I

HARD REAL-TIME SYSTEMS I

H Kopetz

Rapporteur: A I Kistijantoro

Hard Real-Time Systems I

Hermann Kopetz
TU Wien

August 2001

Outline

- ♦ Introduction
- ♦ Why do Computer Systems Fail?
- ♦ Temporal Accuracy of Information
- ♦ The TT-Model
- ♦ Temporal Firewalls
- ♦ Two-Phase Design Methodology

Technology Change: Challenge or Crisis?

It is only a question of "**when**", and not of "**if**" different industries will be forced to make the transition form mechanical/hydraulic control systems to computer-based control systems.

Tomorrow will not be like today.

The aircraft industry has managed this transition successfully in a high-dependability environment. This implies that the technologies to built high-dependability electronic systems are available.

The automotive industry will be next :
"Drive-by-Wire" will follow "Fly-by-Wire"

What can you do Today with 1 mm^2 of Silicon

- Build a 32 bit wide processor (e.g., the ARM 7 processor)
- implement 100 k-bytes of memory (e.g., the 256 Mbit memory chip from Infineon is less than 100 mm^2).

Today, the marginal production cost (without IP, packaging,etc.) of 1 mm2 of silicon is in the order of 10 US cent.

Communication capabilities increase even faster than processing capabilities.

Moore's Law Lives

5

Intel announced technology that can shrink circuits even further- keeping the chip-speed rule on track through 2007, or even 2009.

At a conference in Kyoto, Japan, Intel displayed transistors, or circuits, only 70 to 80 atoms wide. This nanometer technology should lead to low-power chips containing 1 billion transistors running at speeds of 20 GHz. (Today's fastest Pentium 4 models have 42 million transistors and run at 1.7 GHz.)

The coup de grace: These feats can be accomplished using current chipmaking equipment, not with future innovations.

THE INDUSTRY STANDARD MAGAZINE, Mark Boslet, Date: Jun 25, 2001

Consequences of Moore's Law

6

- Hardware cost will be dominated more by the number of packages, then by the functionality of the silicon real-estate in each package.
- Separation of mixed signal chips (e.g, smart MEMS transducers) from large logical processing chips.
- The use of the smart sensor technology will increase.
- Distributed architectures are the only alternative.
- Because of the decreasing feature size, transient hardware faults will increase--need to provide fault-tolerance.

Why Do Computer Systems Fail?

7

- **Internal Physical Faults:** The cause of the failure is, e.g., a physical aging process within a chip. Can be transient (soft) or permanent. *It can be assumed, that multiple failures of chips are statistically independent*--will increase due to reduction of feature size.
- **External Physical Faults:** The cause of the failure is a disturbance external to the chip, e.g., EMC, spikes in the power supply, mechanical shock. Can be transient or permanent. *It **cannot** be assumed that multiple failures of chips are statistically independent.*
- **Design Faults (Software Faults):** The cause of the failure is the design (software or hardware) resulting in inconsistent states and actions. Different components of the same design will fail at the same instant.

 TTA

What are the Mechanisms to Handle these Faults?

8

- **Internal Physical Faults:** Systematic or application specific redundancy, e.g., replication of components. Should not increase the complexity of the application software.
- **External Physical Faults:** High-quality engineering, replication is not the solution, since failures are statistically dependent.
- **Design Faults (Software Faults):** Reduction of the Complexity of a Design--making a design understandable.

 TTA

Fault Hypothesis

9

At the beginning of a design of a fault-tolerant system, the *fault hypothesis* must be specified:

- Divides the fault-space in two partitions: "normal faults" and "rare events".
- States the *assumptions* about the types and frequencies of "normal faults", that the fault-tolerant system must tolerate.
- The probability that these assumptions are met by reality is called the *fault-assumption coverage*.
- Experimental evidence must be collected, that the fault assumption coverage is sufficiently high--i.e., that rare events are indeed "very rare".

If the fault assumptions do not hold, then the fault-tolerant system may fail.

 TTA

Faults Outside the Fault Hypothesis?

10

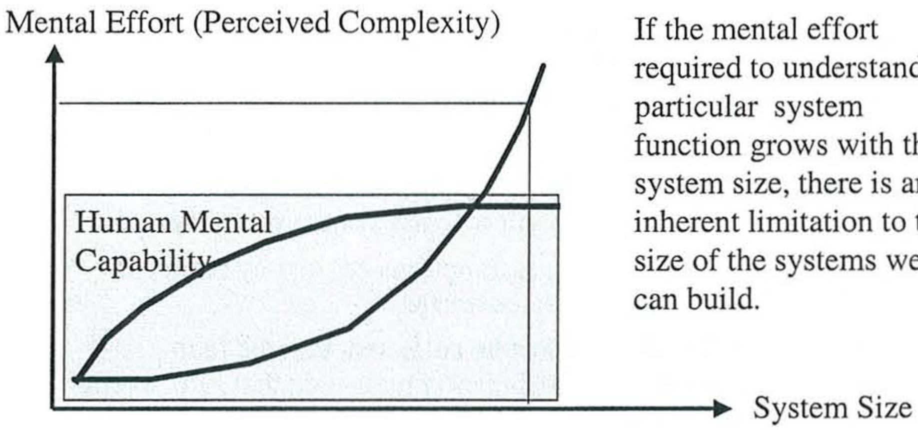
- Must be ***rare*** events, e.g., they should not occur ***once*** during the lifetime of a vehicle.
- Immediate Detection by appropriate algorithms: *"Fast Detection of an Error is an important part of the solution*".
- Never-Give-Up (NGU) Strategy for example: Freeze the actuators and restart the system with the remaining functioning components with a clean state within a defined restart time (e.g., 10 msec).
- Detect and avoid state corruption: There is a high probability that state corruption will lead to inconsistent *safety-critical* behavior.

If faults outside the fault hypothesis are not *rare* events, there is something fundamentally wrong with a design--such a system should never be deployed in safety critical applications.

 TTA

Design Faults: Reduction of Complexity

11



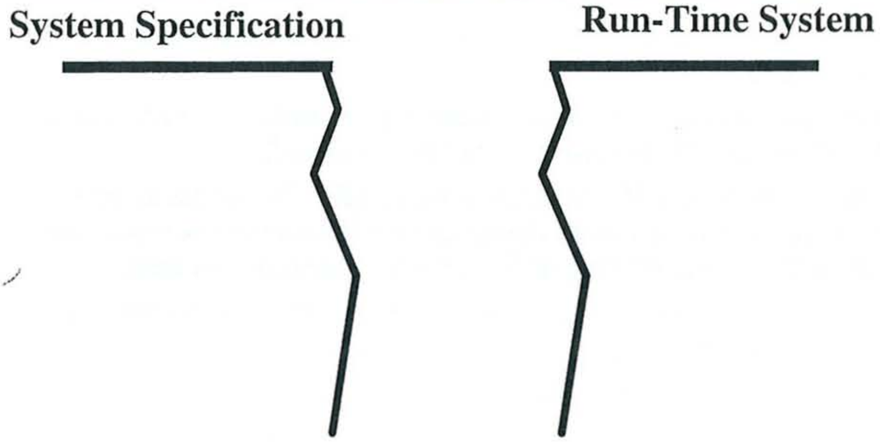
If the mental effort required to understand a particular system function grows with the system size, there is an inherent limitation to the size of the systems we can build.

Design faults have their root in unmanaged complexity.

 TTA

Design Faults: The Implementation Gap

12



Reduce the complexity of a design by basing specification and implementation on the same set of basic concepts.

 TTA

13

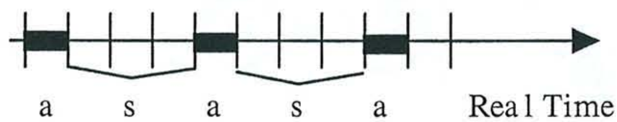
Basic Concepts of Real-Time Systems:

- **Physical Time**: There is only one physical time in the world and it makes a lot of sense to assume that this physical time is available everywhere in a RT system.
- **Deadlines:** A real-time task must produce results before the deadline--a known instant on the timeline--expires.
- **Time-bounded validity of real-time data:** The validity of real-time data is invalidated by the progression of real-time.
- **Distribution and Communication:** Smart sensors and actuators are nodes of a distributed real-time computer system. Communication in a distributed system takes real time.

14

Dense Time versus Sparse Time

It is impossible to perfectly synchronize the clocks at the nodes of a distributed computer system.



a duration of activity
s duration of silence

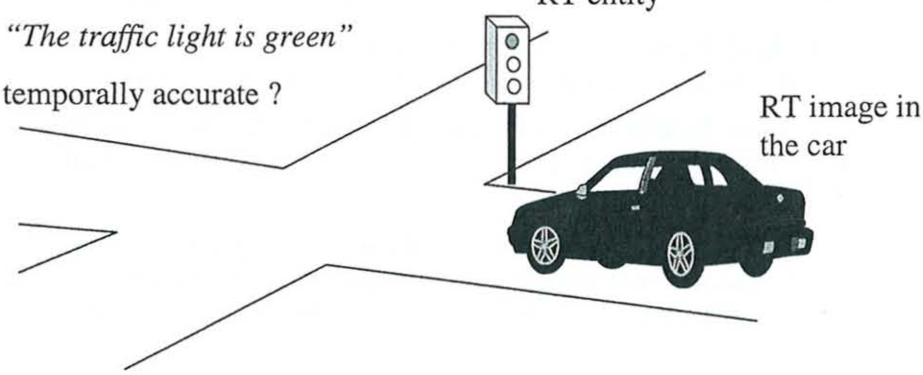
Duration of activity determined by the granularity of the global time

Real-Time Information

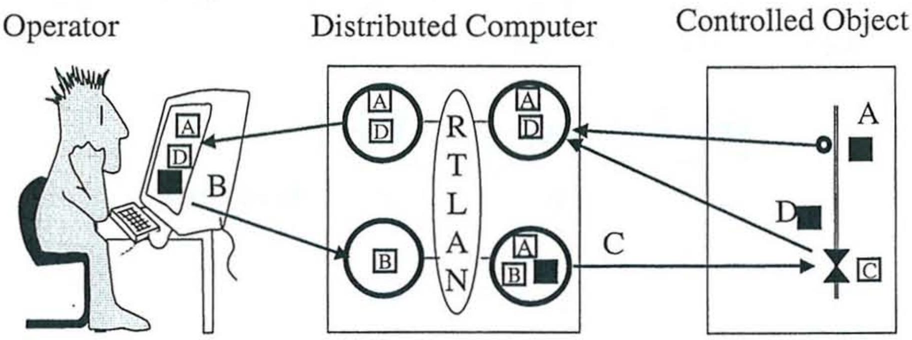
How long is the RT image, based on the observation:

"The traffic light is green"

temporally accurate ?



RT Entities, RT Images and RT Objects



■ RT Entity □ RT Image

A: Measured Value of Flow B: Setpoint for Flow
C: Intended Valve Position D: Actual Valve Position

Definition: Temporal Accuracy

17

The temporal accuracy of a RT image is defined by referring to the recent history of observations of the related RT entity. A recent history RH_i at time t_i is an ordered set of time points $<t_i, t_{i-1}, t_{i-2}, \ldots, t_{i-k}>$, where the length of the recent history

$$d_{acc} = t_i - t_{i-k}$$

is called the temporal accuracy. Assume that the RT entity has been observed at every time point of the recent history. A RT image is temporally accurate at the present time t_i

if

$$\exists t_j \in RH_i: Value(\text{RTimageatt}_i) = Value(\text{RTentityatt}_j)$$

 TTA

Observation of a RT Entity

18

State observation:

<Name of RT entity, Time of observation, full value>

The flow is at 5 l/sec a 10:45 a.m.

Event Observation:

<Name of Event, Time of event occurrence, state difference>

The flow changed by 1 l/sec at 10:45 a.m.

 TTA

State versus Event Observations

19

Characteristic	State Observation	Event Observation
Value	Full Value	Value Difference
Frequency	Periodic	Sporadic
Loss of Observ.	Period lost	Loss of synchr.
Semantics	At-least-once	Exactly-once
Error Detection	At receiver	At sender only

Time-Triggered (TT) Model (I)

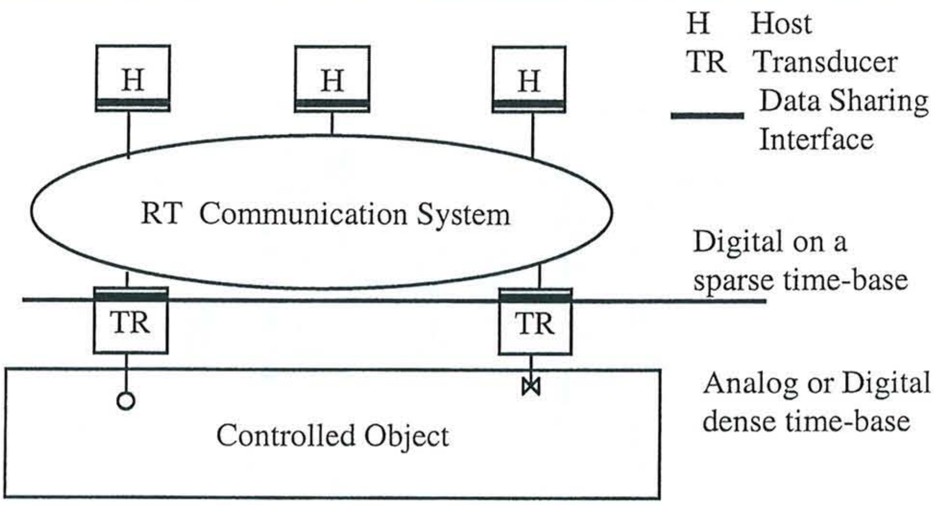
20

Assumes existence of a *sparse global time* and builds on four basic concepts:

- **Interface**: a data-sharing boundary between two communicating subsystems that contains *temporally accurate state observations.*
- **Communication subsystem**: transports real-time data from an output interface to an input interface within a given time.
- **Host computer**: Reads input data from an input interface, performs a data transformation and writes output data into an output interface within a given time.
- **Transducer**: Transforms output data from an interface into a form required by the system environment and transforms data from the environment into the form required by an input interface.

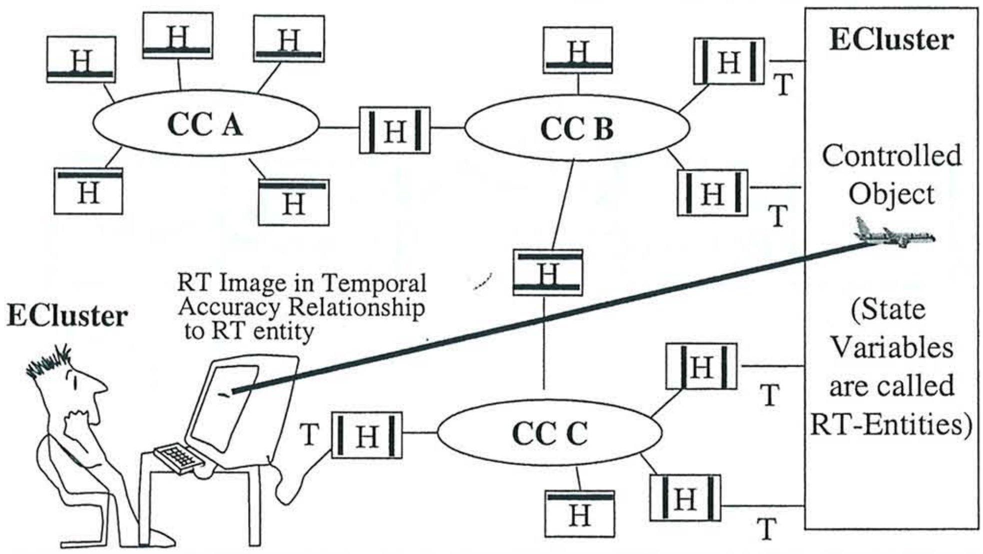
TT Model (II)

21



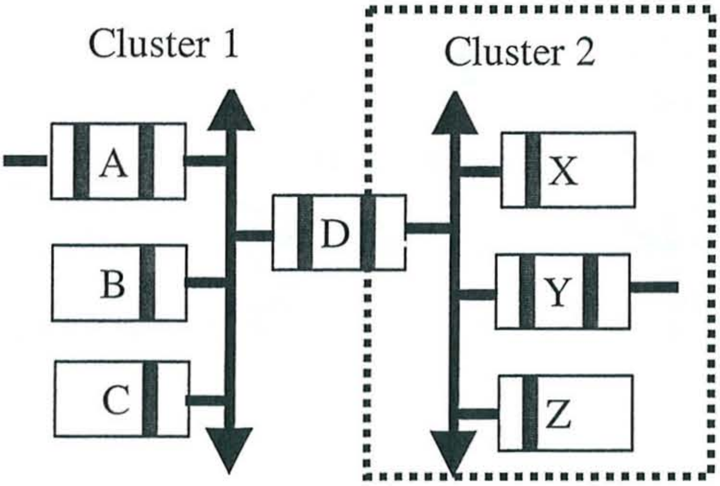
Example: A Five Cluster System

22



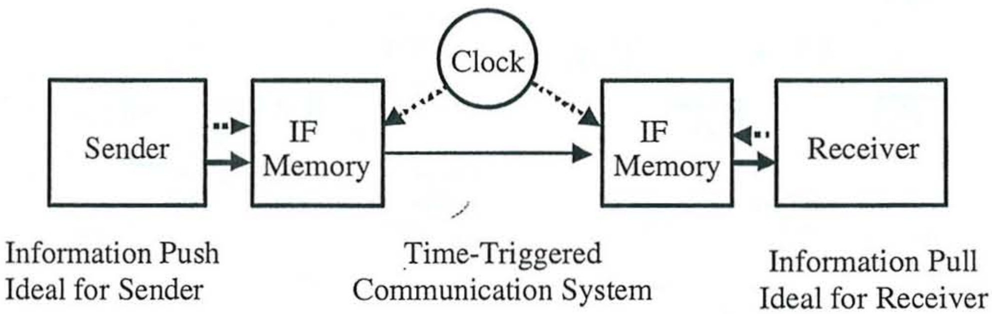
Localized View of Global System

23



Information Transfer in the TT-Model

24



Concept of a *Temporal Firewall*

25

A *temporal firewall* is a unidirectional data-sharing interface with state-observations in the interface memory where at least one of the interfacing subsystems accesses the temporal firewall according to an *a priori* known periodic schedule.

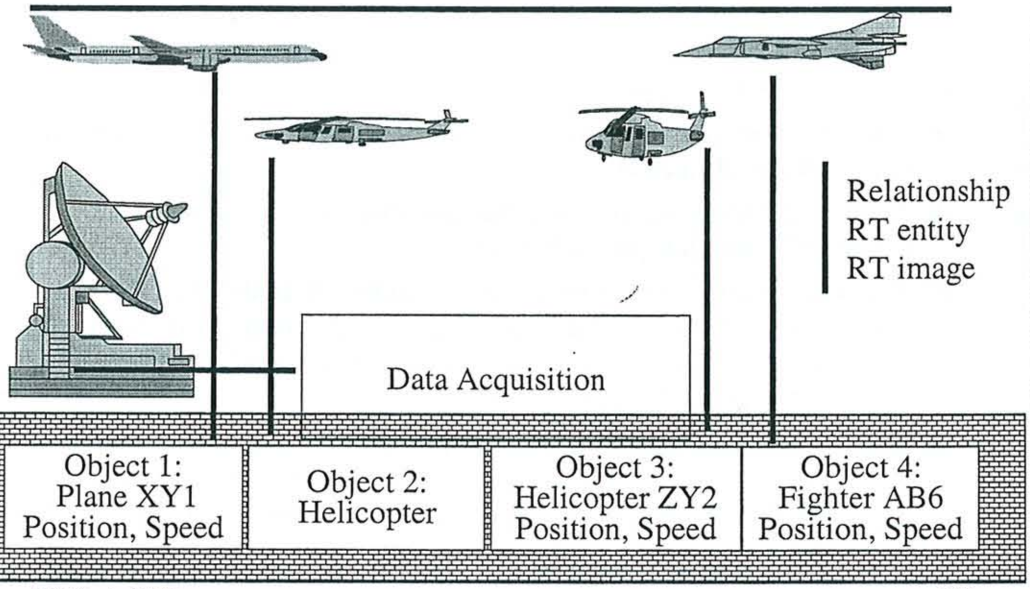
The interface between the host computer and the communication system can be seen as erecting two unidirectional temporal firewalls:
an *input firewall* and an *output firewall* .

A temporal firewalls eliminate control error propagation by design.

TTA

Example of a Temporal Firewall

26



TTA

27

A Temporal Firewall is a Natural Concept

- A temporal firewall is a *high-level* abstract concept.
- It is a small and stable unidirectional interface that provides understandable abstractions of the relevant properties of the interfacing subsystems.
- *Timeliness* is an *integral* part of the temporal firewall concept.
- Conceptually, the RT images in the temporal firewall are closely related to the image presented by a sensor of an analog RT entity in the environment.
- Temporal firewalls are thus based on an accustomed view of the world.

 TTA

28

Stable Properties of Temporal Firewalls

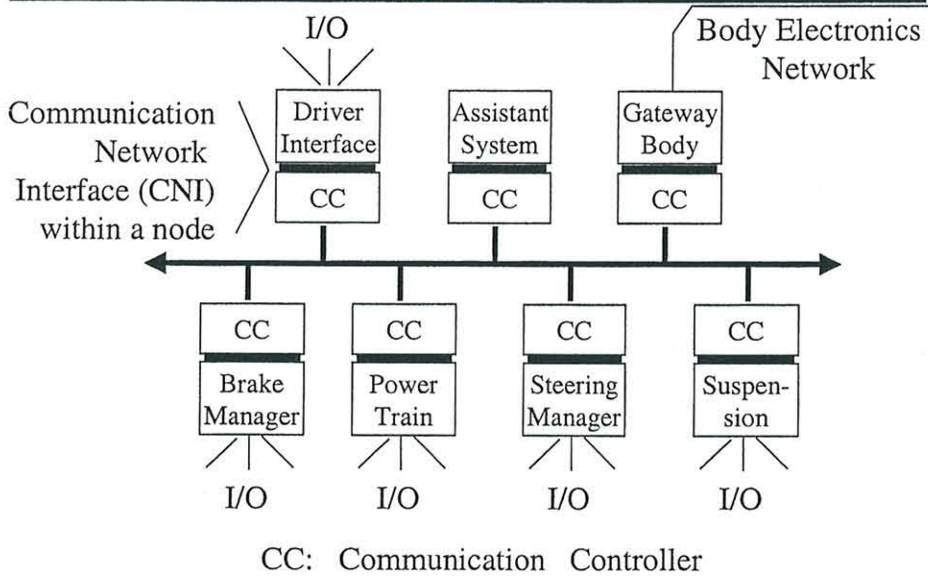
The following stable properties of temporal firewalls are known *a priori* to all interfacing partners:

- The addresses (names) and the syntactic structure of the data items in the temporal firewall.
- A (abstract) model explaining the meaning of the data items contained in the temporal firewall.
- The points on the global time base when the data items in the temporal firewall are accessed by the TT communication system. This information enables the avoidance of race conditions between the producer and the consumer.
- The *temporal accuracy* of the data items in the temporal firewall. This knowledge is important to guide the information consumer about the minimum rate of sampling of the temporal firewall.

 TTA

Temporal Firewalls in an Automotive System

29



 TTA

Temporal Firewall Contents

30

ECU	Data Elements in Input Firewall	Data Elements in Output Firewall	Remarks
Driver Interface	Status Information about Vehicle	Intended Direction (Steering Wheel Angle) Intended Brake Force Accelerator Pedal Position Intended Gear	Will not be accepted by assistant system if outside the envelope of safe vehicle performance.
Assistant System	Driver Intentions Status Information about Vehicle Environment Information (e.g. Vision System), yaw rate	Setpoints for Steering, Braking and Power Train Control according to global view.	Conflict with driver's intention has to be resolved.
Gateway Body Electronics	Status Information about Vehicle	Vehicle Access Control Status (Theft Avoidance)	
Brake Manager	Desired brake force setpoints for the four wheels Yaw rate information Vehicle status information	Actual Brake force on each of the wheels Actual wheel speed	
Power Train	Intended Engine Torque Intended Speed Environment Parameters Intended Gear	Actual Engine Performance Parameters Actual Gear Information	
Steering Manager	Intended Direction (Steering Wheel Angle from Driver) Vehicle Status Information	Actual wheel position	
Suspension	Vehicle Status Information (Steering Angle, Yaw rate, brake force)	Vertical Vehicle Position	

Source: Kopetz and Thurner, SAE Paper 981107, 1998

 TTA

Temporal Firewalls and Composability

31

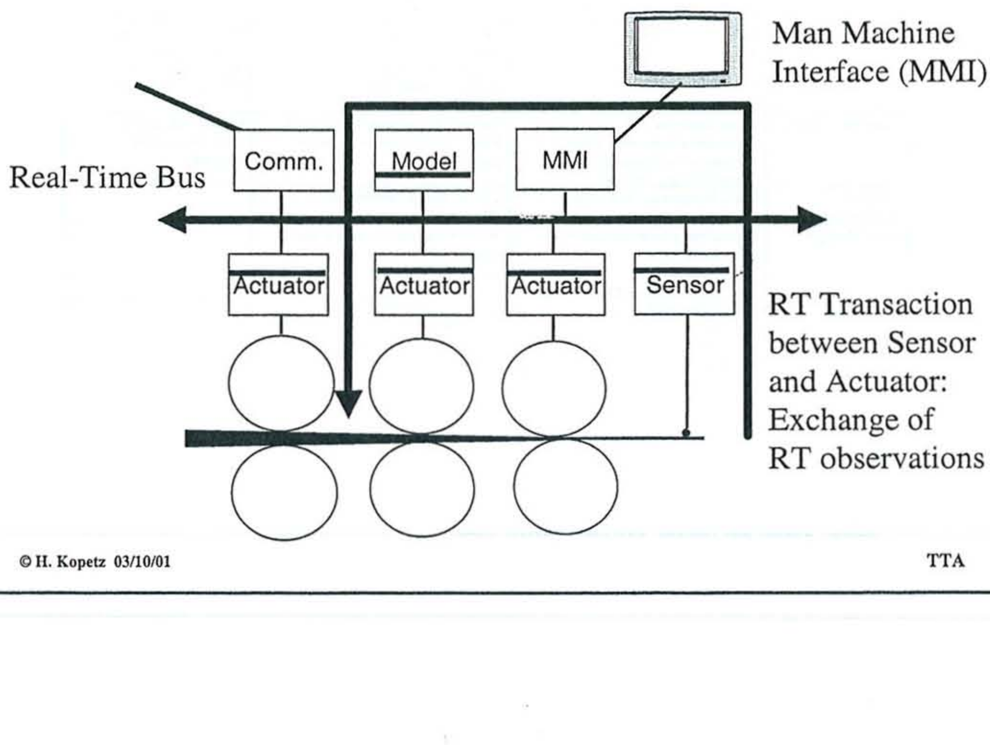
A composable architecture must support the

- *Independent development of components*--relates to the architecture
- *Stability of prior services*--relates to the components
- *Constructive integration of components*--relates to the communication system.
- *Replica determinism*--to support transparent implementation of fault tolerance.

The temporal firewall concept supports these principles of composability.

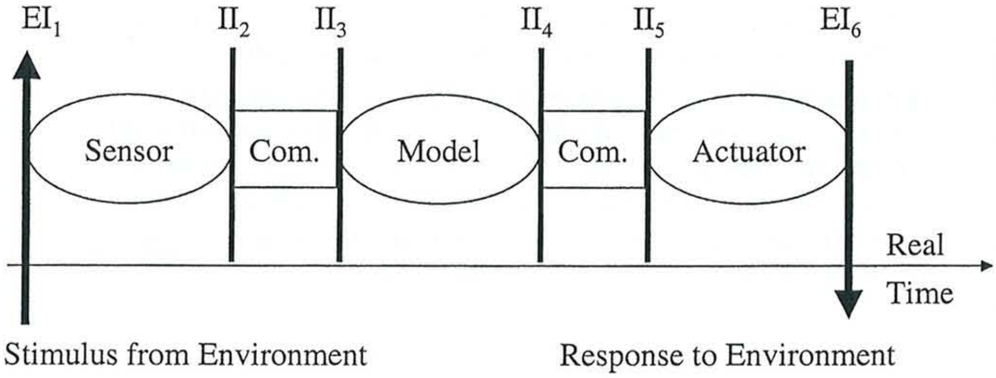
An Example: Rolling Mill

32



33

Real-Time Transaction



If the intermediate interfaces are not fully specified in the temporal domain, composability cannot be achieved.

 TTA

34

Temporal Firewalls and Validation

Assume a host that is encapsulated between two temporal firewalls, and *input firewall* and an *output firewall.* These two firewalls form the only interfaces of this host to its environment.

- The stable properties of the input firewall form important *preconditions* for the validation of the component under consideration. Many assumptions about the environment are contained in the specification of this input firewall.
- The stable properties of the output firewall form important *postconditions* of the validation.
- In the validation process it must be demonstrated that the postconditions, given in the output firewall specification, are always TRUE, provided the preconditions associated with the input firewall hold.

 TTA

Obligations of the Subsystems

35

- **Producer**: The producer of the RT-images stored in the temporal firewall is responsible that the *a priori* guaranteed *temporal accuracy* of the RT-images is always maintained.
- **Consumer**: Based on the *a priori* knowledge about the temporal accuracy of the RT images in the temporal firewall, the consumer must sample the information in the temporal firewall with a sampling rate that ensures that the accessed information is temporally accurate at *its time of use* of this information.

 TTA

Two level Design Methodology

36

The Time-Triggered Model supports a two level design methodology:

System Level specifies the interactions by designing the Temporal Firewalls:

- Data items that are exchanged among the subsystems
- Abstract model of the meaning of the data
- Instants when the TT communication system accesses the data

Component Level is concerned with the Host Software Design:

- The host computer provides the intended function, taking the available temporal firewall specification as constraint.
- Validation with respect to the temporal firewalls.

 TTA

Top Down Design

A top down design proceeds as follows:

- partition the system into a set of nearly autonomous components
- specify the functionality of the components at an high application specific level
- specify the timely information flow among the components, the interaction patterns
- specify the instants when information in the temporal firewall is accessed by the TT communication system.

The first design phase results in the specification of temporal firewalls between the host computers and the communication system.

Bottom-up Design--Reuse of Components

The bottom up design takes advantage of existing prevalidated components:

- The input temporal firewall parameters determine what a user is expected to supply
- The output temporal firewall parameters determine what a user can rely upon

The system design that specifies the interaction patterns must proceed taking these component characteristics as constraints.

Conclusion

The time-triggered model of computation provides a set of concepts and a methodology for the specification of a distributed hard real-time system:

- Global time
- Temporal Firewalls
- Host Computers
- TT Communication System
- Transducers

What we now need is a distributed RT architecture that provides the associated framework for the execution of a design expressed according to the concepts of the TT model.

DISCUSSION

Rapporteur: A I Kistijantoro

Lecture One

Dr Ezhilchelvan suggested that a state based model is a refinement of an event based model, and pointed out that missing messages in an event based model are much more critical than missing messages in a state based model. Professor Kopetz agreed with Dr Ezhilchelvan and replied that there are trade-offs between the two models and one cannot say which model is better, because it depends on the kind of applications. The majority of computing systems, like PAR protocol or TCP/IP protocol, use event based models, but now it is quite agreed that hard real time controlled systems use state based models.

Professor Burns suggested that the classification of state based model and event based model contains a number of concepts that can be decomposed further. On one level there is an issue whether the communication is periodical or sporadic, and on another level there is an issue whether the communicated data is absolute or relative. These are different concepts that can be mixed in different architecture ways. Professor Kopetz agreed and said that he only pointed out the two extremes, state and event, but there are intermediate points between them.

Professor Campbell asked to add another dimension in the diagram of communication systems, instead of just the state based and the event based. Professor Kopetz replied that on the implementation level, there are a number of choices. One can build state messages on top of event messages, or part of the messages are event messages and part are state messages etc. But he meant the diagram is at the conceptual level, not at the implementation level.

Professor Suri asked about the limitation of time triggered technology and when not to use it. According to Professor Kopetz, time triggered technology is not suitable when there are very sporadic messages, and the nature of information is not time triggered. Also in the case when jitter and delay is not an issue, event messages are better from the point of view or resource utilization.

Professor Schneider discussed the compositionality issue of the time triggered architecture, especially on how the clusters are composed into one system when each cluster has different time base or granularity. He questioned that if the granularity is different and the algorithm depends on some sampling and hooks up to the clusters with wrong granularity then it won't work correctly.

Professor Kopetz emphasized that there is only one time base in the time triggered architecture, but each cluster may have different granularity of the time. The correctness of the whole system depends on the specification matching of each subsystem. One of the principles of composability is that if one integrates a component into a context, the prior service of that sub system must still work after the integration. If the component requires the time granularity of certain microseconds, then it should be put in the context that satisfy that specification, otherwise one can not expect the integrated system to work properly.