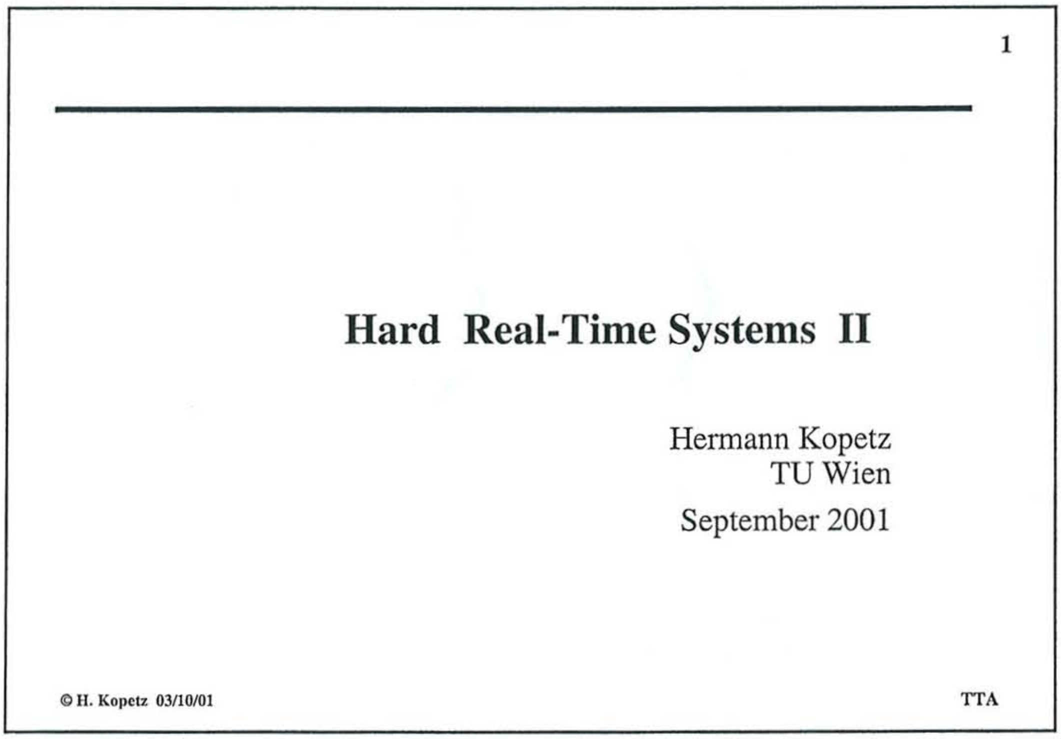
HARD REAL-TIME SYSTEMS II

H Kopetz

Rapporteur: A I Kistijantoro

I.26



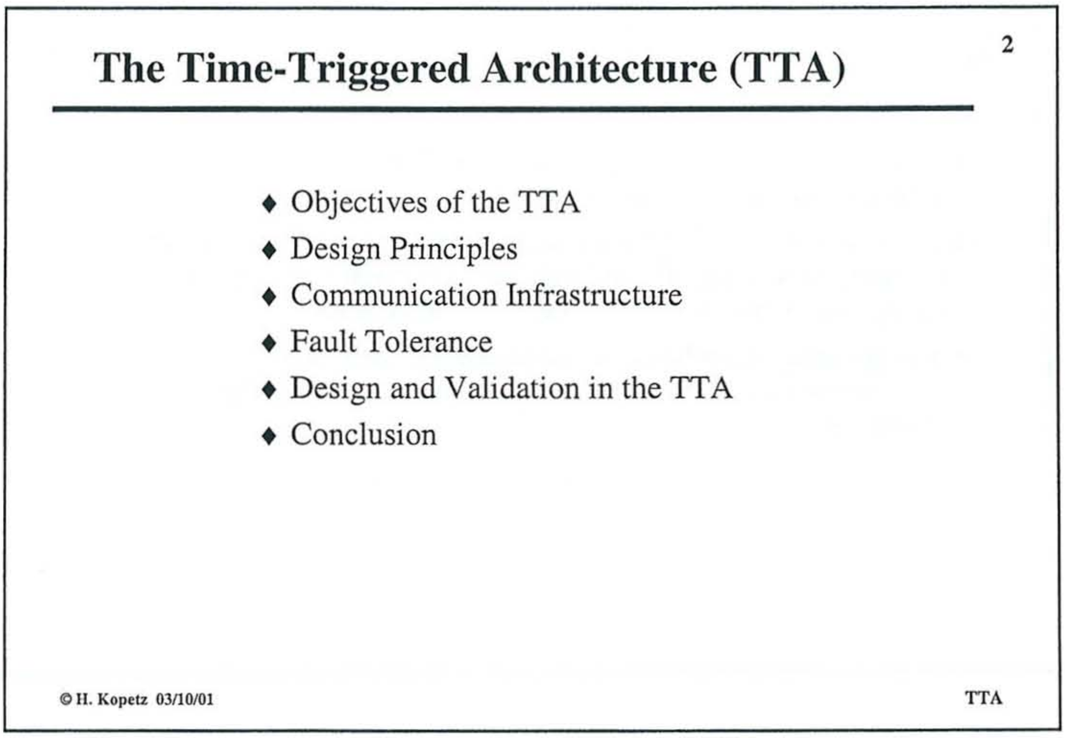
1

Hard Real-Time Systems II

Hermann Kopetz
TU Wien
September 2001

© H. Kopetz 03/10/01

TTA



The Time-Triggered Architecture (TTA)

2

- Objectives of the TTA
- Design Principles
- Communication Infrastructure
- Fault Tolerance
- Design and Validation in the TTA
- Conclusion

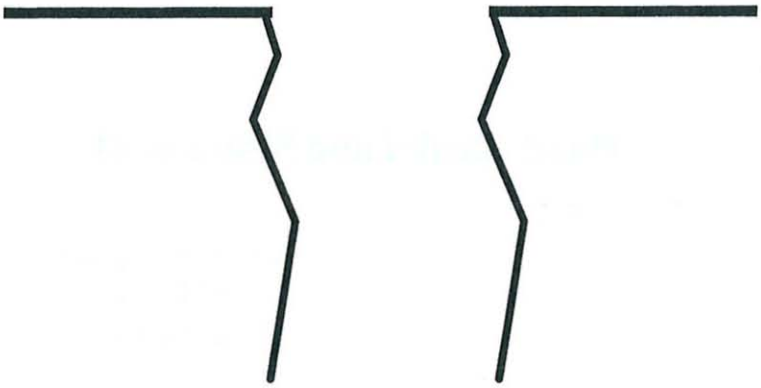
© H. Kopetz 03/10/01

TTA

3

Design Faults: The Implementation Gap

System Specification **Run-Time Architecture**

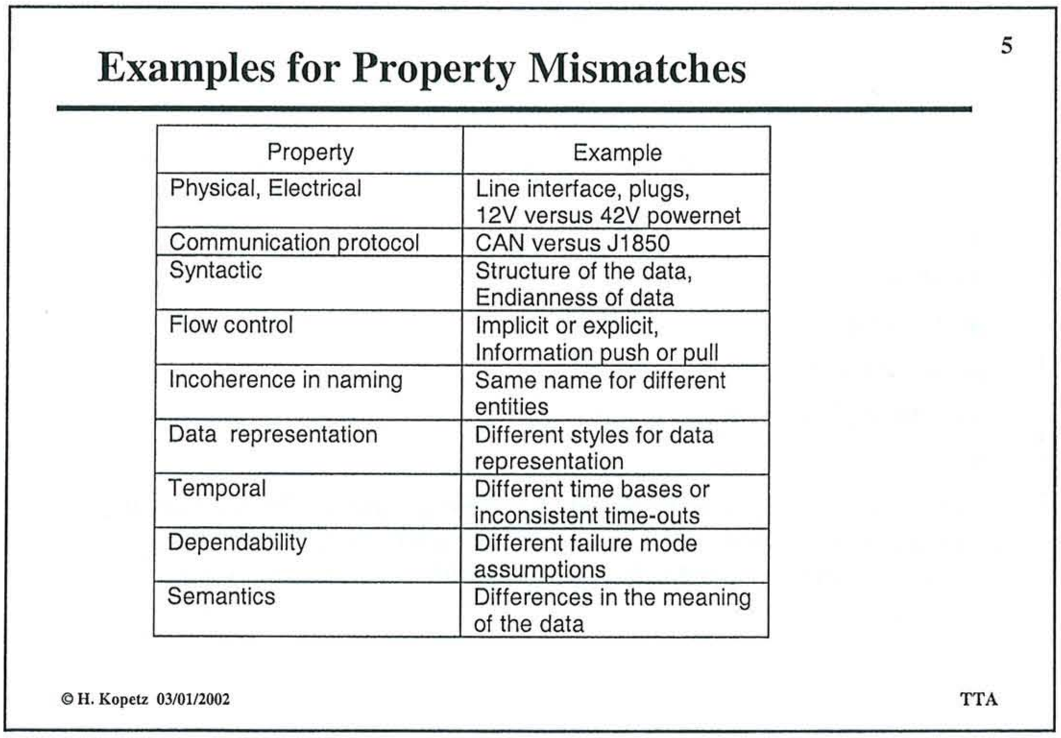


Reduce the complexity of a design by basing specification and implementation on the same set of basic concepts.

4

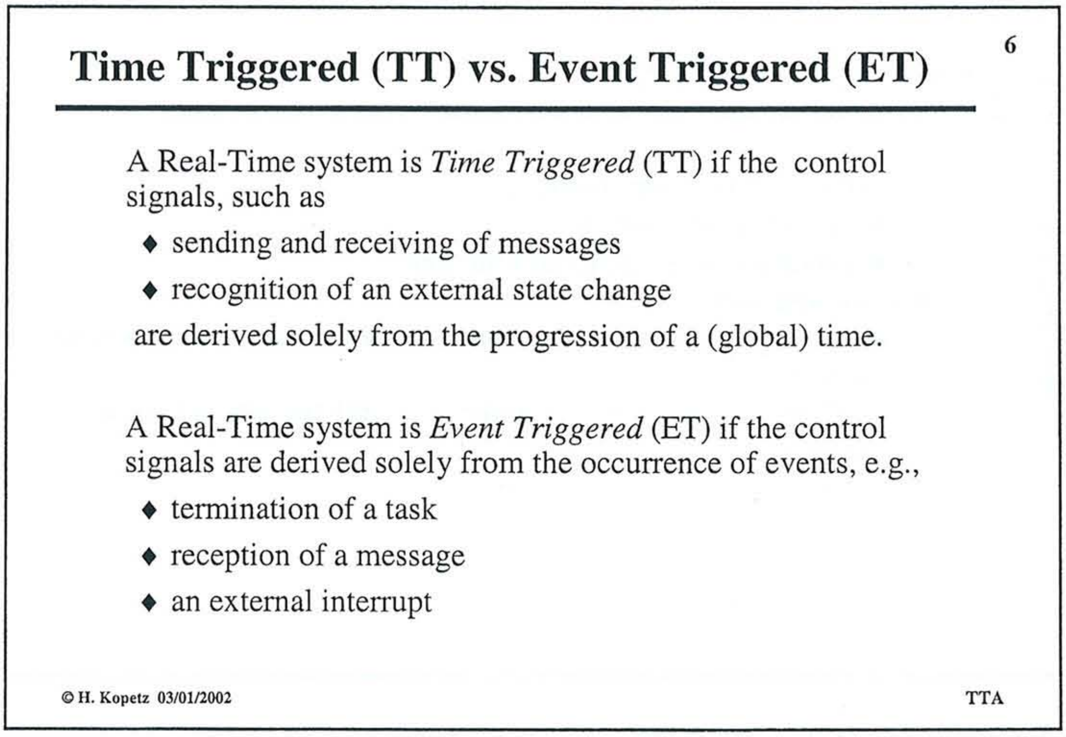
Technical System Architecture

- An architecture is a blueprint and a framework for the construction of a system out of subsystems (components).
- Architectural style: The architecture must provide guidelines for the partitioning of a system into subsystems and for the design of the interactions among the subsystems.
- Components must be designed to comply with the *architectural style* to avoid property mismatches at the interfaces.
- An architecture must *constrain* an implementation in such a way that the ensuing system is understandable, maintainable, extensible, and can be built cost-effectively.



Examples for Property Mismatches

Property	Example
Physical, Electrical	Line interface, plugs, 12V versus 42V powernet
Communication protocol	CAN versus J1850
Syntactic	Structure of the data, Endianness of data
Flow control	Implicit or explicit, Information push or pull
Incoherence in naming	Same name for different entities
Data representation	Different styles for data representation
Temporal	Different time bases or inconsistent time-outs
Dependability	Different failure mode assumptions
Semantics	Differences in the meaning of the data



Time Triggered (TT) vs. Event Triggered (ET)

A Real-Time system is *Time Triggered* (TT) if the control signals, such as

- sending and receiving of messages
- recognition of an external state change

are derived solely from the progression of a (global) time.

A Real-Time system is *Event Triggered* (ET) if the control signals are derived solely from the occurrence of events, e.g.,

- termination of a task
- reception of a message
- an external interrupt

7

Vision of the Time-Triggered Architecture

Development of a generic architecture for high-dependability distributed real-time systems that can be applied in the many different application domains

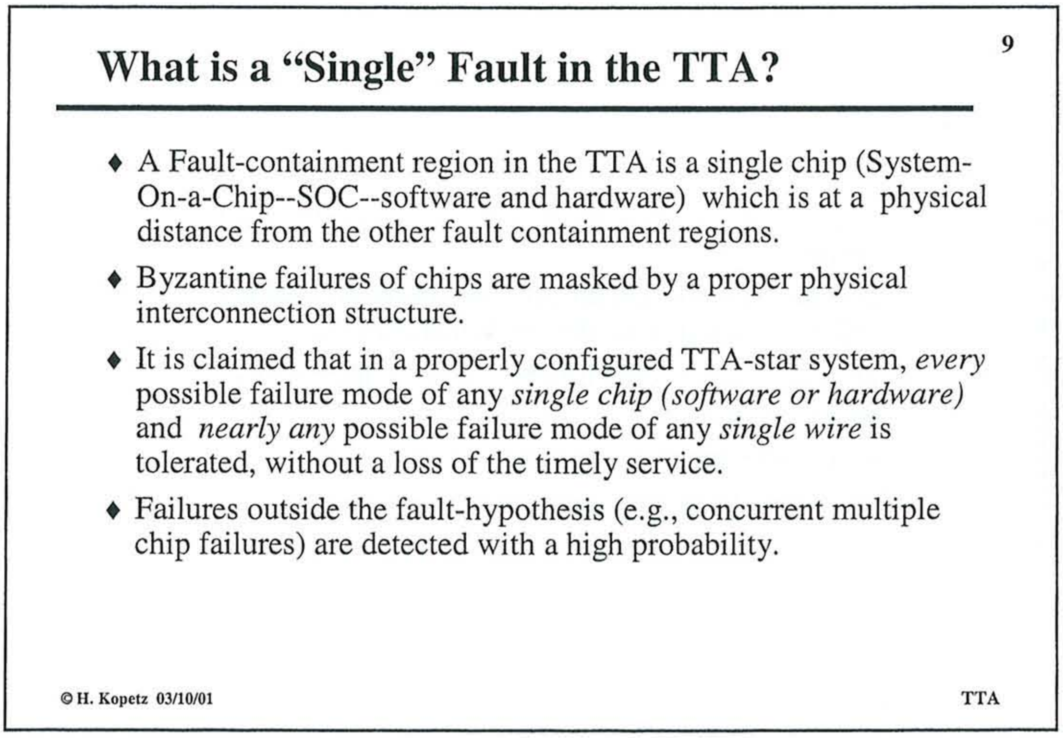
- Automotive
- Aerospace
- Railways
- Industrial Control
- . . . .

Our vision comes closer to reality by the decision of Audi to use the TTA in the automotive domain, by Honeywell to use the TTA in aerospace domain, and by Alcatel to use the TTA in the railway domain.

8

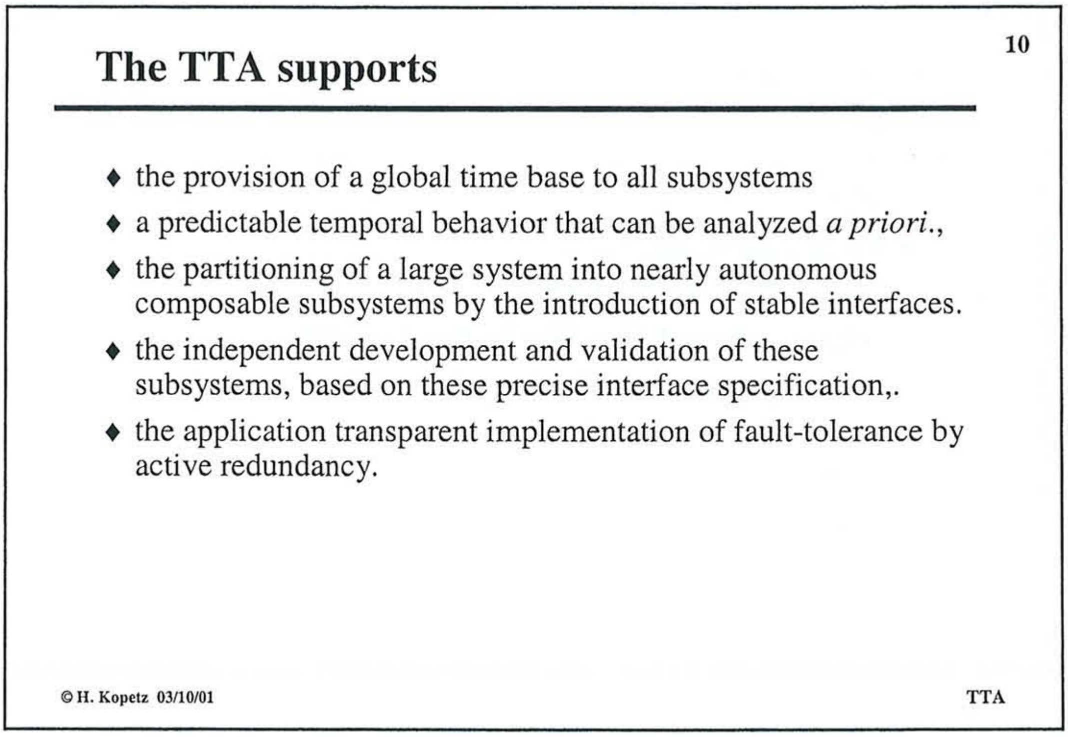
Priorities in the TTA

- **Safety without compromises**
  - No single point of failure
  - Formal analysis of critical functions
- **Composability:**
  - Building systems out of prevalidated components--Component reuse
  - Fully specified interfaces in the temporal domain and value domain
  - Two level design methodology
- **Flexibility**
  - Flexible reuse of existing components



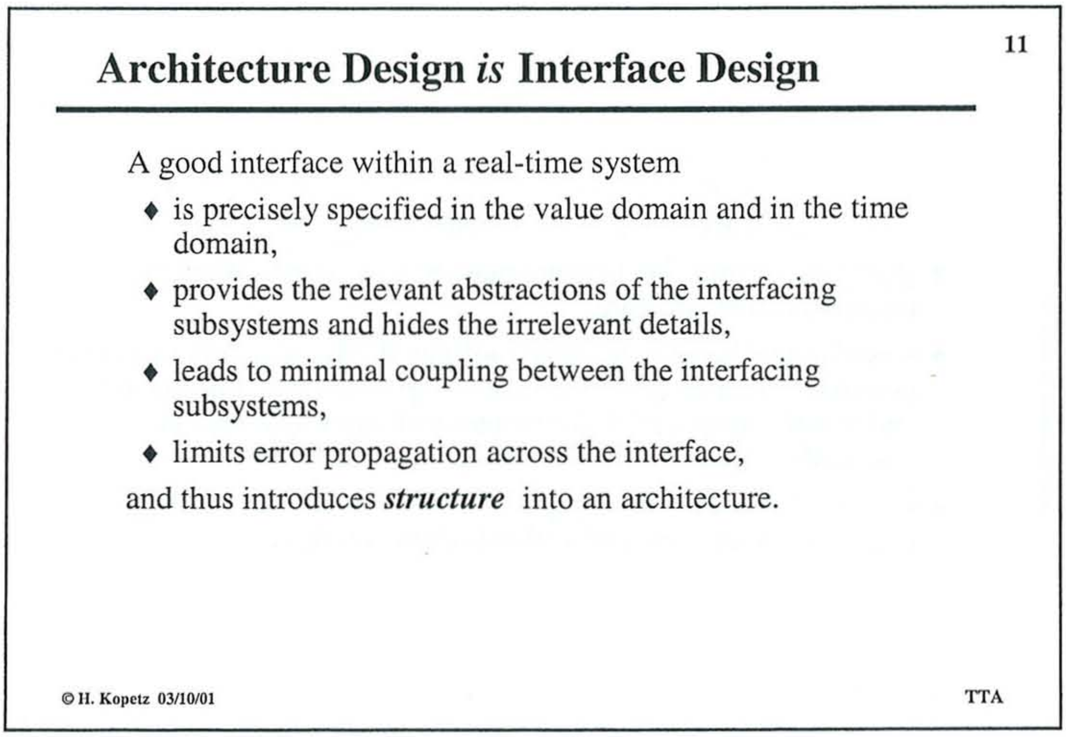
What is a “Single” Fault in the TTA?

- ♦ A Fault-containment region in the TTA is a single chip (System-On-a-Chip--SOC--software and hardware) which is at a physical distance from the other fault containment regions.
- ♦ Byzantine failures of chips are masked by a proper physical interconnection structure.
- ♦ It is claimed that in a properly configured TTA-star system, *every* possible failure mode of any *single chip (software or hardware)* and *nearly any* possible failure mode of any *single wire* is tolerated, without a loss of the timely service.
- ♦ Failures outside the fault-hypothesis (e.g., concurrent multiple chip failures) are detected with a high probability.



The TTA supports

- ♦ the provision of a global time base to all subsystems
- ♦ a predictable temporal behavior that can be analyzed *a priori.*,
- ♦ the partitioning of a large system into nearly autonomous composable subsystems by the introduction of stable interfaces.
- ♦ the independent development and validation of these subsystems, based on these precise interface specification,.
- ♦ the application transparent implementation of fault-tolerance by active redundancy.



11

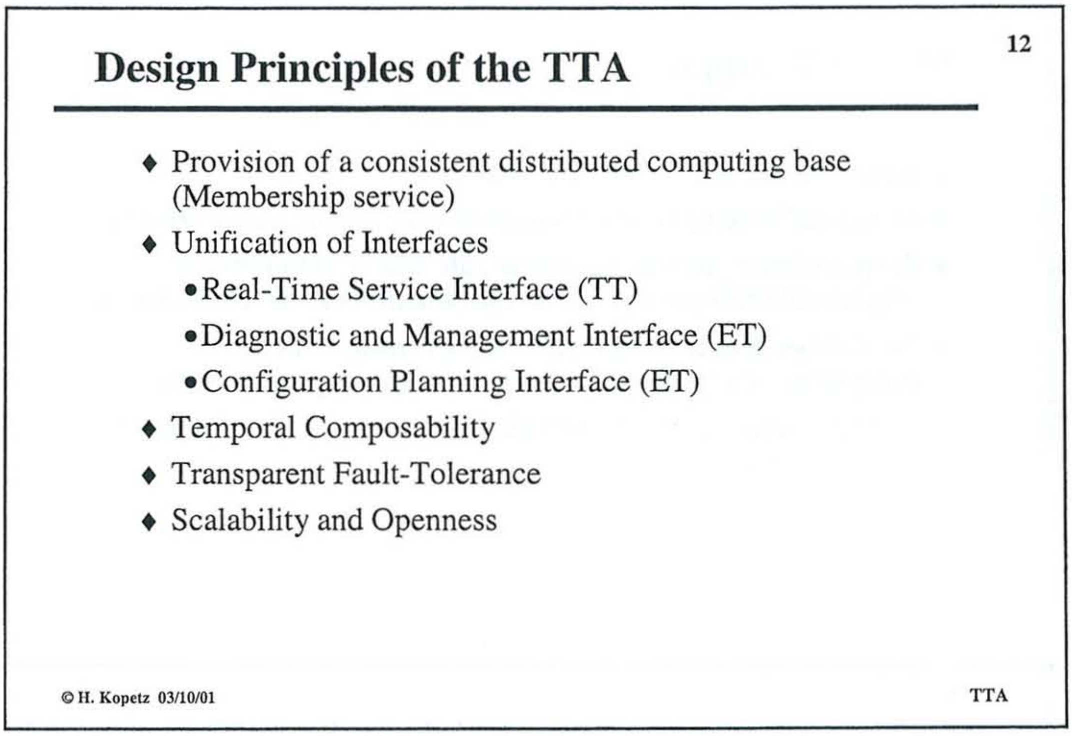
Architecture Design *is* Interface Design

A good interface within a real-time system

- is precisely specified in the value domain and in the time domain,
- provides the relevant abstractions of the interfacing subsystems and hides the irrelevant details,
- leads to minimal coupling between the interfacing subsystems,
- limits error propagation across the interface,

and thus introduces ***structure*** into an architecture.

© H. Kopetz 03/10/01 TTA



12

Design Principles of the TTA

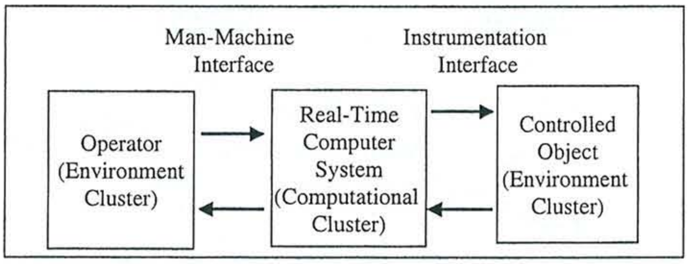
- Provision of a consistent distributed computing base (Membership service)
- Unification of Interfaces
  - Real-Time Service Interface (TT)
  - Diagnostic and Management Interface (ET)
  - Configuration Planning Interface (ET)
- Temporal Composability
- Transparent Fault-Tolerance
- Scalability and Openness

© H. Kopetz 03/10/01 TTA

Structure Overview

Real-time System: Computer System + Controlled Object + Operator
The controlled object determines the temporal requirements.

Cluster: A subsystem of the RT-system with high inner connectivity



Man-Machine Interface

Instrumentation Interface

Operator (Environment Cluster)

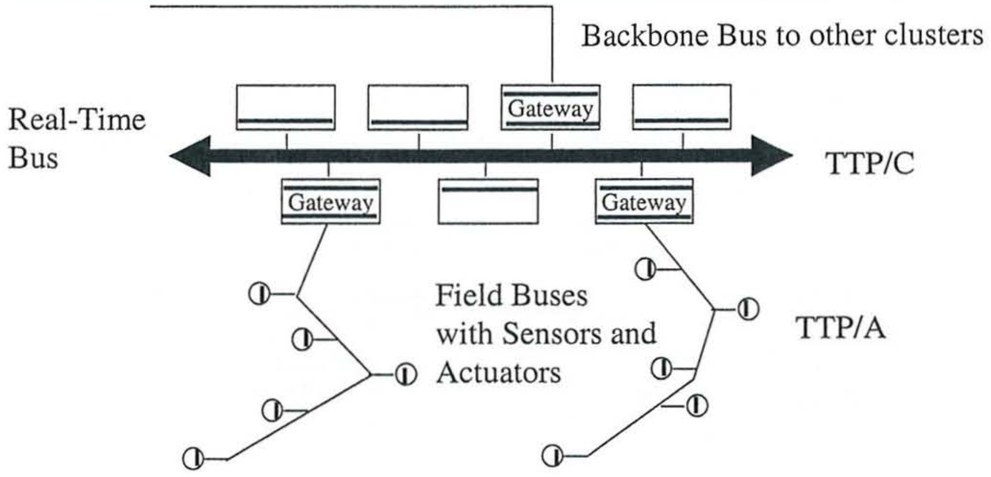
Real-Time Computer System (Computational Cluster)

Controlled Object (Environment Cluster)

Computation Cluster

- **Computational Cluster**: A subsystem consisting of a set of nodes interconnected by time-triggered communication networks.
- **Node**: A host computer and a communication-controller.
- **Gateway node**: A host computer with two communication controllers to two different clusters.
- **Transducer node**: A special gateway node with an interface to the controlled object in the environment.
- **Communication Systems**: Time-triggered communication systems (e.g., Fault-tolerant TTP/C system and sensor bus TTP/A)

Computational Cluster with TTP/C and TTP/A



Backbone Bus to other clusters

Real-Time Bus

Field Buses with Sensors and Actuators

—— Communication Network Interface (CNI), Temporal Firewalls

Communication Network Interface (CNI)

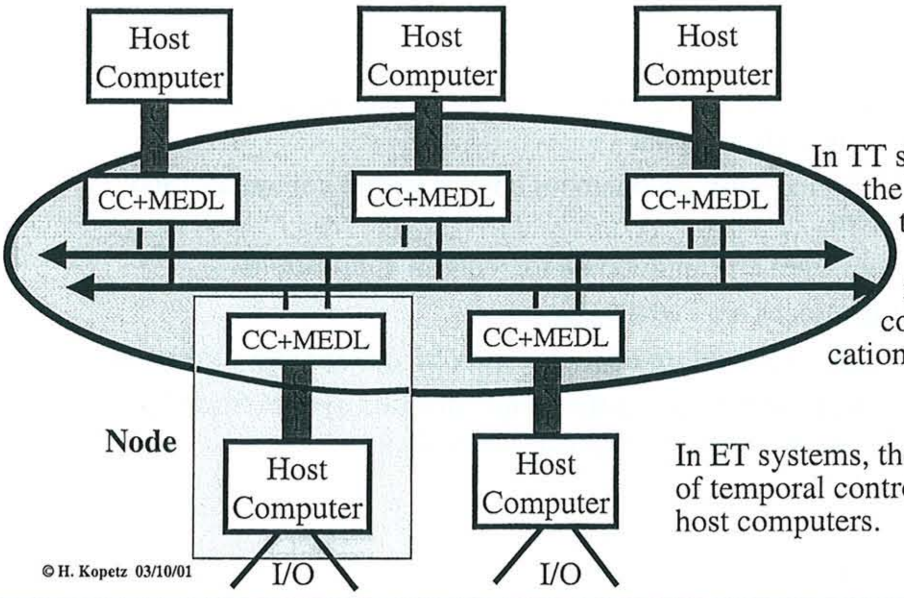
The Communication Network Interface (CNI) is the interface between the communication system and the host computer within a node. It implements a temporal input firewall and a temporal output firewall

- Is fully specified in the value domain and temporal domain
- Acts as an error containment interface
- Contains state information or event information.

The CNI is the most important interface of the time-triggered architecture.

Local versus Global Interactions

17



In TT systems, the locus of temporal control is in the communication system.

In ET systems, the locus of temporal control is in host computers.

The TTA distinguishes between the

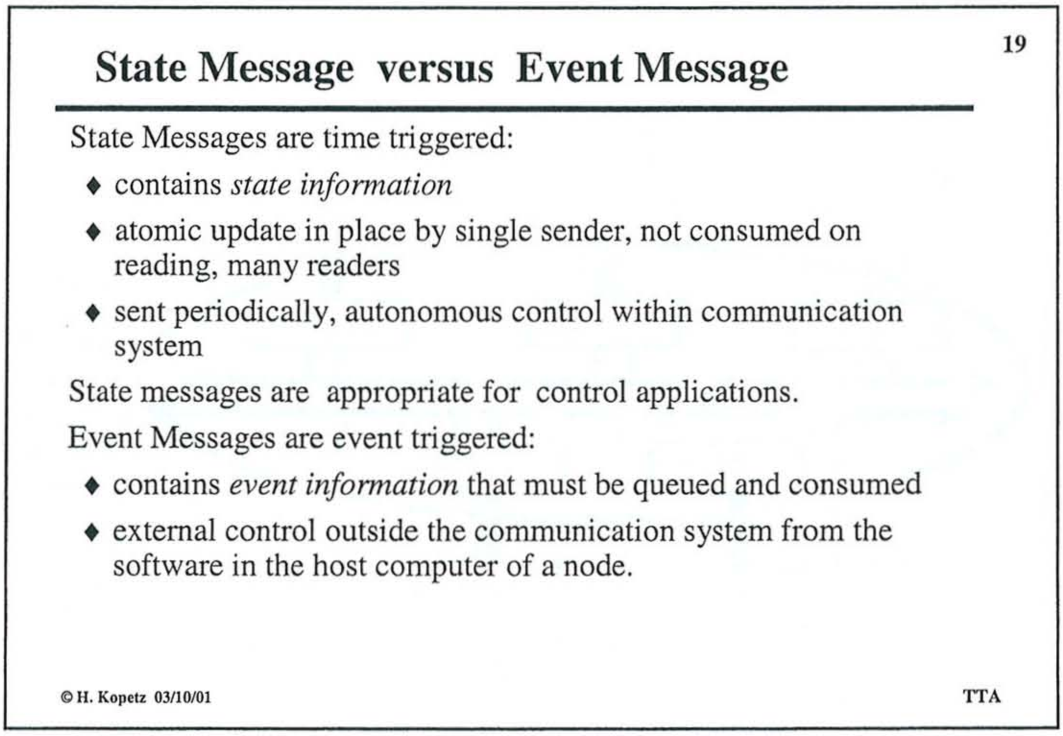
18

Distributed Computational Core (DCC):

- ♦ consists of the set of host computers connected by TTP/C
- ♦ operates on a sparse time base with agreed discrete input values
- ♦ supports the transparent implementation of fault-tolerance

Distributed Input/Output System (DIOS):

- ♦ connects the intelligent sensors and actuators to a node (or a fault-tolerant unit) by one or more TTP/A sensor busses.
- ♦ transforms the analog inputs into a consistent representation of discrete values on a sparse timebase.
- ♦ handles sensor and actuator fault-tolerance by providing the appropriate sensor agreement protocols.



State Message versus Event Message

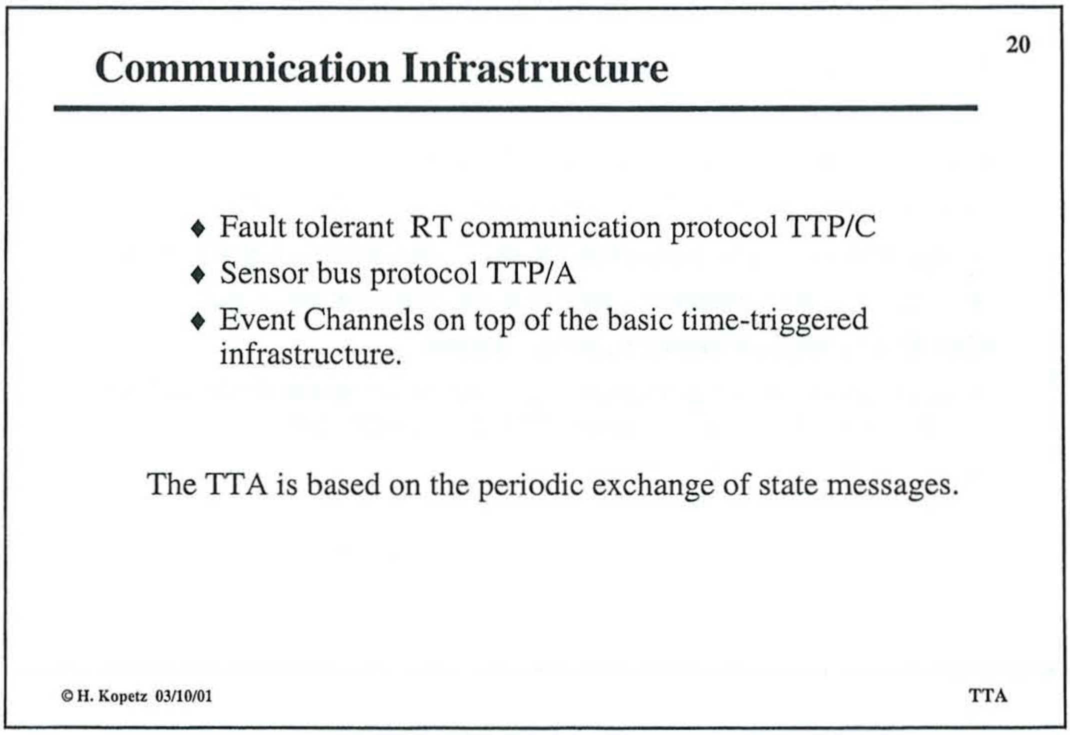
State Messages are time triggered:

- ♦ contains *state information*
- ♦ atomic update in place by single sender, not consumed on reading, many readers
- ♦ sent periodically, autonomous control within communication system

State messages are appropriate for control applications.

Event Messages are event triggered:

- ♦ contains *event information* that must be queued and consumed
- ♦ external control outside the communication system from the software in the host computer of a node.



Communication Infrastructure

- ♦ Fault tolerant RT communication protocol TTP/C
- ♦ Sensor bus protocol TTP/A
- ♦ Event Channels on top of the basic time-triggered infrastructure.

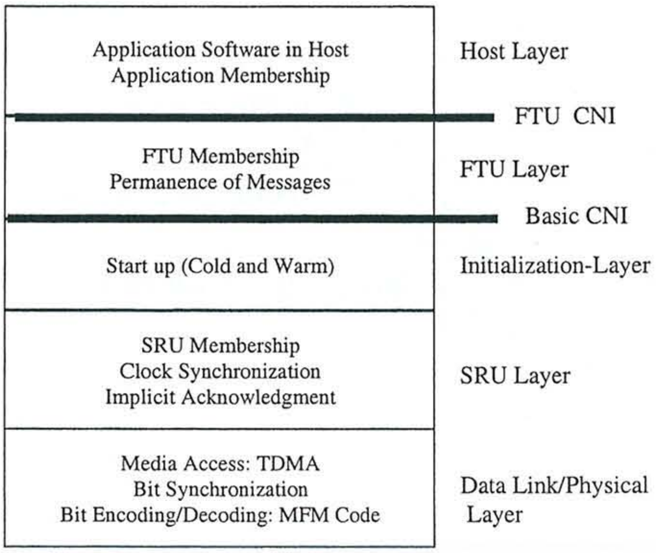
The TTA is based on the periodic exchange of state messages.

TTP/C Protocol Services

The Time-Triggered Protocol (TTP), connecting the nodes of the system, is at the core of the Time-Triggered Architecture. It provides the following services:

- Predictable communication with small latency an minimal jitter
- Fault-tolerant clock synchronisation
- Composability by full specification of the temporal properties of the interfaces.
- timely membership service (fast error detection)
- replica determinism
- replicated communication channels (support of fault- tolerance)
- good data efficiency

TTP Layers



Application Software in Host
Application Membership — Host Layer

FTU CNI

FTU Membership
Permanence of Messages — FTU Layer

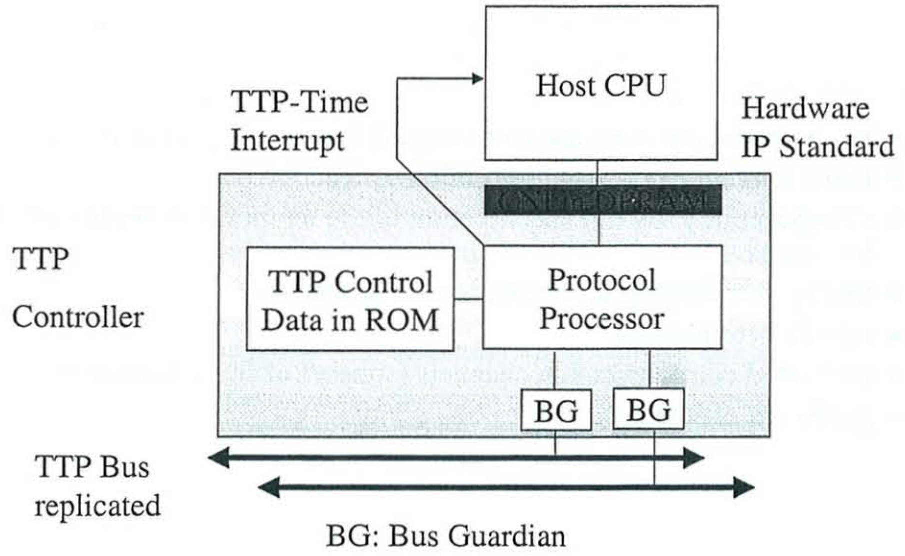
Basic CNI

Start up (Cold and Warm) — Initialization-Layer

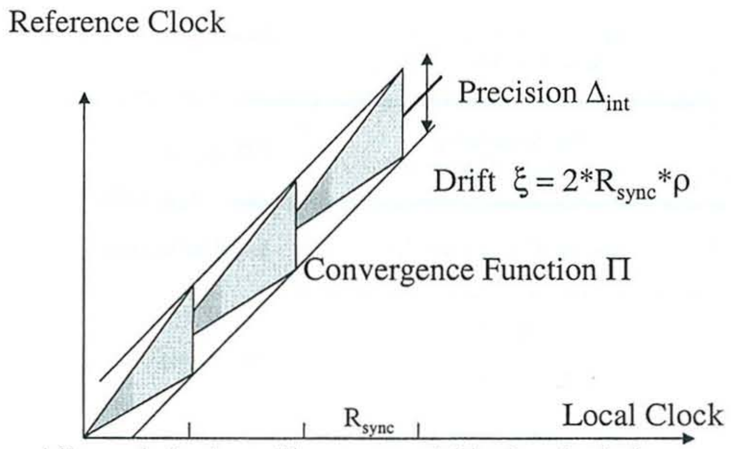
SRU Membership
Clock Synchronization
Implicit Acknowledgment — SRU Layer

Media Access: TDMA
Bit Synchronization
Bit Encoding/Decoding: MFM Code — Data Link/Physical Layer

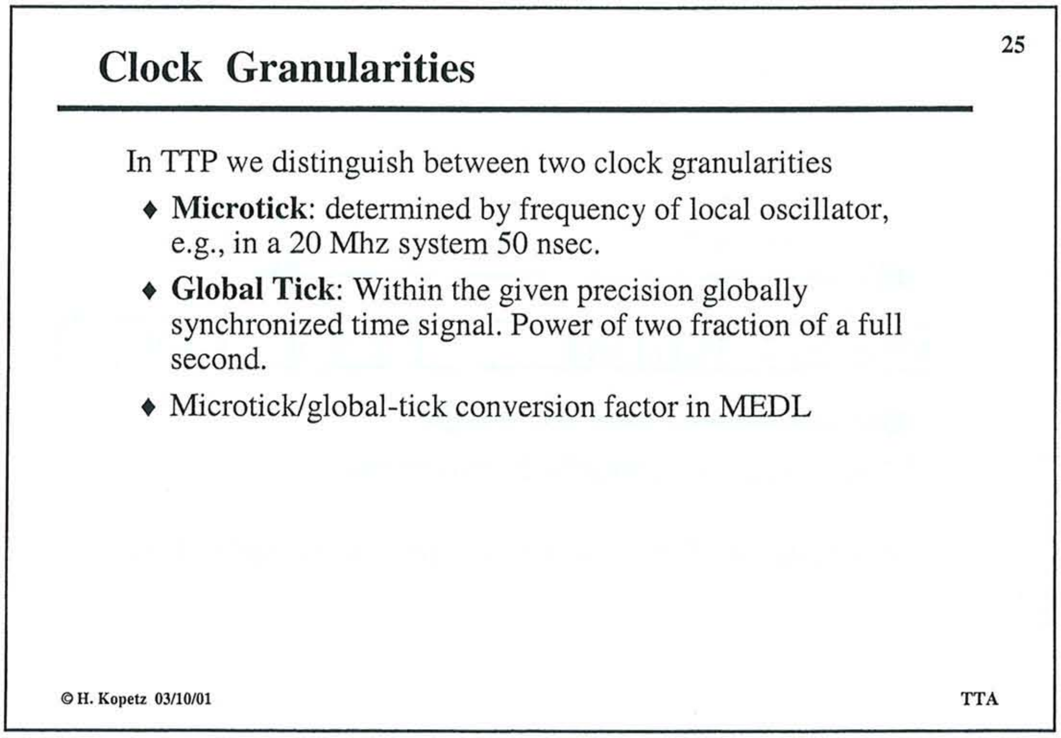
TTP-Controller



The Clock Synchronization Problem



All good clocks will operate within the shaded area



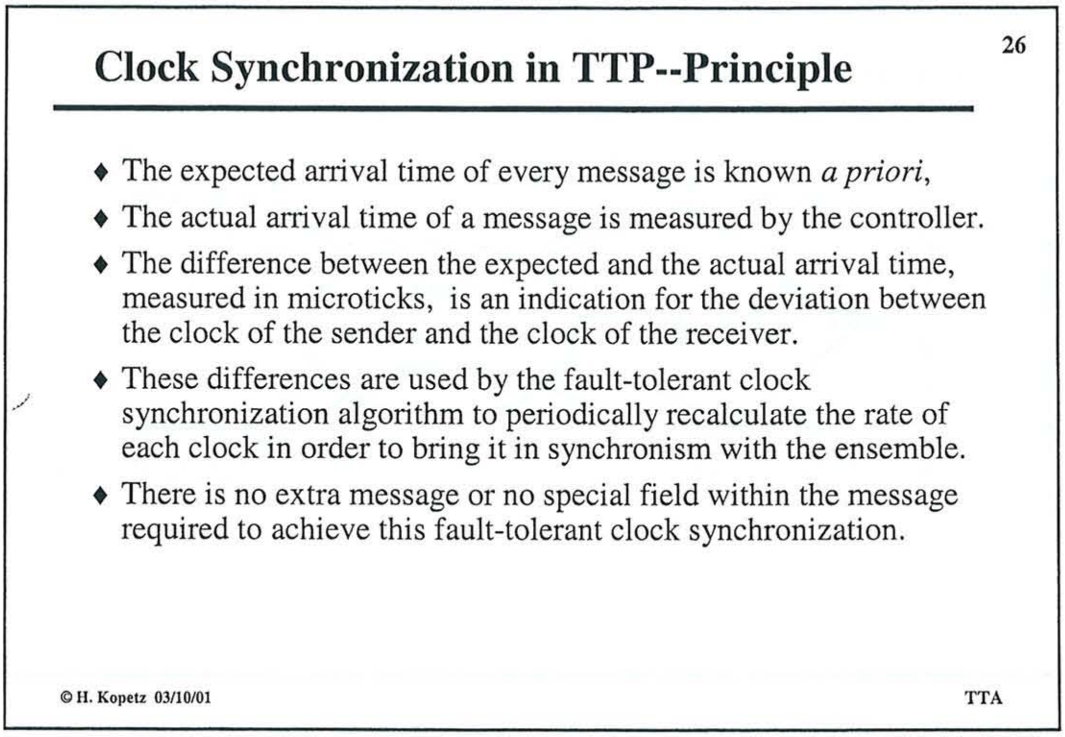
Clock Granularities

25

In TTP we distinguish between two clock granularities

- **Microtick**: determined by frequency of local oscillator, e.g., in a 20 Mhz system 50 nsec.
- **Global Tick**: Within the given precision globally synchronized time signal. Power of two fraction of a full second.
- Microtick/global-tick conversion factor in MEDL

© H. Kopetz 03/10/01 TTA



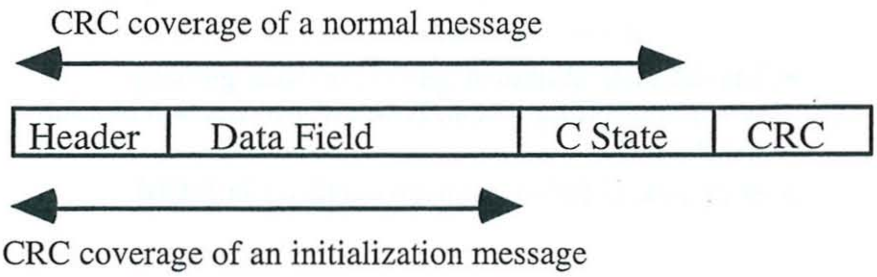
Clock Synchronization in TTP--Principle

26

- The expected arrival time of every message is known *a priori*,
- The actual arrival time of a message is measured by the controller.
- The difference between the expected and the actual arrival time, measured in microticks, is an indication for the deviation between the clock of the sender and the clock of the receiver.
- These differences are used by the fault-tolerant clock synchronization algorithm to periodically recalculate the rate of each clock in order to bring it in synchronism with the ensemble.
- There is no extra message or no special field within the message required to achieve this fault-tolerant clock synchronization.

© H. Kopetz 03/10/01 TTA

CRC Calculation in TTP



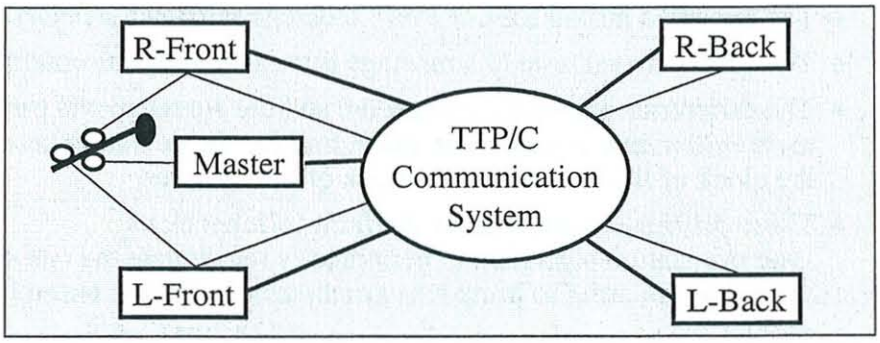
CRC coverage of a normal message

Header	Data Field	C State	CRC

CRC coverage of an initialization message

Controller (C)-State: Time, MEDL Position, Membership

Membership Service



Membership Service: Every node knows consistently (within a known small delay) who is present and who is absent.

Necessary for the detection of outgoing link failures.

Membership: ET versus TT

Every node must inform every other node about its local view of the "health state" of the other nodes--*and this in time*.

Event Triggered (e.g, CAN)

- **Membership difficult--no defined membership instants**
- Message arrival determined by the occurrence of events *unpredictable*
- Large Jitter
- No precise temporal specification of interfaces

Time Triggered (e.g., TTP)

- **Membership easy--membership instants defined**
- Message arrival determined by the progression of time *predictable*
- Minimal Jitter.
- Interfaces are temporal firewalls.

Membership Service--Principle

- A message will only be accepted by a receiver, if the receiver has the same C-state as the sender.
- If the successor of the sender receives at least one of the two messages of the sender correctly (correct CRC), then the successor will set the membership bit of the sender to TRUE.
- If the original sender cannot receive the messages of the successor (CRC error), it will repeat the CRC calculation with its membership bit set to FALSE.
- If this second CRC calculation is correct, the original sender knows that the receiver assumes the original sender has lost its membership.
- This conflict is resolved by the successor of the successor.
- The successor relationship is dynamic and depends on the current membership.

Guardian

A babbling idiot failure occurs if a node sends a message at the wrong time and disrupts the traffic on the complete bus. It is the most serious failure in a bus system.

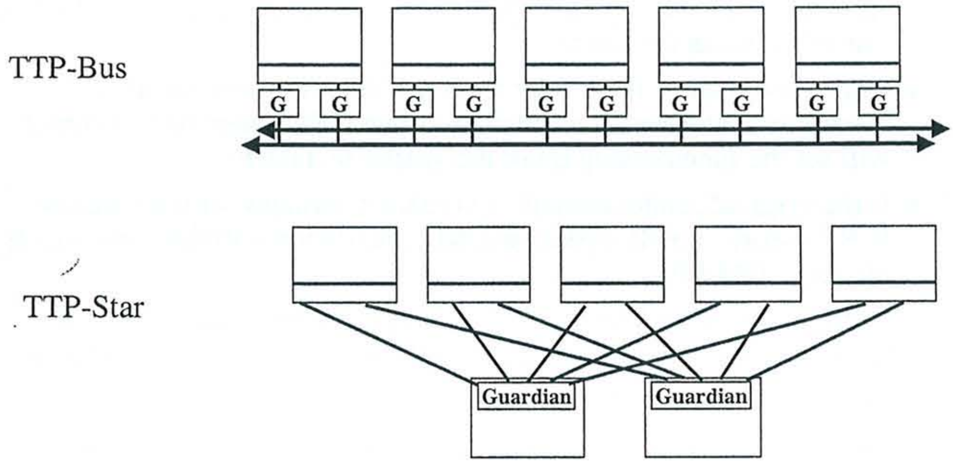
The purpose of the guardian is the elimination of *babbling idiot failures* of a node.

The guardian is a separate device with its own oscillator that monitors the temporal behavior of the TTP controller.

The Guardian allows access to the bus only during the TDMA slot owned by the node.

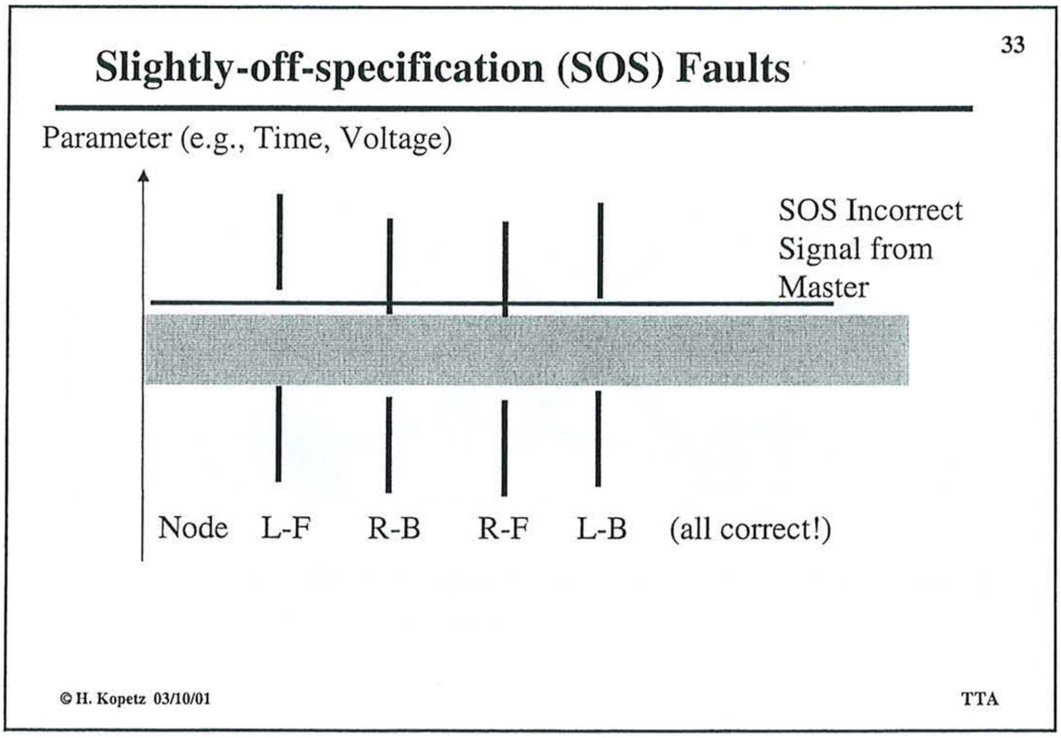
Fault injection experiments have shown that a guardian is necessary if a high error detection coverage must be achieved.

Physical Interconnection Structure



TTP-Bus

TTP-Star

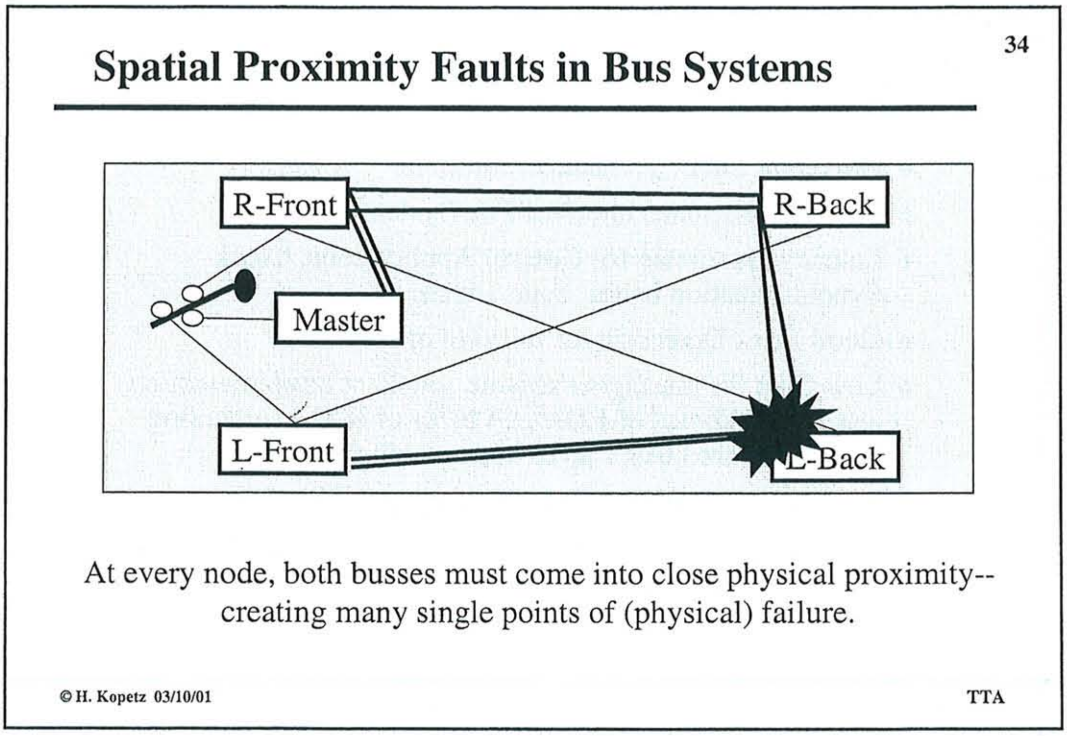


33

Slightly-off-specification (SOS) Faults

Parameter (e.g., Time, Voltage)
SOS Incorrect Signal from Master
Node
L-F
R-B
R-F
L-B
(all correct!)

© H. Kopetz 03/10/01 TTA



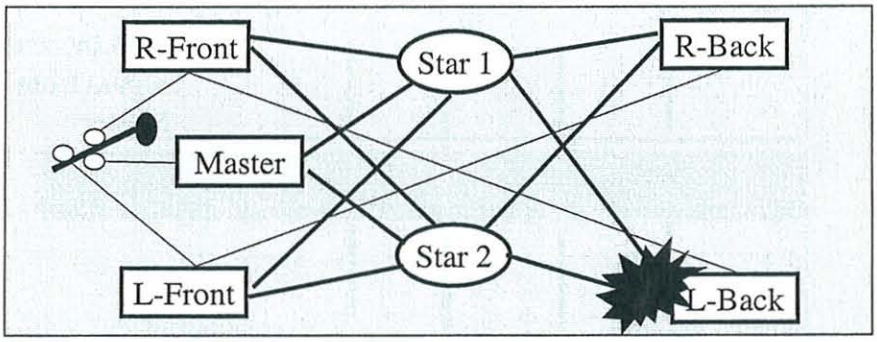
34

Spatial Proximity Faults in Bus Systems

At every node, both busses must come into close physical proximity--creating many single points of (physical) failure.

© H. Kopetz 03/10/01 TTA

Replicated Stars avoid Single Point of Failure



No defined volume of space becomes a single fault containment region, that can be a cause of total system failure.

TTP-A Protocol for Smart Transducers

- Universal Smart Transducer Interface
- Provides Standard Interface File System (IFS)
- Latency Guarantee for Control Applications, Clock Synchronization better than .1 msec
- Good Error Detection for fail safe operations
- Low Cost for intelligent sensors, smallest implementation less than 2 kbytes of ROM, 64 bytes of RAM (including IFS, software UART at 10 kbits on single wire)
- Fault tolerance at system level (duplicated buses)

In the process of standardisation by the OMG, the world's largest organisation for the creation of IT standards.

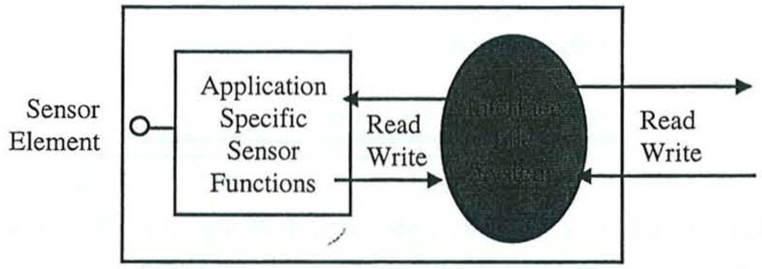
37

TTP/A--Principle of Operation

- ♦ Endpoint of the communication is a record in an Interface File System (IFS) located in the transducer node.
- ♦ Communication is organized into Rounds
  - A round is started by the active master that has knowledge of the global time
  - The first frame of a round is a fireworks frame, followed by data frames. The structure of a round is described in the round-descriptor list (RODL).
  - every round is independent of every other round
- ♦ The arrival of the fireworks frame is the global synchronization event starting a new epoch.

38

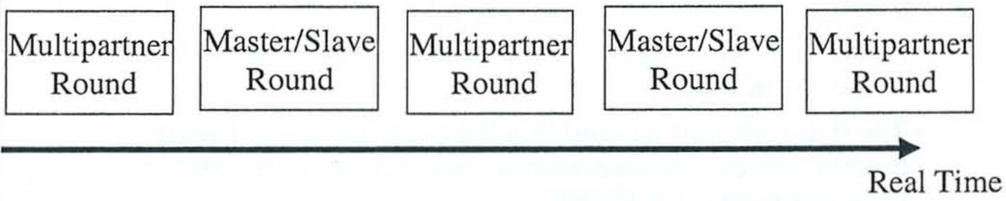
Interface File System



The Interface File System (ISF) encapsulates all information that is exchanged between a smart transducer and its environment. It provides a standardized structured name-space for information access

Interleaving of Rounds

Recommended Schedule:

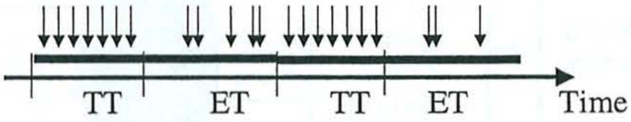


Master Slave Rounds have constant frame length.

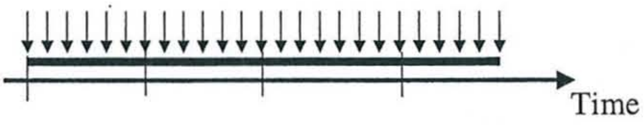
Master Slave Rounds may be empty, if no CM or CP service is requested by the master.

Integration of TT and ET Services

(i) **Parallel**: Time Axes is divided into two parallel windows, where one window is used for TT, the other for ET, Two media access protocols needed, one TT, the other ET



(ii) **Layered**: ET service is implemented on top of a TT protocol Single time triggered access media access protocol.



(iii) **Layered**: TT on ET

41

Event-triggered Traffic (CAN) on TTP/C

It is possible to implement event channels on top of the TTA.

Advantages:

- ♦ Precise temporal interface specification and composability (sparse time base) is maintained
- ♦ CAN controller interface for legacy software
- ♦ No feature interaction of parallel media access protocols

Disadvantage:

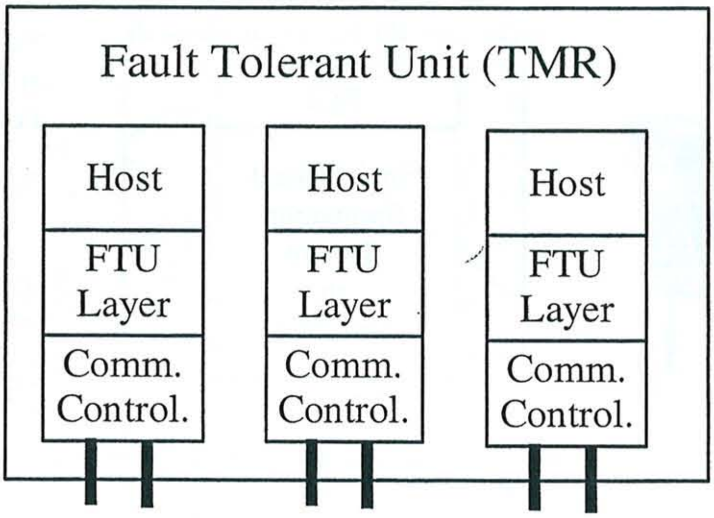
- ♦ Bandwidth sharing among nodes not supported

© H. Kopetz 03/10/01

TTA

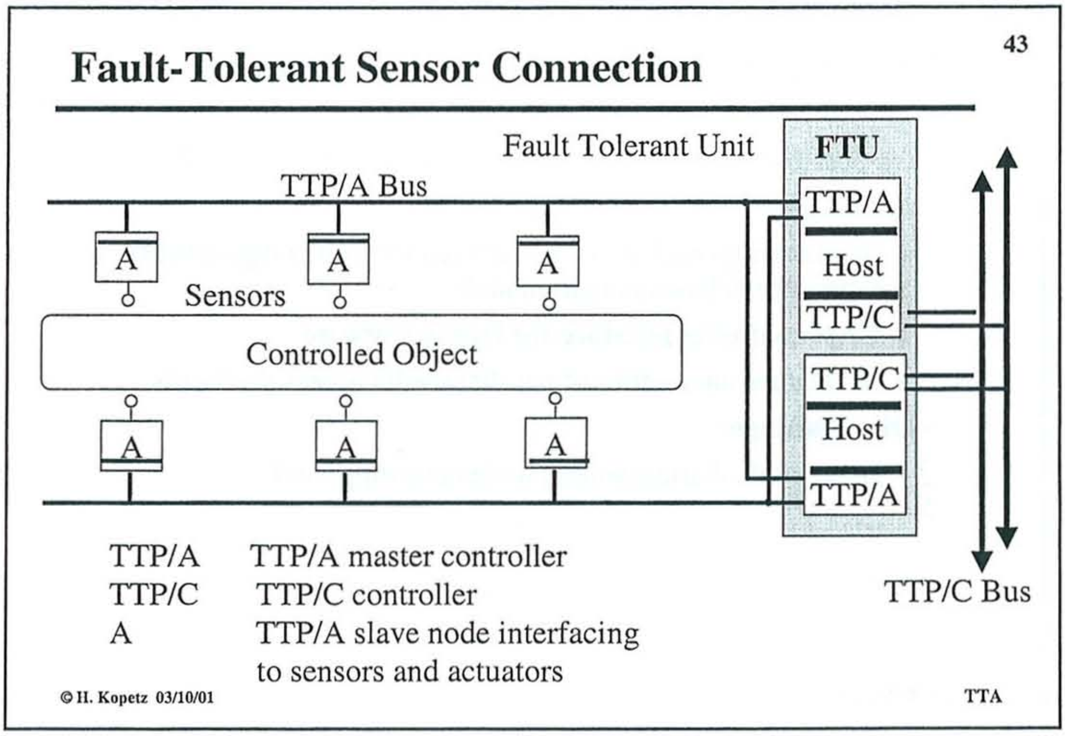
42

Replication in the TTA



© H. Kopetz 03/10/01

TTA



43

Fault-Tolerant Sensor Connection

Fault Tolerant Unit

FTU

TTP/A Bus

TTP/A

Host

TTP/C

TTP/C

Host

TTP/A

A

A

A

Sensors

Controlled Object

A

A

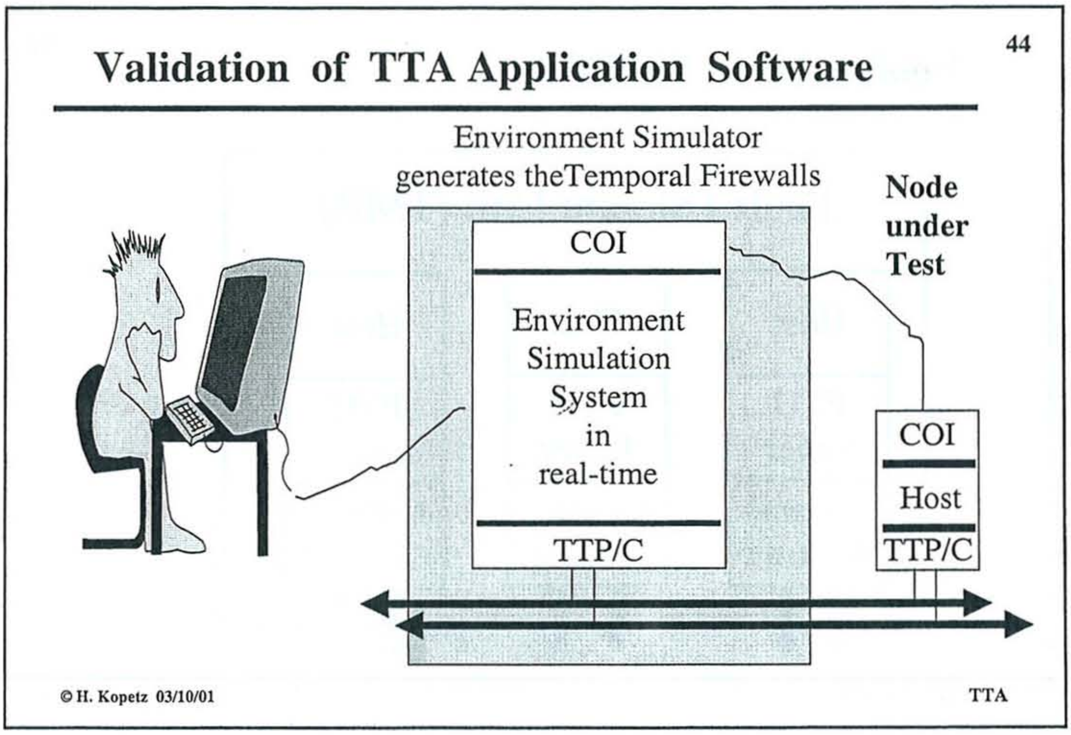
A

TTP/C Bus

TTP/A	TTP/A master controller
TTP/C	TTP/C controller
A	TTP/A slave node interfacing to sensors and actuators

© H. Kopetz 03/10/01

TTA



44

Validation of TTA Application Software

Environment Simulator generates theTemporal Firewalls

Node under Test

COI

Environment Simulation System in real-time

TTP/C

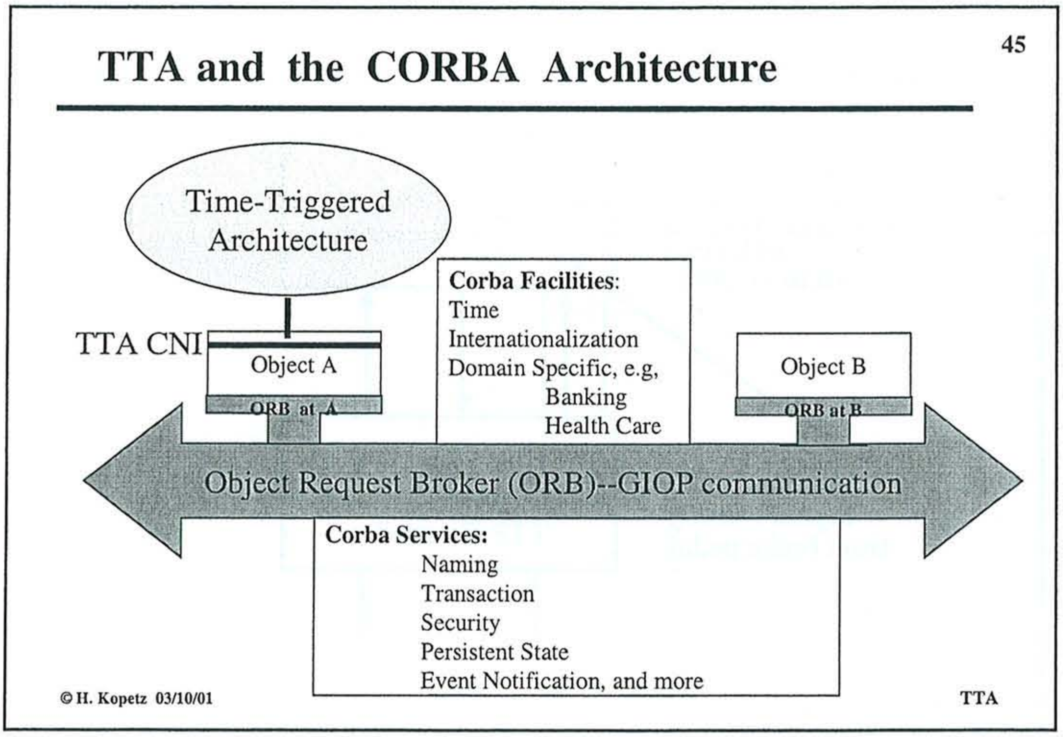
COI

Host

TTP/C

© H. Kopetz 03/10/01

TTA



45

TTA and the CORBA Architecture

Time-Triggered Architecture

TTA CNI

Object A

ORB at A

Corba Facilities:
Time
Internationalization
Domain Specific, e.g,
Banking
Health Care

Object B

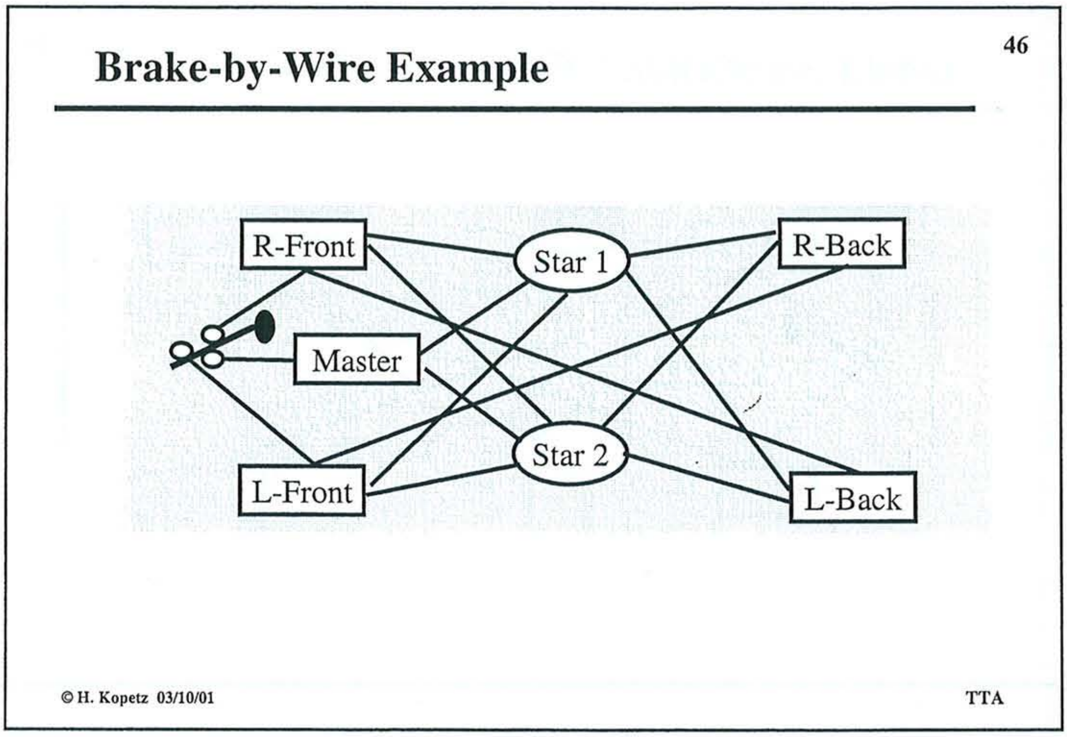
ORB at B

Object Request Broker (ORB)--GIOP communication

Corba Services:
Naming
Transaction
Security
Persistent State
Event Notification, and more

© H. Kopetz 03/10/01

TTA



46

Brake-by-Wire Example

R-Front

Star 1

R-Back

Master

Star 2

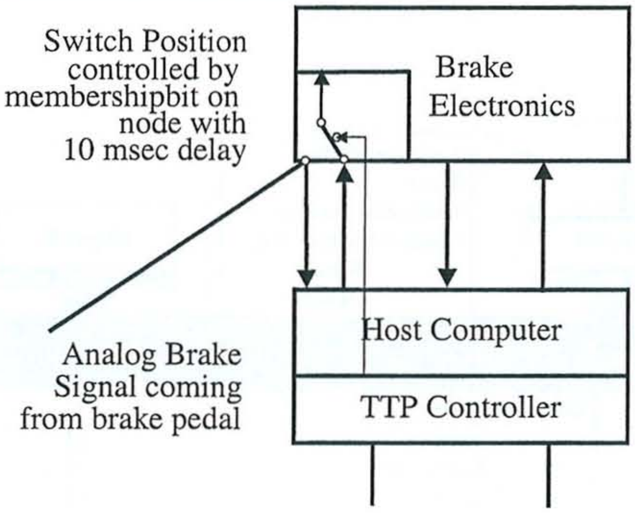
L-Front

L-Back

© H. Kopetz 03/10/01

TTA

Wheel Computer Interface



Switch Position controlled by membershipbit on node with 10 msec delay

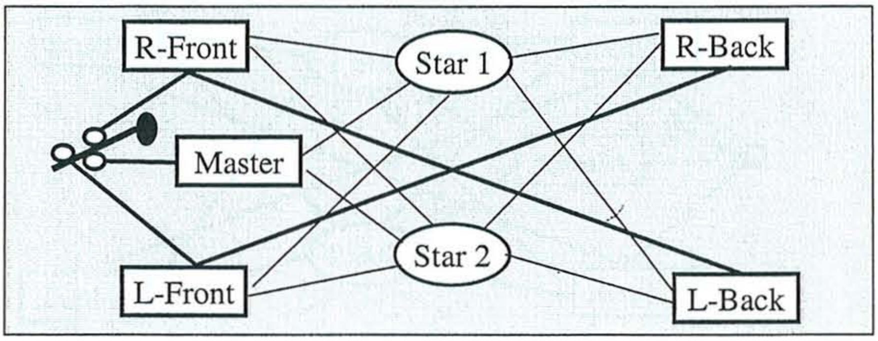
Brake Electronics

Host Computer

TTP Controller

Analog Brake Signal coming from brake pedal

Total Loss of Digital Communication



R-Front

Star 1

R-Back

Master

Star 2

L-Front

L-Back

The LULEA Car

49

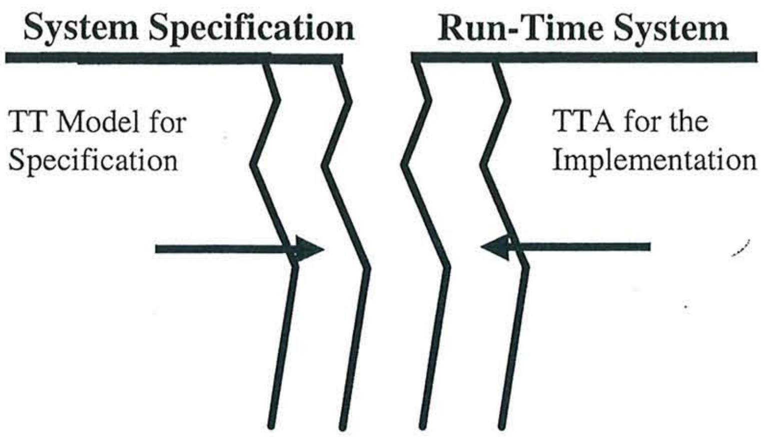
QuickTime™ and a
Photo - JPEG decompressor
are needed to see this picture.

© H. Kopetz 03/10/01

TTA

Closing the Implementation Gap

50



System Specification

Run-Time System

TT Model for
Specification

TTA for the
Implementation

© H. Kopetz 03/10/01

TTA

DISCUSSION

Rapporteur: A I Kistijantoro

Lecture Two

Professor Randell speculated on what will be put on a single chip in the future. He wondered that it will be so huge and complex, so that if TTA still ignores the problems that happened inside chip, it will be missing the boat. Professor Kopetz replied that he believes that in 10-20 years, it is possible to build a complete cluster in a single die.

A participant asked about the existence of several architectures in the market and the requirements for developing components that can be integrated to TTA. Professor Kopetz replied that the issue of how many and which architecture will survive in the market does not only depend on technical aspect, but also on non technical aspect. He believes that in the future there will be only one or two architectures that survive. Regarding to the requirements for the component in TTA, he said that the component should satisfy all the properties of the TTA or one can build an interface that can reconcile all properties mismatches.

Professor Jones asked the definition of decomposition. Professor Kopetz replied that each component has a specification at the interfaces. This specification is only part of the full specification of the component. Any component that will be integrated into a certain architecture must satisfy all requirements needed at the interfaces.

Professor Kim argued whether it is really necessary to apply very fine grain, high priority, global scheduling, as only specialized applications may require this kind of scheduling. Professor Kopetz replied that it depends on the application. For the applications in hard real time domain, every message has to be understood in term of its timing, but for tasks that are not time critical, then we don't have to send the message with a very fine grain scheduling.