Chapter 2
Distributed Real-Time Systems

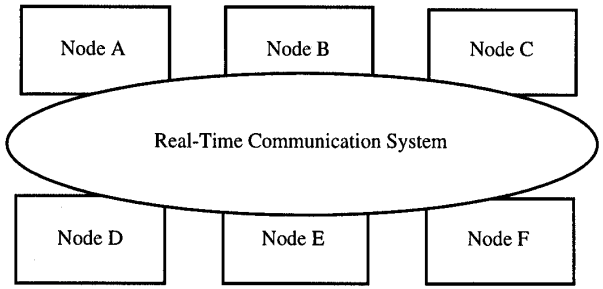
Overview

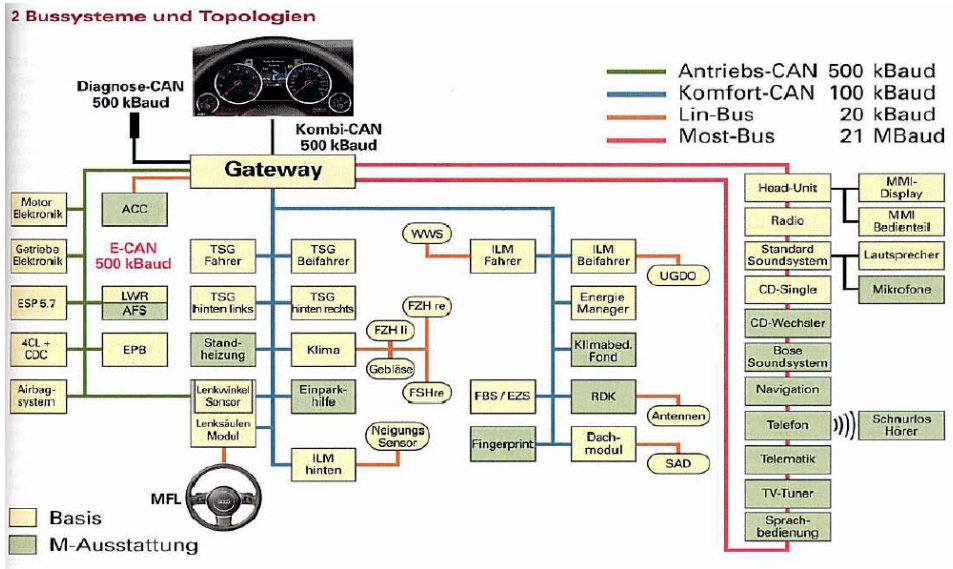
- From functional point of view there is no difference between centralized architecture and distributed architecture
- Overview of distributed real-time system architecture with a set of nodes and communication network interconnecting these nodes
- Introduction to concept of event messages and state messages
- Composable architecture as preferred alternative for hard real-time systems
- The communication network interface between host computer in node and communication network
- Implications on scalability
- Implications on dependability

2.1 System Architecture

In a distributed system, it is feasible to encapsulate the logical function and associated computer hardware into s single unit, a node.
All essential functional and temporal properties are hidden behind a simple and stable node interface.
The nodes in a distributed architecture are less complex than they would be in a centralized architecture. Functional failure can be better pinpointed to failure of a single node.

A distributed application can be decomposed into an operator cluster, a computational cluster, and a controlled object cluster. In case the computational cluster is implemented as a distributed system, the system looks like shown below.

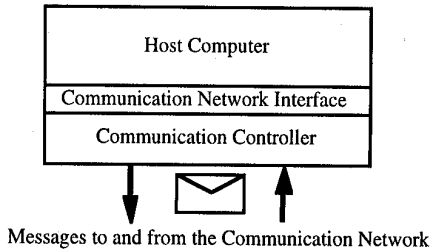




Hardware structure

A node can be portioned into two subsystems:

- host computer
- local communication controller



The communication-network interface (CNI) is located at the transport level of the OSI reference model.
The set of all communication controllers, along with the physical interconnections, form the real-time communication system of the cluster.

Communication-network interface

Purpose of the real-time communication system: Transport messages from the CNI of the sender node to the CNI of the receiver node

- within predictable time
- with a small latency jitter
- with high reliability

Data semantics

Two types of messages at the CNI: event messages and state messages.

Every event is significant; event messages must not be lost (loss of synchronization).
Event messages must be queued at receiver and be removed upon reading.
Order in queue must correspond to temporal order of event occurrence.

State messages (e.g. current temperature) may overwrite old state messages. State values may be read more than once.

In real-time systems, state semantics is needed more frequently than event semantics.

Control strategy

When must a message be sent? Decision either within sphere of control of host computer (external control) or within sphere of control of communication system (autonomous control).
Most common used strategy is external control.

With external control, a “send” command issued by the host computer to the CNI initiates transmission of message. When a message arrives at receiver, a control signal (interrupt) from the communication system crosses the CNI and unblocks a “receive” command.

With autonomous control, the communication system decides autonomously when to send the next message, and when to deliver the message at the CNI of the receiver. Autonomous control is normally time-triggered. The communication system contains a transmission schedule.

Event message

Event message combines event semantics with external control. Event messages require one-to-one synchronization between sender and receiver.

State message

State message combines state-value semantics with autonomous control. State messages can be considered similar to global variables, except for

- The communication system guarantees atomicity of a message write operation
- There is only a single sender (writer) process

There is no need for one-to-one synchronization; loose coupling between nodes.

Communication system

A number of design alternatives are available:

- single channel system, such as bus or ring
- multiple channel system, such as a mesh network

Increase communication reliability by message retransmission in case of failure, or by always replicating messages, or replicating message channels.

Communication system is a critical resource of a distributed system. Reliability should be an order of magnitude higher than the reliability of individual nodes.

Gateways

A gateway exchange relative views between two interacting clusters. A gateway node has either

- an instrumentation interface and a communication interface (interface node)

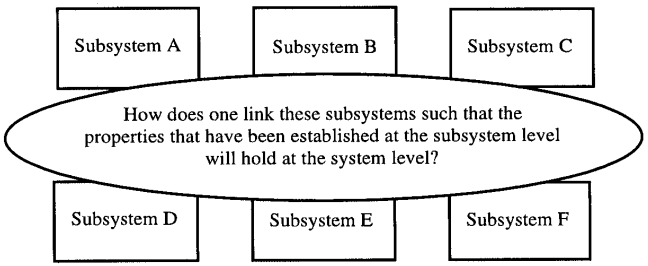
or

- two communication interfaces, each interfacing to one of the interacting clusters

The gateway host transforms data formats of one cluster to those expected by the other cluster. Gateways are often used to hide internal features of legacy systems.

2.2 Composability

In an architecture supporting composability it is ensured that properties of a subsystem are maintained during system integration. Thus, large systems can be built by composition of a number of smaller, well tested subsystems.



An architecture is said to be composable with respect to a specified property if the system integration will not invalidate this property once the property has been established at the subsystem level (e.g. timeliness or testability).

In a composable architecture, system properties follow from subsystem properties.

Event triggered communication systems

In an event-triggered (ET) communication system temporal control is external to the communication system.
Access conflicts can occur when several host computers decide to send a message simultaneously. The probability of access conflicts increases with the number of nodes and messages on the network.
Temporal control in an ET system is a global issue, depending on the behaviour of all nodes in the distributed system. From the point of view of temporal control ET systems are thus not composable. Example: CAN bus.

Time-triggered communication systems

In a time-triggered communication system, temporal control resides within the communications system, and is independent from the application software of the host computers.

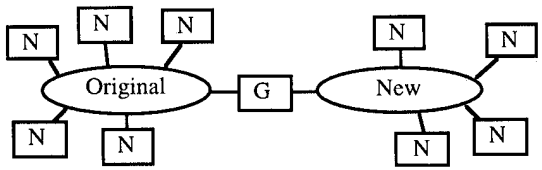
State messages are transported at predetermined points in time from the sender CNI to the receiver CNI.

The CNI acts thus as a temporal firewall, isolating the temporal behaviour of the host computer from the temporal behaviour of the network.

System integration will not change the temporal properties of the CNI. A TT architecture is thus composable with respect to communication timeliness.

2.3 Scalability

Most automotive systems evolve over an extended period of time (years or decades). Evolving requirements are thus the rule. Functions are being modified, and new functions are being added. A scalable architecture is open to such changes, and does not impose an upper limit on the capabilities of the system.



Extensability

Means:

- nodes can be added within the given capacity of the communication channel, to add processing power to the system
- new clusters can be added using gateway nodes when a cluster has reached it communication capacity limit, or the processing power limit

without changing the original nodes and networks.

Complexity

Complexity of a system relates to the number of parts, and the number of types of interactions among the parts.

In a scalable architecture, the effort to understand a single function remains constant and does not grow with the size of the system.

Complexity can be reduced by encapsulating functionality behind simple and stable interfaces.
The partitioning of a system into subsystems, the encapsulation of the subsystems, the preservation of the abstractions in case of faults, and strict control over the interaction patterns among the subsystems are the key mechanism for controlling complexity.

Thus, time-triggered architectures can handle complexity better than event-triggered architectures.

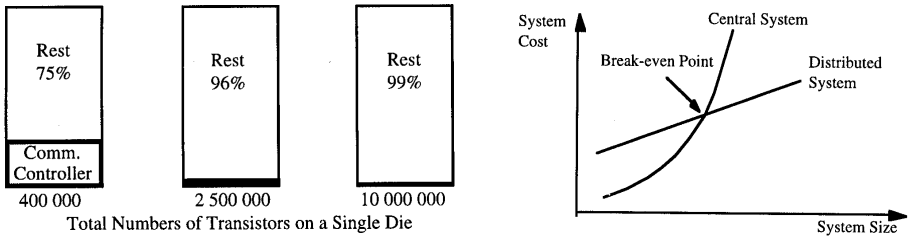
Silicon cost

The die size of an integrated circuit determines its cost. The cost of a die is approximately:

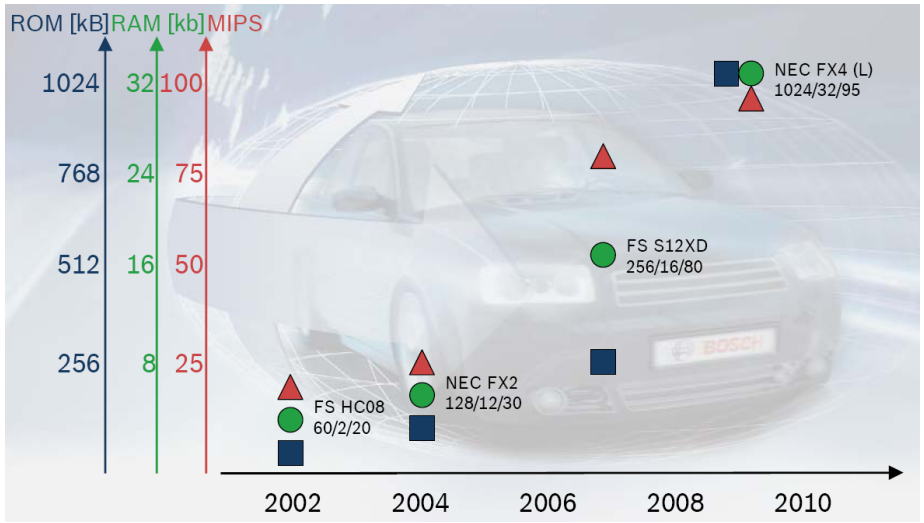
$$\text{Cost} = K \cdot (\text{die area})^3$$

The die area required for the communication controller is increasingly unimportant with regard to total cost. The cost of a distributed architecture starts out higher, but increases slower with system size than that of a centralized architecture.

However, there are many tradeoffs determining the system cost, and silicon cost is becoming an increasingly less important factor here.



Example: requirements for body computer microcontrollers:



2.4 Dependability

Implementing a dependable system requires effective fault containment, error containment, and fault tolerance.

Error-containment region

Fault-tolerant system must be structured into partitions that act as error-containment regions.

This way, errors in one partition can be detected and corrected or masked before consequences corrupt the rest of system.

Error containment region must be specified as a service, delivered across a small interface to the outside world. Error of the service must be detectable at the interface.

Error-containment coverage: probability that an error in the containment region is detected at the interface.

Error-containment regions can be introduced at level of functional hardware block, level of task. In a distributed system, a node can constitute an error-containment region.

Replication

Node failures can be masked by providing actively replicated nodes. The replicas must show deterministic behaviour. It requires careful hardware/software design to obtain replica determinism, i.e. replicated nodes “visit” the same states at about the same time. This precludes asynchronous preemptive scheduling.

Certification support

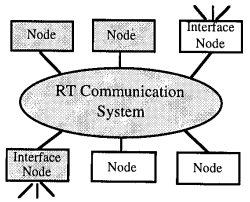
Often design of safety-critical systems must be approved by independent certification agency. Certification agency bases its assessment of the system on the analysis of the safety case presented by the designer.

A safety case is the accumulation of credible analytic and experimental evidence that convinces the certification agency that the system is safe to deploy (see ISO 26262).

Faults are categorized (here from the aircraft industry):

- Catastrophic: fault can be cause of an accident
- Hazardous: fault reduces safety margin of the redundant system, operation is critical
- Major: fault reduces safety margin such that immediate maintenance is required
- Minor: fault has only a small effect on the safety margin. Repair at next scheduled maintenance
- No Effect: fault has no effect on the safety margin

Key concern is remainder of safety margin after occurrence of a primary fault.



The consequence of a fault is an error.

2.4 Dependability

Errors in non-safety critical subsystems must be detected before they can propagate into a safety critical subsystem.

Architecture must rule out by design unintended interactions among subsystems of different criticality.

In a centralized architecture, it is very difficult to demonstrate error-containment.

2.5 Physical Installation

2.5 Physical Installation

There are a number of reasons for physical integration of a small microcontroller into a mechanical subsystem for compact mechatronical components:

- Less cabling: Electromechanical parts can interface directly with microcontroller, without error-prone and expensive cabling and connections
- Components become self-contained and achieve higher degree of autonomy. Functions can be performed and tested without the need to interconnect the component to a distant control system.
- The communication to and from this component can be serialized and accomplished by a single wire or twisted-pair field bus, simplifying installation of the component.

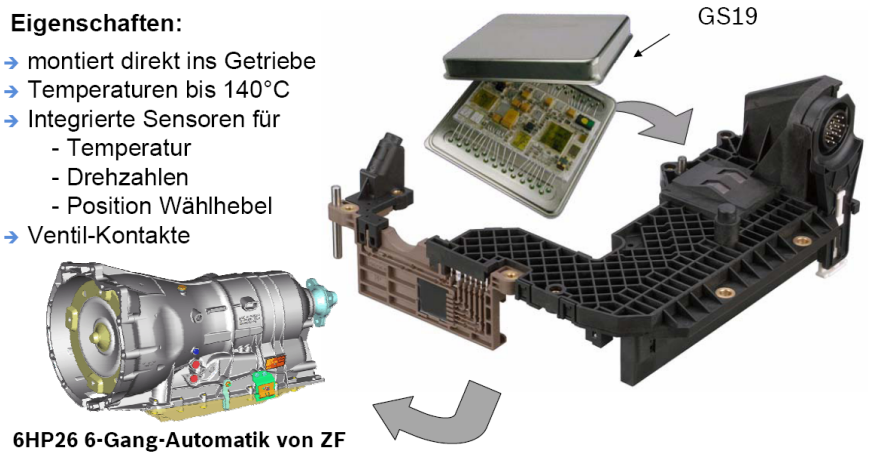
Example: gear control box from Bosch/ZF

2.5 Physical Installation

2.5 Physical Installation

Eigenschaften:

- montiert direkt ins Getriebe
- Temperaturen bis 140°C
- Integrierte Sensoren für
  - Temperatur
  - Drehzahlen
  - Position Wählhebel
- Ventil-Kontakte



6HP26 6-Gang-Automatik von ZF

Points to Remember

Points to Remember

How can you make the various functionality composable?

What is the CNI?

If you want to make the system fault tolerant?

Fault hypothesis required?

Error containment – how?

Temporal Requirement – MSC (message sequence chart)

Functional Requirements?

State messages or Event messages?

Include a new node in the system – myEcu

All button press should be routed through my ecu

And the temporal requirement should not be affected.