

**Artist2-RTC cluster workshop  
Beyond Autosar  
23-24 march 2006, Innsbruck,  
Minutes by Albert Benveniste**

<b>1. Participants.....</b>	<b>5</b>
<b>2. Stefan Kowalewski (Aachen) : on the relation between software development and control function development in automobile embedded systems.....</b>	<b>5</b>
2.1 Short CV and experience from industry .....	5
2.2 Relations between control & software engineers, referring to V-cycle .....	5
2.3 Research: ZAMOMO project.....	7
2.4 Conclusion .....	7
2.5 Discussion .....	7
<b>3. Karl-Erik Arzen (Lund): Time, events and components in automotive embedded control systems.....</b>	<b>7</b>
3.1 Trends in automotive systems and consequence for control .....	7
3.2 Controller timing.....	8
3.2.1 Pros/Cons of TT vs ET:.....	8
3.2.2 Latency versus jitter:.....	8
3.3 Analysis tools.....	9
3.3.1 Jitter margin .....	9
3.3.2 Jitterbug .....	9
3.3.3 true time .....	10
3.4 Controller components.....	10
3.5 Conclusions .....	10
3.6 Discussion .....	10
<b>4. Carlos Canudas-de-Wit (LAG, Grenoble): Control design for X-by-wire components: steering by wire .....</b>	<b>10</b>
4.1 Introduction: challenges .....	11
4.1.1 Challenges.....	11
4.1.2 Car product design evolution .....	11
4.2 Examples.....	11
4.2.1 Clutch synchronisation.....	11
4.2.2 Steer-by-wire.....	12
4.3 Conclusions .....	13
4.4 Discussion .....	13
<b>5. Panel session, moderated by Bengt Jonsson (Uppsala).....</b>	<b>13</b>
5.1 Panelists: speakers, plus Bengt.....	13
5.2 Discussion .....	13
<b>6. RTC cluster working meeting.....</b>	<b>14</b>
6.1 Kopetz-Caspi proposal for a next meeting .....	14
<b>7. Werner Damm – Introduction to the workshop .....</b>	<b>16</b>
7.1 Drivers for change .....	16
7.2 Anticipating changes in processes.....	16

7.3	Success criteria for this meeting .....	16
<b>8.</b>	<b><i>Christian Salzmann (BMW car IT): AUTOSAR, first experiences and the migration strategy of BMW group [slides confidential, revised version provided later]</i></b>	<b>16</b>
8.1	Facts .....	16
8.2	Model based development under AUTOSAR .....	17
8.2.1	AUTOSAR 1 <sup>st</sup> experiences, model based development under AUTOSAR: .....	17
8.2.2	AUTOSAR 1 <sup>ST</sup> experiences: Virtual Function Bus .....	17
8.3	History and lessons learned for BMW .....	18
8.4	BMW strategy for migration .....	18
8.5	Future steps beyond AUTOSAR .....	18
8.5.1	Timing and scheduling .....	18
8.5.2	Safety aspects at model level .....	18
8.5.3	Error handling at VFB level .....	18
8.5.4	Concluding remarks .....	18
8.6	Discussion .....	19
<b>9.</b>	<b><i>Stefan Sonck-Thiebaut (Carmeq GMBH) co-authors, F. Schöttler, Carmeq, and B. Kundel, VW AG: The AUTOSAR component model – canceled</i></b>	<b>19</b>
<b>10.</b>	<b><i>Kai Richter (Symtavision): the AUTOSAR timing model – status and challenges</i></b>	<b>19</b>
10.1	Disclaimer .....	19
10.2	AUTOSAR in general & target use cases .....	19
10.3	To-down: SW architectures vs execution platforms .....	20
10.4	Acloser look at technical details; Talk and Discussion .....	20
10.5	Bottom-up Integration & timing analysis practice today .....	21
10.6	Implications wrt AUTOSAR goals .....	21
10.7	Conclusions .....	21
10.8	Discussion .....	21
<b>11.</b>	<b><i>Panel: Beyond AUTOSAR. A. Benveniste (INRIA, recording), W. Damm (Offis, moderator)</i></b>	<b>22</b>
11.1	What are the key industrial challenges? .....	22
11.1.1	Alberto Ferrari: robustness to change .....	22
11.1.2	Rolf Ernst: tight intertwining of time in platform and functionalities as components .....	22
11.1.3	Rolf Ernst: problem with the many scenarios .....	23
11.1.4	Kai Richter: do we need complete models at all? .....	23
11.1.5	Martin Törngren: multi-paradigm integration, timing and safety/reliability .....	23
11.1.6	Julio Medina: components .....	23
11.1.7	Christian Salzmann: special conditions of development process .....	23
11.1.8	Werner Damm: safety is important and should be handled like timing .....	23
11.1.9	Rolf Johansson: is safety a component feature? .....	23
11.1.10	Heiko Dörr: enabling a smooth transition of all future benefits into a smooth existing design environment .....	23
11.1.11	Heiko Dörr: we use a lot of tools with lots of uncertainties about the modeling assumptions, find appropriate restrictions that will make the analysis feasible .....	24

11.1.12	<i>Heiko Dörr: at present we have with AUTOSAR systems which are have simple environment models; upcoming systems will be more complex (hybrid behicles...), with stronger interactions between mechanical components</i>	24
11.1.13	<i>Werner Damm: is AUTOSAR going to open to the academic community?</i>	24
11.2	Position statements regarding research	24
11.2.1	<i>Julio Medina</i>	24
11.2.2	<i>François Terrier</i>	24
11.2.3	<i>Alberto Ferrari</i>	24
11.2.4	<i>Stefan Kowalewski</i>	24
11.2.5	<i>Rolf Ernst</i>	25
11.2.6	<i>Susanne Graf</i>	25
11.2.7	<i>Rolf Johansson (Mentor Graphics): on timing model</i>	26
11.2.8	<i>Willem-Paul de Roeve: very quickly compositionality reaches a dead end</i>	26

ARTIST2-RTC Afternoon on  
**Control and Embedded Systems in Automobile**  
March 23, 2006

## 1. Participants

See attached sheet.

## 2. Stefan Kowalewski (Aachen) : on the relation between software development and control function development in automobile embedded systems

Software and control functions are not smoothly integrated at the moment, this is a technical problem. There are also soft issues related to culture and background. Some believe software and control are the same. Aim of the talk: share experiences and suggesting ways of improving integration. The opinions expressed here are personal.

### 2.1 SHORT CV AND EXPERIENCE FROM INDUSTRY

I have background in control, I worked during 2000-2003 at Bosch; since then I am with the computer science department at Aachen.

When I entered industry, my background was in control, discrete event systems, and formal verification. Main work topics I worked on then were different: SW engineering, SW architecture design and analysis, SW reuse and variability management. There was clearly a mismatch of background. The reason is that soft topics were more perceived by the management as being critical at that time. Most problems had been caused by SW mastering problems. Automotive supplier industry felt it was well experienced at developing new functionalities; problems arrived when additional customers call for variants.

“Hard” methods were not in the focus at that time (control, verification). Control design is considered mastered, formal verification was not of interest.

Many software architecture analysis workshops were organised. Their basis was: architecture trade-off analysis method (ATAM), delivered by the SW engineering institute, Pittsburgh. Participants were: marketing, architects, SW developers, testers, and, whenever possible, management;

Experiences gained from this:

- Requirements management, with late changes in architecture-relevant requirements;
- Communication between marketing & development was not best;
- Organizational structures did not fit any more because of domain-crossing functionalities;
- Reuse and variability of products did not fit market requirements; there are up to 1500 variants of gasoline engine control systems sold per year;
- There is no cost model for software, no lifecycle cost consideration, product prices are determined by hardware, HW fixed before SW development begins;
- There is a permanent misunderstanding between control & software engineers.

### 2.2 RELATIONS BETWEEN CONTROL & SOFTWARE ENGINEERS, REFERRING TO V-CYCLE

- Requirements analysis, architecture design, module/algorithms specification, integration and acceptance tests: all this is under the umbrella of control engineers;

- Implementation and unit test are under the umbrella of SW engineers, with manual handing over of printouts of ASCET or SIMULINK designs.

*Question: who is responsible for fixed point arithmetic issues?* When dynamics was important, then control engineers were responsible; otherwise SW engineers were.

How do control & SW engineers see each other?

- Control engineers think that system structure (the trivial part) and algorithms (the difficult part) follow from control requirements  $\Rightarrow$  they think that the system should be designed by control engineers; remaining tasks would be for SW engineers.
- SW engineers in research departments think that control engineers make it wrong in the architecture phase  $\Rightarrow$  they think that the system should be designed by SW engineers; on the other hand, algorithm design is trivial (there are tools for that!); they think that computer aided control engineering tools are used to escape from architecture considerations via code generation.

Need:

- Better understanding between SW & control engineers;
- At least mutual sensitivity to challenges on both sides.

Clarifying goals and responsibilities:

- For control engineers: they should understand that there is more to the quality of control SW than just correct functionality and control loop performance;
- For SW engineers, they should realize that control function design is not a SW design problem, closed loop dynamics rather is a challenge in its own.
- What seems helpful: strict separation between functional & non-functional requirements or properties  $\Rightarrow$  should this be taught?

Further clarification, two requirements analysis paths:



- Technically oriented, dealing with functionalities (the meaning is clear when considering SW; for control, it consists of the set of constraints for control).
- Business oriented: analysis of expected qualities perceived by the customer, driving qualities, optimization criteria for control.

Both analysis paths are inputs to architecture design, and design is seen as optimization. This raises a question to AUTOSAR: **are all important questions considered in AUTOSAR?**

Example for preparing SW engineers: course on *dynamic systems for CS students*.

**Example for preparing software engineers:  
Course „Dynamic Systems for CS students“**

- Signals
  - mappings from time to value, no matter whether domain and co-domain are discrete, continuous or hybrid
- Systems
  - mappings from input signal space to output signal space
  - State as a general concept for representing dynamics  
 $\Rightarrow$  automata, cont. state space, hybrid systems
  - Linear systems
- Analysis
  - General properties: Causality, controllability, observability, reachability, stability
  - Continuous systems: Frequency domain analysis, time domain analysis
  - Discrete systems: Temporal logic, model checking
  - Hybrid systems: reachability
  - Simulation (integration of ODEs, DES simulation)
- Design
  - Continuous systems: Linear controller design
  - Discrete systems: Supervisory control synthesis, game theoretic methods


Slide 1


Remark from Albert Benveniste: teaching the important philosophy about modelling, where models are approximations, is important for computer scientists.

## 2.3 RESEARCH: ZAMOMO PROJECT

Integration of model-based SW and model-based control systems design; partners are RWTH Aachen (control & SW labs) VEMAC GMBH, AVL GMBH, Fraunhofer institute. The vision is to introduce early consideration of non-functional requirements in control system design, with early introduction of plant models.

## 2.4 CONCLUSION

Challenge: bridging the gap between SW & control development. There is a need to prepare both disciplines by appropriate teaching.

## 2.5 DISCUSSION

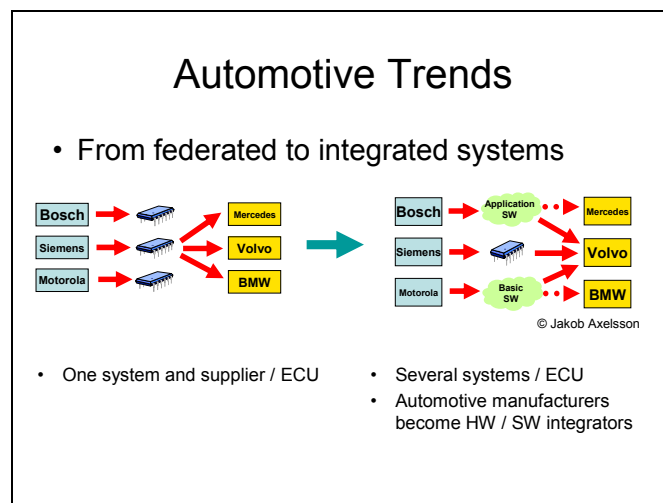
Q: where does it appear that control engineers think synchronously whereas SW engineers think asynchronously? You are right: this is a problem, by experience. Still, there are good reasons for asynchronous paradigms since it facilitates replacement; of course other issues are much more important.

## 3. Karl-Erik Arzen (Lund): Time, events and components in automotive embedded control systems

Self-presentation: somewhere in between control and embedded systems engineer. Disclaimer: no direct knowledge of AUTOSAR

### 3.1 TRENDS IN AUTOMOTIVE SYSTEMS AND CONSEQUENCE FOR CONTROL

The role of control is increasing in modern cars. Quality of performance and control loops should be top priority. There is a move from federated to integrated systems on the same ECU:



This results in increased functionality and complexity → calls for standardised architectures and support for reuse: this is what AUTOSAR addresses (virtual bus for communication) and component technology.

What are the consequences for control?

- A sensor may be used by several systems; should it be part of vehicle platform or part of one system and then shared with other?

- A same actuator may be used by several systems (not so many actuators are available!); actuator components will be special, as some actuator controllers will play a role for different functions sharing a same actuator.
- Hence, cascaded control structures will dominate, hierarchically layered; the different control components will be part of different systems sitting on the same or different ECU's.

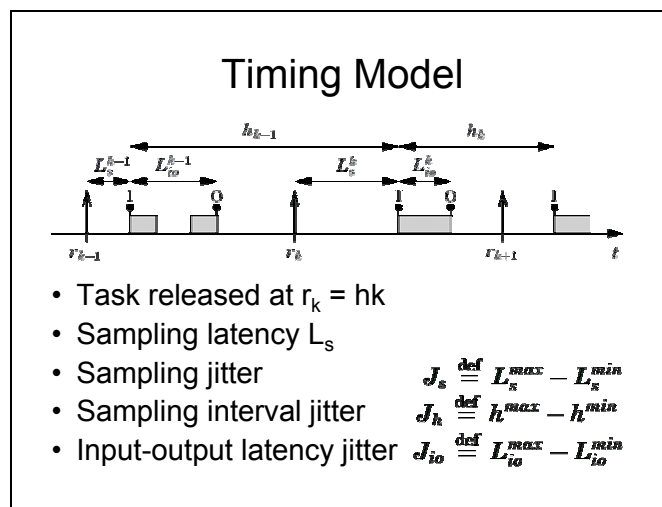
## 3.2 CONTROLLER TIMING

Classical control assumes deterministic sampling. In most cases sampling is periodic (not for engine control). Too long sampling interval or too much jitter cause poor closed loop performance. Classical control assumes constant or negligible latencies. Too much jitter causes problems.

Elements of networked control timing:

- tasking systems may cause temporal nondeterminism;
- limited communication resources may cause nondeterminism too.

Timing model:



Of course, strict policies could be imposed on sampling times.

### 3.2.1 PROS/CONS OF TT VS ET:

- A TT approach with global clock maximizes temporal determinism and protection.
- However, maximizing temporal determinism may degrade control performance since it amounts to sample at maximum bounds for latency (experiments show that this is worse than random latency for closed loop control performance); it is also inflexible.
- Thus choice is not simple.

### 3.2.2 LATENCY VERSUS JITTER:

- Both degrade control performance.
- Jitter can be lowered by buffering (TT approach).
- It is easier to compensate for constant rather random delays.
- Which is worse: latency or jitter?

Reducing latency: try to minimize the interval between sampling and output by splitting the code into 2 parts:

1. compute output and then
2. update state.

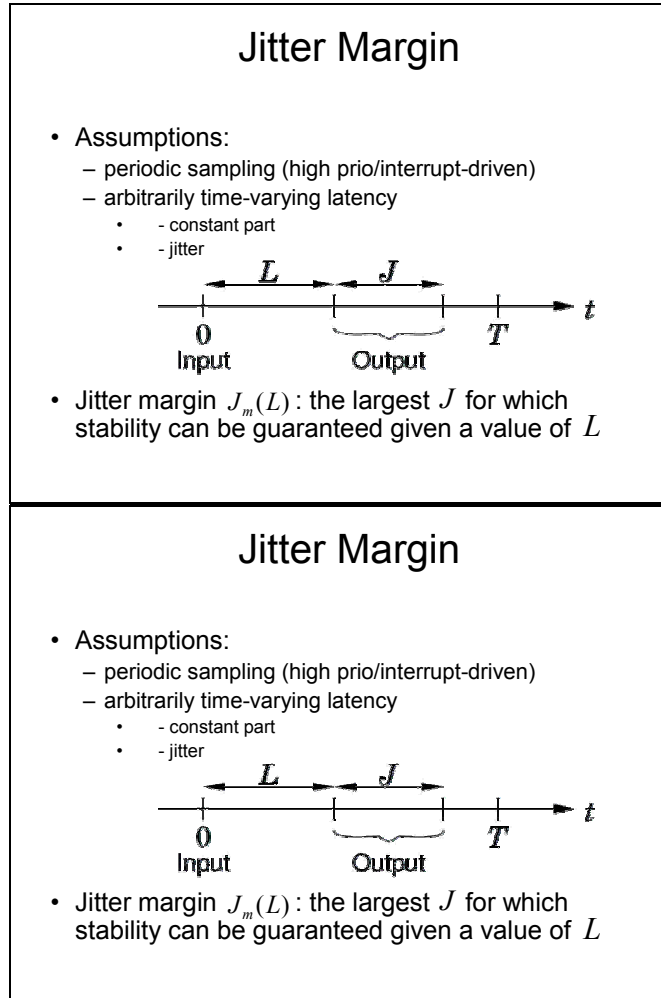


Pre-calculate much as possible.

### 3.3 ANALYSIS TOOLS

#### 3.3.1 JITTER MARGIN

the technique presented here is an extension of the classical phase margin / delay margin analytic approaches; it is also an analytic approach.



This technique uses the small gain theorem; it provides only sufficient conditions, but it is not very conservative. It is valid only for linear systems, with a single latency to be considered. It ends up with a graphical frequency domain test.

#### 3.3.2 JITTERBUG

This is a Matlab based toolbox for analysis of control performance versus timing characteristics, expressed as a quadratic performance criterion function. It uses jump Markov processes theory. The model uses hybrid continuous/discrete time blocks driven by white noise. The perturbation of timing models of the nodes is simulated by stochastic timing models. This is a statistical tool. TT and ET types of approach can be compared with the help of this tool.

Experiments with this tool showed that random latency is better than worst case latency, for control loop performance: this speaks against the TT approach which results in deterministic but larger latency. But the desire of synchronised sampling for other reasons speaks in opposite side.

### 3.3.3 TRUE TIME

This tool performs simulation of network control loops under shared computing & communication resources. Several communications infrastructures are provided in libraries.

One can do co-simulation of functions and some aspects of architectures. It is mainly used by universities, some industrials have tried it. (Bosch has extended the blocks with Flexray and TTCan). >1100 downloads.

*(These are features that are also provided by the RT-Builder tool by TNI-Software.)*

### 3.4 CONTROLLER COMPONENTS

Components models for embedded systems are often based on the “pipe and filter” model. Alike Simulink blocks. Components usually describe logical signal flow. However, this is not enough for controller components.

Problem 1: there is a need to minimize the latency {sensor → actuator} and this requires *re-architecturing* the algorithm differently.

Problem 2: bi-directional signal flow can arise, due to actuator saturation and multiple controller modes.

This problem was already observed by ABB in the late 80's and lead to ABB adding a lot of features to their component model, for code generation. The sorting of code is done automatically, based on this model. This gives the plug-and-play functionality (still recompilation is needed).

### 3.5 CONCLUSIONS

There are 2 communities:

- AUTOSAR, SW engineers, UML2, MDA...
- Simulink, control, modelling for simulation.

How to get a convergence of views on the concept of component?

Time or events to trigger actions: this is not an easy question but can be a trade-off.

There are analysis tools to assess this.

Reuse and performance puts special requirements on the components concept.

### 3.6 DISCUSSION

*Q (Volvo): do you want to standardize the pattern related to component model? Yes.*

*Q: in the timed picture, was the mapping of functions on architectures amenable of capturing stability issues? No, this is too fine grained; the tool is only capable of coarse grain architectural features. Detailed architecture analysis is beyond the capability of TrueTime.*

*Q: did the AUTOSAR people miss the impact of architectural choices on control? Yes, agree.*

*Q: in AUTOSAR they are mainly fixing syntactic aspects of architectures, no so much behavioural ones. What was discussed here seems orthogonal to the issues considered in AUTOSAR. Yes.*

*Q: pros and cons of TT/EE. Should call for reactions from promoters of TT? None.*

*Bottom line remark from Albert: fine coding of data dependencies should be part of the component model. This was observed already in the area of synchronous programming, for different reasons related to compositionality and separate compilation. Therefore, this does not come to a surprise.*

## 4. Carlos Canudas-de-Wit (LAG, Grenoble): Control design for X-by-wire components: steering by wire

I originate from the control community; I have worked with Renault for 10-12 years.

My work addresses a class of systems with a driver in the loop: important fact.

New control paradigms: example of Segway 2 wheels scooter, which cannot move without control. There is a need to define what is

- safety
- comfort
- pleasant ride

for such systems.

## 4.1 INTRODUCTION: CHALLENGES

### 4.1.1 CHALLENGES

Drivers have different skills. An important issue is how to partition between driver & automated system. There is a trade off between increase of work load and lack of awareness or over confidence. The problem with cruise control system is a well known case. The objective can be summarized as “**fun to drive**”.

Technical challenges that result:

- Lack of well defined metrics to measure comfort and safety. No control concept or metrics (stability, robustness...) meets this. One would like to have very few parameters to tune, for cost issues and ease of testing/design.
- Transfer of standard control notions is not always straightforward.
- How to translate subjective vehicle specifications into more precise control specifications?

There are many examples of such issues:

- automatic clutch
- steer-by-wire,
- chassis control
- adaptive cruise control

### 4.1.2 CAR PRODUCT DESIGN EVOLUTION

Car makers are now moving to customer-perceived performance: drivability, quality,... Distinctive features tend to become basic features.

*Example of high torque diesel engines.* This calls for higher torque capacity for clutches: higher pre-constraint forces, larger friction plates, multi-clutch plates... All this has limits (increase of pedal effort, increase of moment inertia). The bottom line is that computer assistance cannot be avoided and people look for clutch-by-wire systems. Clutch-by-wire improves comfort and increases life of motor. (Of course, removing the clutch is another solution.)

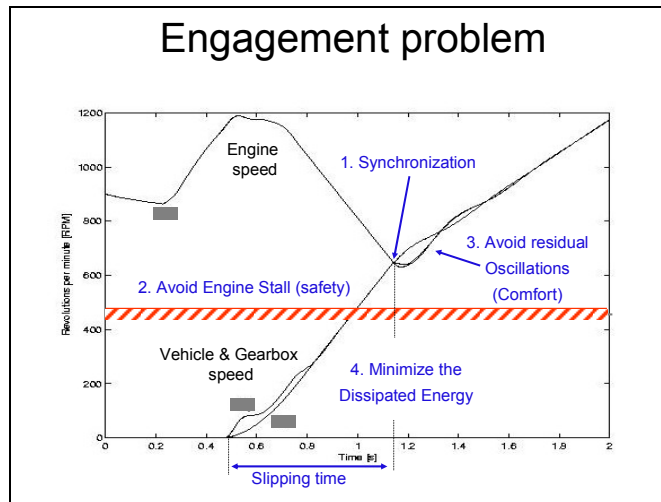
Different control setups or architectures can be considered, with more or less inputs considered.

## 4.2 EXAMPLES

### 4.2.1 CLUTCH SYNCHRONISATION

A key point is that models need to be simplified for control analysis and synthesis; take symmetries into account, simplify the dynamics of tires...

The engagement problem is shown here:

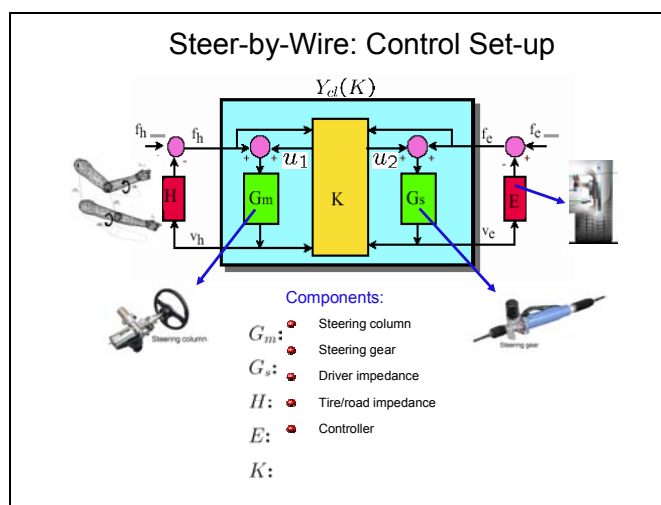


The control objective is formulated in terms of a reference trajectory that the system should best follow. Having this reference trajectory, tracking it is implemented by solving a two-boundary linear-quadratic optimal control problem in finite horizon, where energy aspects and dissipativeness are handled. This allows taking comfort into account. It is important that an analytic solution could be found to have it implemented on-line. This gives a family of possible solutions with only 2 parameters that remain for tuning to satisfy other constraints.

#### 4.2.2 STEER-BY-WIRE

Want to have: variable force amplification, variable gear ratio, active safety systems. Variable amplification systems already exist, either via hydraulics, or by inserting an electrical motor. This motivates considering steer-by-wire system, with no column at all. Many sensor measurements are used for this task.

STW technology:



The control setup consists in putting a *virtual column* between the gear and the steer. The virtual column must handle both the control that the driver wants to apply and the feedback force that the road returns to the driver. *Passivity* is an important mathematical concept that captures the fact that the driver feels driving the car and not the converse.

Try to reproduce hydraulic assistance by software. It is important that the design ends up with having a few parameters whose tuning relates to the characteristics of car ride.

The technique consists in trying to mimic a desired behaviour by control. This is called **model following control**, where the aim of the control is that the closed-loop system should behave in a pre-specified way in terms of dynamics. This can eventually be stated

as a known mathematical optimization problem on certain domains of functions, preferably convex (so that solving them is feasible).

#### 4.3 CONCLUSIONS

More applications are coming in which control specifications are subjective.

Complexity limits are reached in terms of control. It is unclear that engineers really master what's happening inside the loops. This complexity in design results from the product design history.

What is a component for control? Model following control might be a good approach: the desired behaviour is taken as the *interface model* and every other component that fits the same desired behaviour can be put instead.

Also, wireless communications start being considered by some car makers (in the US). Feasibility is addressed at this moment.

#### 4.4 DISCUSSION

*Q: what you mean by having several control loops on top of each other, where and why does it happen?* In engine control, objectives are new (pollution, efficiency...). This causes adding more loops on top of each other.

*Q: what assumptions are made regarding tires and other when analysing clutch; are they very sensitive and thus can change? How accurate models should be for them being useful?* In fact coarse but nicely designed models will do best for this purpose; they have wide range of validity; this is based on a good understanding of physical phenomena and what are the dominant factors.

*Q: what about variable delays?* There are control methods that allow compensating for that.

*Q: How can you reflect in your approach a change in style of driver (Italian vs British)?* It is one aspect of control design that you want to structure your control in such a way that parameters are interpretable in these terms.

### 5. Panel session, moderated by Bengt Jonsson (Uppsala)

#### 5.1 PANELISTS: SPEAKERS, PLUS BENGT

#### 5.2 DISCUSSION

*Werner Damm:* when you said you need to teach the computer scientist control issues, the question I have is how to address the gap between control design and implementation. To what extent did Carlos follow the actual implementation of the control law into actual computers, with the detailed technical effects it can have on the implementation?

*Carlos:* there is an understanding that control cannot continue the way it did. There is a move in the community toward networked control. The control community is conscious that the techniques need to be adapted to introduce 1/ communications aspect, and 2/ computations aspects. What happens if packets are lost from time to time? What is the relation between control and the amount of information to be communicated (for resource reasons)? The problem of asynchronous timing and measures has been a problem in Carlos' experiments. Smoothly degrading control must be considered in the spirit of QoS. The people from adaptive real-time consider QoS for their scheduling strategies; this is similar to what can be done in control. This is a new set of problems considered. The US community moves faster toward this direction than other parts of the world.

*Werner Damm:* you say that rather than separating control function design from architecture aspects as they try to perform in AUTOSAR, you should do the converse.

*Carlos:* yes in some sense. The problem cannot be solved by just plugging pieces together.

*Werner Damm:* Stefan, do you have suggestions of how to address this within AUTOSAR?

*Stefan K:* the aim of AUTOSAR is a bit different. They have a component approach and want to integrate pieces together. Two possibilities: either you give complete freedom and then you need to analyze things globally; or you have some kind of abstraction that simplifies the problem of interfacing.

*Rolf Ernst:* I would imagine that, having characteristics of the hardware infrastructure you control guys would be able to design a controller that works with these constraints.

*Karl-Erik:* resource awareness is now recognized as a main concern for control engineering.

*Stefan K:* you need such a concept like the *rich component model*, where several aspects are considered together.

*Albert B:* there are in my mind two fundamental issues in control:

1. ensure that robust control design techniques effectively address the artefacts created by the distributed architectures in use for embedded systems (e.g., ET, TT, Airbus schemes...).
2. there is a need for control people to figure out what control components should be and how can one make fundamental control objectives better compositional; e.g., by replacing stability (not compositional) by passivity.

*Carlos CDW:* the notion of robust systems with respect to time-varying phenomena is not well developed. The problem of allocation of resources for control needs to be addressed.

*Stefan K:* compositionality and control design is something which goes difficult together. Because, by essence, control analyses are global.

*Ramesh:* control system design is getting more influenced that control system design is influenced by computer aspects; I would like to see also the other way around, namely that computer scientists can capture issues from control in their activities.

*Karl-Erik A:* yes we have to meet.

?: I am not sure that compositionality in control is necessary. It is not a key issue.

*Alain Girault:* this problem also arises in computer science. For example deadlock is not compositional. But computer scientists know how to address this and still avoid deadlock for the global system.

*Rolf Ernst:* not clear that this issue of compositionality comes with AUTOSAR. Not sure how it can be done.

*Carlos CDW:* decentralized control is an area that tries to deal with this aspect of controlling interconnected systems. Also recent studies such as formation control in which control is forced to be local and still stability can be guaranteed. Stability is not compositional, but passivity is better. Different styles of control are needed, with different criteria (stability is not enough and not appropriate for Composability).

*Werner Damm:* I like the picture of fish formation control (with local control). How does this achieve stability?

*Carlos CDW:* what is achieved is not strictly speaking stability in this case, but something of the same kind. What is needed for each local subsystem to do in order that federated systems behave nicely, globally.

*Stefan K:* the OEMs have the vision that they can buy a device from supplier and just plug it in the system without any particular care from the control viewpoint.

## 6. RTC cluster working meeting

### 6.1 KOPETZ-CASPI PROPOSAL FOR A NEXT MEETING

Willem-Paul de Rover supports the topic "mobile embedded systems" (it is the topic of his current EU project). Werner Damm and several others support having two workshops: one on MoCs and one on mobile embedded systems.

Werner Damm and Reinhard Wilhelm propose another one day workshop on the predictability of hardware in automotive/avionics and semiconductor industry. The workshop should invite

key industrials (BMW, Daimler, Infineon, Bosch, IBM, ...) and leading academics. The proposed date is June 14<sup>th</sup> in Munich.

Votes from the assembly show that there is an interest for the three, with a non empty intersection. Each workshop should be one or two days and give appropriate time for science. But four days in a row is too much, meaning that the workshops must be organised separately.

The Artist RTC cluster gives support to all three workshops.

Alberto Ferrari and Alberto Sangiovanni-Vincentelli also want to organize a workshop in fall 2006 on building automation and security of building automation, and 2007 will see a workshop similar to today's workshop but with aeronautics industry.

## AUTOSAR the industrial perspective, 24 march

### 7. Werner Damm – Introduction to the workshop

#### 7.1 DRIVERS FOR CHANGE

- flexibility: decouple growth rate from number of functions and growth rate from number of electronic components;
- adaptability: decouple life time of functions from hardware;
- cost: decouple growth of number of functions from increase in cost;
- and quality is still to be maintained.

#### 7.2 ANTICIPATING CHANGES IN PROCESSES

There is a strong push toward virtual subsystem models (function level):

- target independent
- topic in AUTOSAR.

Strong push towards component based development:

- topic in AUTOSAR.
- requires component characterization dealing with nonfunctional aspects

There is a need to boost quality:

- to support IEC 61508 customized to Automotive domain by paying attention to safety cases
- reduce # of recalls
- topic in AUTOSAR.

Deployment analysis capabilities will be a key competence, allowing for integration of subsystems from suppliers and value capture.

#### 7.3 SUCCESS CRITERIA FOR THIS MEETING

- Identification of key industrial challenges as seen by the AUTOSAR consortium;
- Identification of possible lines of attack based on research competence represented in workshop;
- Documented in a published report: ***Beyond AUTOSAR, key challenges in component based development of automobile systems.***

### 8. Christian Salzmann (BMW car IT): AUTOSAR, first experiences and the migration strategy of BMW group [slides confidential, revised version provided later]

Subsidiary of BMW, member of BMW group. CS is responsible for the assembly of SW system. He was involved in the specification of the AUTOSAR run time.

#### 8.1 FACTS

In premium cars 200-300 MBytes of codes are deployed on over 60 ECUs (This amount is not going to increase for the next 8 years), which are connected by 6 types of buses. It is expected that having that 1GByte code in the car will be reached within 5 years. This will yield a very



complex computing system. 5 years ago, the most complex communication concept was a global variable. This is why there is a strong pressure toward model based development techniques and system architectures allowing new ways to develop SW, adapted to the specific needs from automotive domain.

This is the aim of the AUTOSAR standard.

## 8.2 MODEL BASED DEVELOPMENT UNDER AUTOSAR

This is a development partnership between car makers and major suppliers. It aims to develop common standard architecture for common SW in ECU's. 1<sup>st</sup> tier suppliers integrate things within ECU's, whereas OEM integrates ECU's into the system.

Today the infrastructure simply consists of a couple of libraries and function applications. Each OEM bundles a couples of elements of his libraries to build his standard core. Each OEM does this differently: the aim is to standardize this.

[slide 1 in appendix]

The more interesting part is that, within this architecture, there is a component model: the *application component*. Also important is the *AUTOSAR run time environment (ARE)*. This aims at decoupling application layer from infrastructure, allowing for possibly orthogonal architectures. This plays a role like CORBA. We'll focus on the upper layer of the AUTOSAR architecture for the rest of the talk.

### 8.2.1 **AUTOSAR 1<sup>ST</sup> EXPERIENCES, MODEL BASED DEVELOPMENT UNDER AUTOSAR:**

[slide 2 in appendix]

At the beginning, only the component types are specified, without the applications. No specification exists of which component is deployed on which ECU.

**Each component must indicate a large number of features regarding communication interfaces (port types, type of communication blocking / nonblocking, queued, how the other extremity handles the communications, etc...). There is thus fine typing regarding communications. If two components want to communicate, they express this in terms of the virtual bus, without knowing the actual localization of each other, nor the type of actual bus technology being used. This is not completely plug-and-play, for reasons of scheduling.**

System deployment and ECU description are provided in XML documents. The run time environment is generated that performs the mapping on the actual deployment that was specified. This is shown on the bottom part of the diagram. The run time environment should be very efficient, with little overhead possible. This run time generation is performed off-line while configuring the car system.

This calls for a different organization of embedded systems market. ECU integrators are needed, in addition to the actual suppliers.

Only syntactic aspects of interfaces are specified at this point, not behaviors, not functional aspects.

### 8.2.2 **AUTOSAR 1<sup>ST</sup> EXPERIENCES: VIRTUAL FUNCTION BUS**

[slide 3 in appendix]

This virtual bus offers 48 communication variants that are formally specified by the AUTOSAR standard. For example, if I have a local communication, I could implement this by call back (communicate if the state change); it can be implemented cyclically if you have a CAN bus.

Any implementation of the VFB must implement all these possibilities.

How did the AUTOSAR consortium come up with this list of communication modes? This was the task of the group working on that topic. The problem was not to fulfill all the requirements. It was rather to find the right abstractions for communications.

## 8.3 HISTORY AND LESSONS LEARNED FOR BMW

There was a huge gap being done within the culture of the company. This started in January 2004. There were three phases:

1. Proof of concept
2. Piloting, confirming applicability by piloting existing functions with existing ECU's.
3. Consolidation phase. Integration of resulting SW and tools into the BMW Standard Core 6.

Proof of concept – June 2004:

[slide 4 in appendix]

This is a simple example consisting of the mirror management system.

Architecture of the BMW SC/RTE:

[slides 5, 6 in appendix]

This way components can be subsequently reused on other cars.

At this point, not all features of the VFB have been implemented by BMW; the implementation was demand driven.

## 8.4 BMW STRATEGY FOR MIGRATION

There was a need to find a migration strategy, from the existing components base, to an AUTOSAR compliant base.

[slides 7 – 12 in appendix]

There is a need to generate a Run Time Environment (RTE) against an existing COMMunication matrix. This migration is supported by a tool called ORPHEUS, helping to define the mapping between VFB signals and effective COMM signals. ORPHEUS also includes a code generator (through ASCET). It is coupled to ASCET and performs scheduling, partition of SW, and targets CAN and Flexray bus. But the vision is to go the other way around, by generating the COMM matrix from RTE; the deployment tool that will go with AUTOSAR will generate the COMM matrix at deployment.

Voices back 2002: it will never work; overhead of RTE will be huge, it will double RAM/ROM needs... In 2005, the powertrain pilot has been developed in cooperation between BMW, BMW Car IT, and Siemens VDO. The overhead is very low and performance is satisfactory; because everything on the RTE is generated off-line.

## 8.5 FUTURE STEPS BEYOND AUTOSAR

AUTOSAR is seen as an enabler for doing what was already done. It could do more:

### 8.5.1 **TIMING AND SCHEDULING**

It is hoped that this can be made at modeling level.

### 8.5.2 **SAFETY ASPECTS AT MODEL LEVEL**

This would allow using redundancy; it requires system wide models. By having the run time environment, we have an exact view of system communications. Therefore potential error impacts can be finely traced.

### 8.5.3 **ERROR HANDLING AT VFB LEVEL**

In addition, exception handling could be performed at VFB level in a unified way, for the whole system.

### 8.5.4 **CONCLUDING REMARKS**

Following AUTOSAR is feasible. It is getting into series production. It is an enabler for future innovations concerning timing, error management, and safety.

Iterative prototyping is an appropriate way to boost the quality and acceptance of SW innovations.

## 8.6 DISCUSSION

*Q: you never refer to time in your talk; does it mean that SW architecture comes 1<sup>st</sup> and then only timing comes? Yes, in some sense. There are various aspects in a car. For some domains, time is important, not for other (entertainment). Important is to keep it simple and then think of timing aspects. There is a “timing group” addressing this problem in AUTOSAR.*

*Q: regarding possibilities to analyse models in your framework, are you attaching other aspects than functional to your components? Some attributes can be included.*

*Q: you say that AUTOSAR is an enabler; how about introducing new technologies in SW and scheduling techniques? AUTOSAR does not constrain you too much regarding behaviors; regarding schedulability I would be cautious because it concerns the whole system.*

*Q: how can you estimate WCET's with your very flexible and nondeterministic approach? Have statements and hypotheses related to the deployment entities and uses them for this.*

*Q: you have abstract communication models and physical devices for this; you have a tool for the mapping; if you want to change this assignment, to what extent can this be done incrementally in case of partial changes? I would have to define a complete mapping, but fortunately this occurs seldomly.*

*Q: are you sure that so-called “complex device drivers” will not act like Trojan horses for the AUTOSAR process (by allowing deviating from it)?*

*Engine control is not going to be developed according to the AUTOSAR methodology, several sources said.*

## **9. Stefan Sonck-Thiebaut (Carmeq GMBH) co-authors, F. Schöttler, Carmeq, and B. Kundel, VW AG: The AUTOSAR component model – canceled**

## **10. Kai Richter (Symtavision): the AUTOSAR timing model – status and challenges**

Symtavision is a young spinoff of TU Braunschweig by R. Ernst, founded in 2005. The company develops timing and scheduling analysis tool suite.

### **10.1 DISCLAIMER**

This talk presents personal viewpoints, not official positions of the AUTOSAR consortium.

### **10.2 AUTOSAR IN GENERAL & TARGET USE CASES**

- portable SW components
- VFB
- ports and connectors

Key AUTOSAR approach and mapping in more detail:

[slides 1 and 2 in appendix]

Standardised RTE eases compiling & linking together several SW components.

Typical use cases:

- functional distribution & partitioning, with changing architecture from application to computing modules
- adding new functions
- optimizations, re-mapping of SW components
- new business models and supply-chain organizations and liabilities.

### 10.3 TO-DOWN: SW ARCHITECTURES VS EXECUTION PLATFORMS

Timing effects:

- control functions impose timing requirements;
- this results in high-level specs on SW components;
- the AUTOSAR goal regarding this is to break down the SW structure into manageable blocks, including timing aspects;

[slide 3 in appendix]

(the described concepts are not standardized yet.)

### 10.4 ACLOSER LOOK AT TECHNICAL DETAILS; TALK AND DISCUSSION

SW components vs “runnables” and tasks

[slide 4 in appendix]

A **runnable** is a schedulable unit that can be concurrent with other runnables, unless dependencies exist. Runnables involve scheduling and timing dependencies.

Some information on the code inside is needed to perform subsequent scheduling, timing dependencies, and tasking organization. **The breaking of SW components into finer grain tasks results in a breaking of the original SW architecture:** how can this be traced? How can this be scheduled? You cannot decide how it will perform on a purely local basis, without some abstract model of the environment or other subsystems it interacts with.

*Werner Damm: Interfaces of runnables should indeed to be treated like interfaces of components?*

The reason for distinguishing between runnables and components is that components are units for reuse, whereas runnables are units for executions.

*Werner: but this is not a fundamental difference, it is rather a syntactic difference.*

This discussion leads to the core of the problem. Reasoning about timing could do well if questions were nicely answered. But, with SW components, there are conflicting issues, because the components architecture is different from the runnables architecture. Thus there are two views that are conflicting.

*Stefan K: This converges to the same picture that was already shown by Karl-Erik Arzen in his talk on re-structuring the code for execution and control loop performance.*

*Werner Damm: by opening up the possibility to deploy runnables instead of components, you open the need to keep track finely of information flow.*

Still, SW components are atomic with respect to the deployment over ECU's. Just, when scheduling comes into play, then one goes down to the granularity of runnables.

[slide 5 in appendix]

In principle this cannot be done prior to having the detailed execution architecture.

SW component structure versus timing dependencies:

- SW component views capture *logical* dependencies,
- but *timing* dependencies in the implementation can be very different and actually more difficult to have correct (type of communication, over/undersampling, TT or ET activation). This important point is not captured in the AUTOSAR component model.

*Pree (Wien): With your approach, if you add a component, you must redo the entire scheduling; this is terrible. This programming model is a mess.*

Protocols versus non standardized BSW: it is claimed priority based, ET, and flexible, but when you look closely at details, there are lots of timers. Overall, the analysis of this is a mess. Challenge: associating schedules with timing chain segments, taking into account complex mutual dependencies.

[slide 6 in appendix]

Key message: local changes can have a global effect, due to shared resources. Several cases are described in the slides.

*Q: The component should have attached a RT-analysis model, to facilitate RT analysis and reusability.* I fully agree with this. However, how can this be analysed, controlled, and designed? Even worse, sometimes you only have partial information on black-box components, with no information regarding RT behaviour: what can you do? How to analyze this at all?

*Werner Damm: In fact grey-box models are needed to do this.*

*Christian Salzmann: Yes, indeed. This is what I meant by AUTOSAR being an enabler. Such add-ons need to be built on top of it. But at the moment, such grey box information is simply not available and one has to live with this fact.*

## 10.5 BOTTOM-UP INTEGRATION & TIMING ANALYSIS PRACTICE TODAY

- Local analysis of individual components: good systematic approaches available, simplified environment models → later integration problems
- When testing (sub)systems after integration, the whole environment is available but critical interactions are unknown, which prohibits corner case coverage → decreases reliability of testing.

Today, integration is problematic. With bottom-up system integration:

- local decisions have global effect;
- system level modeling of complex timing interaction is needed;
- business issue: contracting;
- all this needs improvement.

## 10.6 IMPLICATIONS WRT AUTOSAR GOALS

AUTOSAR shall be a vehicle for modularity, portability, reuse, adaptability.

Unfortunately, timing is not as modular as the SW itself. SW architecture does not reflect timing dependencies. Timing is mapping dependent, not modular.

There is no point in modeling something that cannot be analyzed.

## 10.7 CONCLUSIONS

Many approaches for timing exist. No one has been chosen for AUTOSAR.

This happens just because corresponding issues are new to suppliers both technically and from business viewpoint. It is the responsibility of suppliers that they match the OEM's networking characteristics.

Regarding OEMs, networking effects are out of the suppliers' responsibility; there is a need to deal with contracts including QoS aspects to address this.

Overall, it is too bad that the research community has not been asked for assistance for a long time on this large set of problems.

[slide 7 in appendix]

## 10.8 DISCUSSION

*Q: is it possible to define timing requirements at RTE level or at SW level?* Yes. If you have several parties, a contract-based approach is desirable in which requirements/promises are manipulated. Such a contract based approach would be very helpful, but is very difficult to achieve.

*Q: there are solutions to analyze such situations, provided that communication media are used that keep the overall system analyzable; if you do so you can recover composability properties.*

*Q: suppose you trace constraints and you try to analyze the effect of changing something, is it a relevant situation? Yes this is an important use case.*

*Q from CS: I mostly agree with your concept of timing constraints. I am not sure what the appropriate model unit is, where to attach timing constraints? Is it an interface type? Do I attach it to types, or to instances? An interesting question, I do not have the solution in my pocket.*

*Q (Julio M.): I do have the answer in my pocket. You need to specify individual timing properties, shared resources, and flow of execution of all possible executions, in a parametric form. This goes along with types; when instantiating you attach values to the parameters. Of course, the suppliers need to expose more information regarding their subsystems. Also you need to have WCET for all possible configurations.*

*Q (Rolf Ernst): Control loops also should be part of this component analysis; you need to include control loops as part of the types.*

*Q: We have to understand timing constraints at system level, otherwise how can we get these at component level? Where do you put end2end latencies associated to functions?*

*Q (W Damm to CS): how would you migrate to AUTOSAR model, given the previous question? We cannot do this at the moment because we do not have timing model. We replace by extensive testing and validation on RT aspects. At the moment, such issues are not foreseen within the AUTOSAR scope.*

*Q (WP de Roever): this looks like a victory à la Pyrrhus: not sure that this whole machinery will simplify and not make things more complicated, just because hard difficulties have been hidden. In fact jumping over fundamental difficulties may be a step back, not a step forward.*

*Q (A. Ferrari): design space exploration is indeed needed where all these difficulties are really considered.*

*Q: You are advocating a process where there is a desire to arrive at a local analysis to derive interface models related to timing; and then at the system level linking is needed between these interfaces.*

*Q: in addition, if you have a lot of nondeterminism, then you won't even be able to reproduce behaviors and not be able to do any testing or analysis.*

*Q: does it mean that even if you have enough time budgets you have nondeterminism?*

## **11. Panel: Beyond AUTOSAR. A. Benveniste (INRIA, recording), W. Damm (Offis, moderator)**

Panelists are listed when they are recorded.

### **11.1 WHAT ARE THE KEY INDUSTRIAL CHALLENGES?**

#### **11.1.1 ALBERTO FERRARI: ROBUSTNESS TO CHANGE**

AUTOSAR is about decoupling functionality from platform. The platform must be reflected back at some appropriate intermediate level, where functions are also reflected (the ASV platform-based design triangles). There is a need to decorate components with rich aspects.

There is a need to face changing conditions that are not predictable at design time: designing for dynamic integration.

There is a need to capture the metrics for scalability and extensibility/robustness. You do not want to recompute the entire bus schedule when a small change occurs. There is a tradeoff between this objective and that of performance.

#### **11.1.2 ROLF ERNST: TIGHT INTERTWINING OF TIME IN PLATFORM AND FUNCTIONALITIES AS COMPONENTS**

I would like to have a layer, delivering some kind of guarantee to me. Is it possible? Yes but very inefficiently. There is a need for automotive platforms that are more extensible and flexible than existing infrastructures, and allow better for a clean separation of issues above



and below. Think of traffic shaping in Internet traffic. We should change our view regarding communication platform to get them better extensible by essence.

## 11.1.3 ROLF ERNST: PROBLEM WITH THE MANY SCENARIOS

They must be known by suppliers in the future, taking into account dynamic scenarios. The question is what kind of information do you need to capture there in order to capture those scenarios.

## 11.1.4 KAI RICHTER: DO WE NEED COMPLETE MODELS AT ALL?

Do we need just a framework based on formalisms, but no complete and detailed models?

## 11.1.5 MARTIN TÖRNGREN: MULTI-PARADIGM INTEGRATION, TIMING AND SAFETY/RELIABILITY

We should also remember that this also involves mechanics and control in addition to SW. There are different skills for this. Is it mature for convergence? Not quite. What theory can we provide to help for this?

Related to timing, there are techniques, not well transferred so far.

The situation is worse for safety/reliability. Safety is related to system, reliability is related to components. There are hidden design assumptions; there are HW/SW dependencies.

One has to consider functional level for design & reuse; SW component level is not sufficient.

Overall, there are concerns related to AUTOSAR:

- lack of systematic modeling and analysis, there is a need for functional reference architecture (*this is ongoing at AUTOSAR, CS; this is a very tricky issue; it is not considered a key challenge for AUTOSAR*);
- structured information management including SW, domain tool integration;
- elaborated design methodologies, systematic analysis;
- dynamic configuration: modes, fault tolerance, upgrades, integration; some kind of dynamic configuration has to be considered; question the basic hypothesis of AUTOSAR that all configuration is performed at design time.

## 11.1.6 JULIO MEDINA: COMPONENTS

We need to understand the target granularity of components. The effectiveness of modeling techniques for analysis and the scheduling space during deployment depend on this. We need to better understand what the target is. *CS: in fact it depends on context and functions.*

A contract based approach could be considered, such as in networks and processors, to separate scheduling concerns at deployment time, using the servers paradigm.

## 11.1.7 CHRISTIAN SALZMANN: SPECIAL CONDITIONS OF DEVELOPMENT PROCESS

The development process is very distributed with different roles. Having an AUTOSAR model that allows modeling the entire car with a component aspect, the question would be:

**How can we slice this information in such pieces that we can allocate the adequate pieces to the different roles, and without harming IPs and without knowing in advance some pieces of information that are coming later in the process?**

## 11.1.8 WERNER DAMM: SAFETY IS IMPORTANT AND SHOULD BE HANDLED LIKE TIMING

What has been said for timing can be roughly transferred and reformulated for safety. Safety must be captured in relation with the functions, at system level.

## 11.1.9 ROLF JOHANSSON: IS SAFETY A COMPONENT FEATURE?

It does not make sense to talk of safety at component level, since it is indeed a system level property. There are, however, elements impacting safety that are component level.

## 11.1.10 HEIKO DÖRR: ENABLING A SMOOTH TRANSITION OF ALL FUTURE BENEFITS INTO A SMOOTH EXISTING DESIGN ENVIRONMENT

The issue of smooth transition is important.

**11.1.11 HEIKO DÖRR: WE USE A LOT OF TOOLS WITH LOTS OF UNCERTAINTIES ABOUT THE MODELING ASSUMPTIONS, FIND APPROPRIATE RESTRICTIONS THAT WILL MAKE THE ANALYSIS FEASIBLE**

**11.1.12 HEIKO DÖRR: AT PRESENT WE HAVE WITH AUTOSAR SYSTEMS WHICH ARE HAVE SIMPLE ENVIRONMENT MODELS; UPCOMING SYSTEMS WILL BE MORE COMPLEX (HYBRID BEHICLES...), WITH STRONGER INTERACTIONS BETWEEN MECHANICAL COMPONENTS**

How can we find appropriate abstractions for such features, thus making components accessible with their mechanical aspects? Models of mechanical or combustion systems and plants should be part of modeling repository.

**11.1.13 WERNER DAMM: IS AUTOSAR GOING TO OPEN TO THE ACADEMIC COMMUNITY?**

CS: as far as I know, research labs can attend AUTOSAR as lessening members. I do not know what will be the strategy in the future. But this is not really an open strategy since you must sign all sorts of NDA. So the work is definitely not open. But this is not really an obstacle if you want really to go in. Maybe the academics are not sufficiently pushy.

*We propose that the present report will send the present research report to AUTOSAR and ask if this can be an opportunity for better interaction in the future.*

## **11.2 POSITION STATEMENTS REGARDING RESEARCH**

**11.2.1 JULIO MEDINA**

We have solutions regarding composability properties for schedulability analysis models. Of course there are requirements that can make such thing happening. It is not possible to do this for any kind of system. This could be used to model promises of components regarding timing. The sad part of the story is that we need to have instances of components to perform the analysis of the complete system; prototypes are not sufficient.

The distributed platform may also be modeled to complement the analysis, using a holistic approach.

**11.2.2 FRANÇOIS TERRIER**

**Numatec** automotive initiative in France: development of safety critical systems, allowing for a mix of levels of criticality; will be compliant to AUTOSAR. This complements with a research on traceability of requirements.

Another related project is of **SW factory**: management of heterogeneity and interoperability of tools needed to develop embedded systems. The corresponding platform will be open source (related to *OpenEmbedded*). AUTOSAR could take benefit from research done in this community.

Challenges are:

- semantic equivalence between the different formalisms, MoCCs
- how to model execution platforms
- standardization actions at OMG: MARTE profile for timing, safety, and architecture aspects of the system.

**11.2.3 ALBERTO FERRARI**

I am missing from AUTOSAR what the requirements are for timing. The architectural space that is addressed is huge. This could invalidate the whole approach. The academics needs to work on the real case and should therefore become aware of the AUTOSAR issues.

Finally, we also need a functional view of the AUTOSAR model.

**11.2.4 STEFAN KOWALEWSKI**

Any architecture is a tradeoff between different types of requirements. Up to now, main objectives for AUTOSAR have been portability, modifiability, distributability of SW and work, maintainability, security, reuse. AUTOSAR implies a different way of developing SW; there are some risks.

There are other non functional requirements that were not taken into account originally: timing, availability, robustness, safety. For the moment, these are handled afterwards in



later phases of design. This is risky. Therefore, we have to either refine the timing model, or add new views.

Another risk is that AUTOSAR does not get accepted. Possibly it may not be accepted. Maybe you will have complex device drivers instead. This is a risk: how to deal with this. Doing the timing analysis at the very end is not what is desired as we want to use models very early.

What is the suitable abstraction level? Can we do it at the level of the component or do we need to do it at a different level?

How can we evaluate the quality of a deployment? Can we explore this? Better, can we envision a synthesis approach?

At the moment, the management thinks that all problems will be solved by AUTOSAR, which is clearly not true.

**Overall, my wish is that the AUTOSAR community would be less autistic.**

## 11.2.5 ROLF ERNST

AUTOSAR provides a modular and flexible SW integration platform. This was a necessary step. It is in large part based on a client-server mechanism: there is currently no solution within AUTOSAR regarding the timing model.

I agree with SK that the management has a wrong perception that all problems have been solved by AUTOSAR. But:

- timing dependencies are mapping dependent
- the dependencies are fundamental and will not disappear with time; Flexray helps but is not sufficient.

What can we do?

- be “conservative”, adopt the TT view; control performance and cost issues, integration issues (have a good feeling of the car)
- use formal models and strategies to control timing, use advanced and predictable scheduling, control jitter and delay, avoid integration legacies, analyze and adapt the system carefully (requires models and tools); establish timing and QoS contracts between suppliers and OEMs.

Formal techniques: revolution or evolution?

- Most basic data are available regarding timing; measurements exist, why not using them?
- AUTOSAR introduction can pave the way, timing contracts are needed; fix liability issues;
- this is an engineering evolution, but a management revolution

Is AUTOSAR in good shape? Not really:

- There will be much SW developed now that does not adhere to or is qualified according to any timing standard
- AUTOSAR urgently needs a timing standard now

**The revolutionary step would be a systematic consideration of realistic HW timing and execution platform control strategies in SW engineering.**

## 11.2.6 SUSANNE GRAF

AUTOSAR is a basic 1<sup>st</sup> step. Deeper semantics issues must be considered as well. We want to have a framework allowing for guarantees, without imposing too many constraints. There is a need for studies on a general theory of architecture abstractions. We need a theory of predictability in environments that are not predictable. We want to compose already validated parts of the system without revisiting things. We need correctness-by-construction results for generic properties such as deadlock-freedom, liveness, and safety. We need to provide support for component integration and generation of glue code meeting given requirements.

## **11.2.7 ROLF JOHANSSON (MENTOR GRAPHICS): ON TIMING MODEL**

From my experience with VOLCANO, I can say that having a timing model is very difficult (in my former experience, we almost had it but we stopped because it was felt too complicated). Communicating timing information is very difficult. This I see as a real challenge for the academic community to offer such type of requirements language.

## **11.2.8 WILLEM-PAUL DE ROEVER: VERY QUICKLY COMPOSITIONALITY REACHES A DEAD END**

It is known that studies on compositionality ends up requiring to know everything about the environment, contrary to what it is meant to.