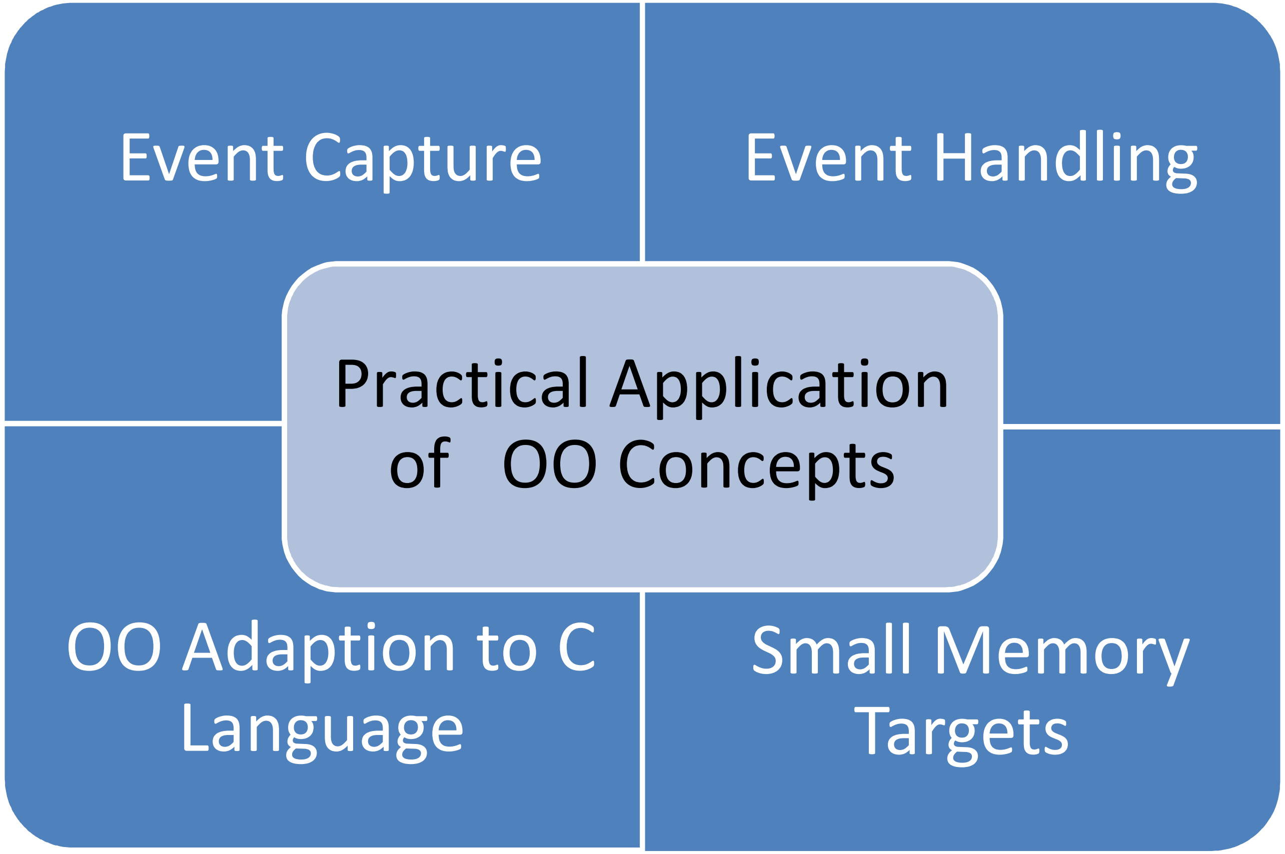
Event Behaviour

Compare and Contrast OO Methodology

Event Patterns

Event Processors

Event Driven Real TimePrograming


Event Capture
Event Handling
Practical Application of OO Concepts
OO Adaption to C Language
Small Memory Targets

Tools and Text for the Course

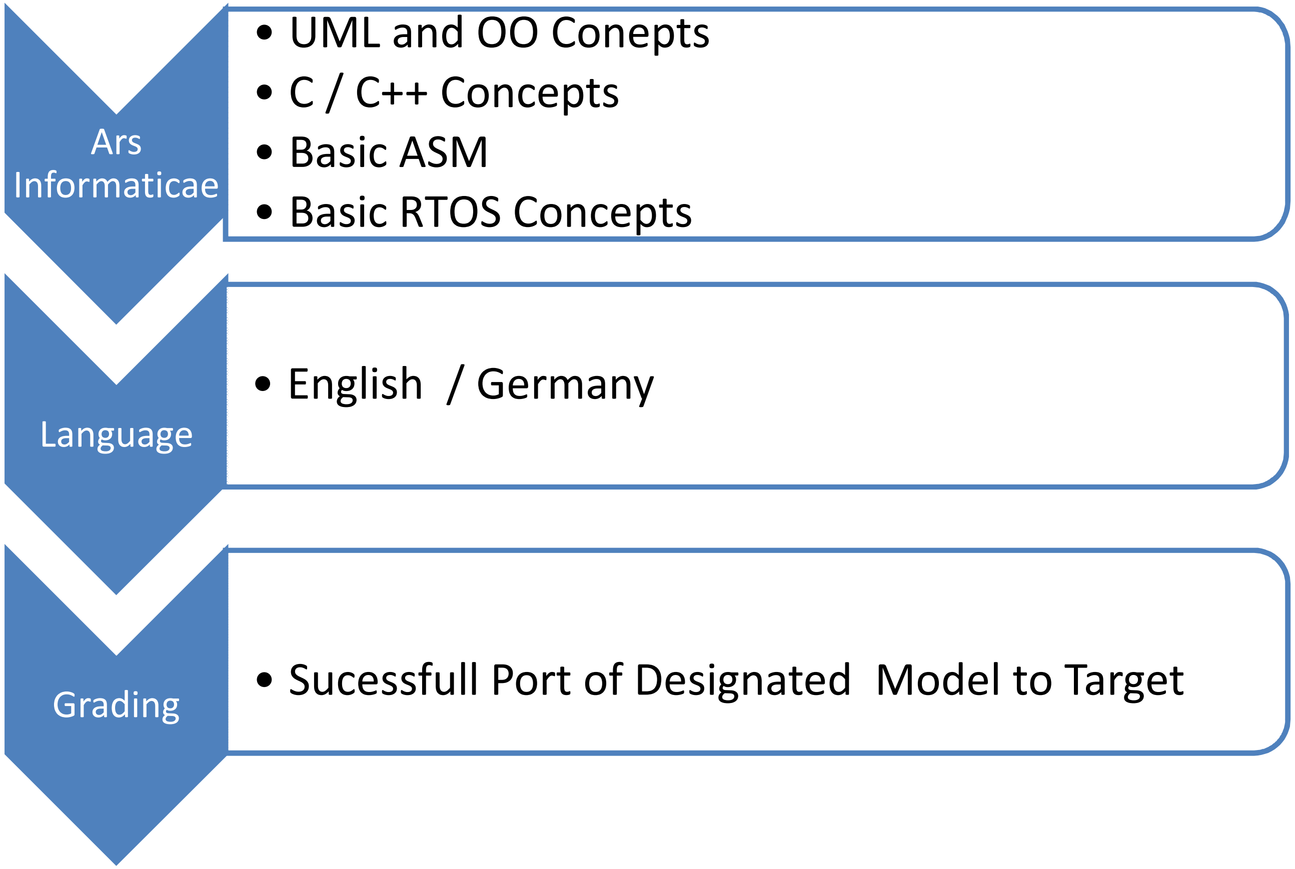
Text – 12 Copies in Library

- Samek, Miro Practical UML State charts in C/C++: Event Driven Programming for Embedded Systems , Butterworth Heinemann; 2nd Edition 2008.

Tools

- IDE µVision
- UML Modelling – Quantum Modeller
- QPC Framework
- HW: Keil   MCB 2300 ARM 7 TDMI

Course Requirements



Ars Informaticae

- UML and OO Conepts
- C / C++ Concepts
- Basic ASM
- Basic RTOS Concepts

Language

- English / Germany

Grading

- Sucessfull Port of Designated Model to Target

Embedded – Real Time - Reactive Systems

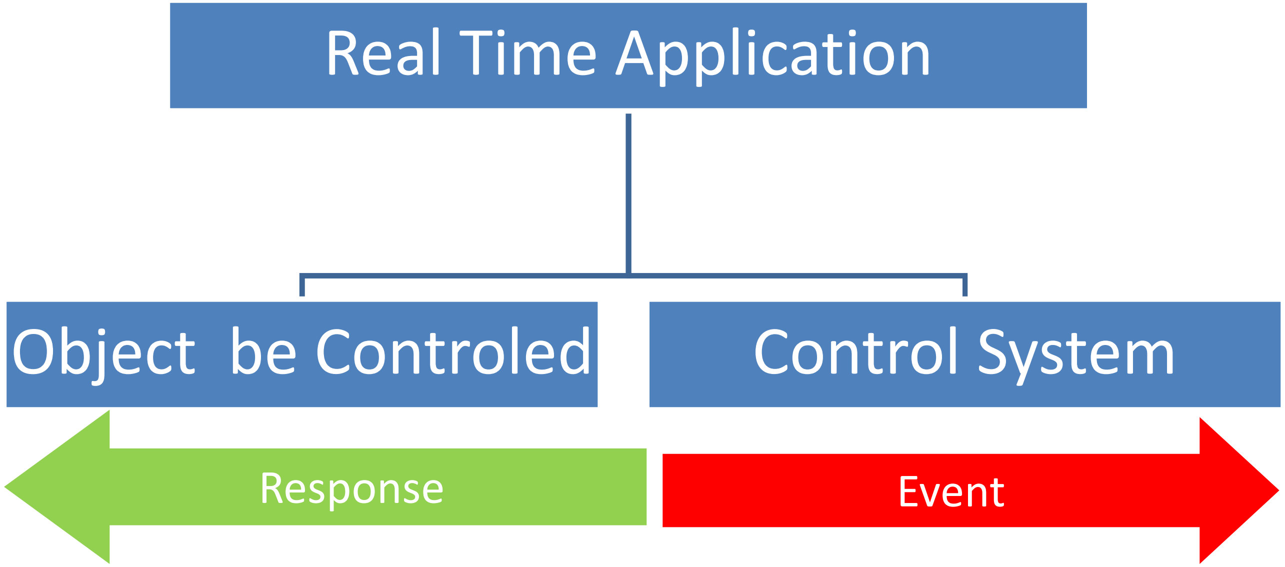
OVERVIEW OF REACTIVE SYSTEMS ARCHITECTURES

HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN · FH
MÜNCHEN

Embedded Real Time Event Driven Systems

Real Time Systems are used to Control Real World Applications

The Object is the Real Time Entity



Real Time Application

Object be Controled

Control System

Response

Event

Real Time System Types

- Regardlesss of the trigger event – two basic approaches
  - Event Driven – event type determinse the state change of the RT Entity
  - Time Triggered – periodic time slices determine the state change of the RT Entity

Event Driven Systems

- Real Time Systems are „event driven“ when Program control is a function of an event occuring in the system.
  - External Interrupts
  - Termination of a process
  - Receipt of a message
- Event Driven Systems describe event behaviour

Event Types

- Predictable Events
  - Function of physical activity
    - Pressure in vessel exceeds a certain limit
  - Determinstis, hence resource allocation and reservation is integral part of system design
- Chance Events
  - Event occurance is coincidental
  - Non Deterministic
    - Stochastic Principles required (Markov)

Time Triggered Systems

- Control signals are function of observing status of RT Entites during a time progression
- State Status infomation transmitted within the time slot of the observed RT Entity
- Granulatrity of oberservation is critical :
  - – Time slice too large – risk of missing state status
  - – Time slice too small – risk of unstable stat status

Event vs Time

- Predictability:
  - Event – dynamic response architecture critical
  - Time - precise planning of scheduling critical
- Resource Requirements
  - Event - CPU basically idle until event occurs
  - Time – CPU is always active
- Maintaince
  - Event – Depends on Code Model
  - Time - Depends of Data Flow between Nodes

Conclusion-Focus on Event

Attribute	Event	Time
Predictability	-	+
Resource Requirement	+	-
Flexibility	+	-