



FlexRay und AUTOSAR

Stephan Reichelt, Dr. Karsten Schmidt, Frank Gesele,
Nils Seidler, Prof. Dr. Wolfram Hardt

Inhalt


- ▶ Herausforderungen
- ▶ Systemkomplexität
- ▶ Vernetzungskomplexität
- ▶ Fazit



Herausforderungen

Technologietreiber

FlexRay™ und AUTOSAR

- standardisiert anno 2000 + x
- keine Felderfahrten
- nicht kundenrelevant
- vorerst: erhöhte Entwicklungskosten
- Controller Area Network 1987 standardisiert
-  in 2006: 926.180 Fzg. ohne FlexRay / AUTOSAR



Modellvielfalt



vernetzte Funktionsvielfalt



➤ bis zu 40 Teilnehmer Komfort / Infotainment

+ Kombi + FAS + Chassis + Powertrain + Gateway

Kommunikationslast

	2007	2009	per ECU
Komfort		+175%	+85%
Antrieb		+ 35%	konst
Anzeige		konst	+10%

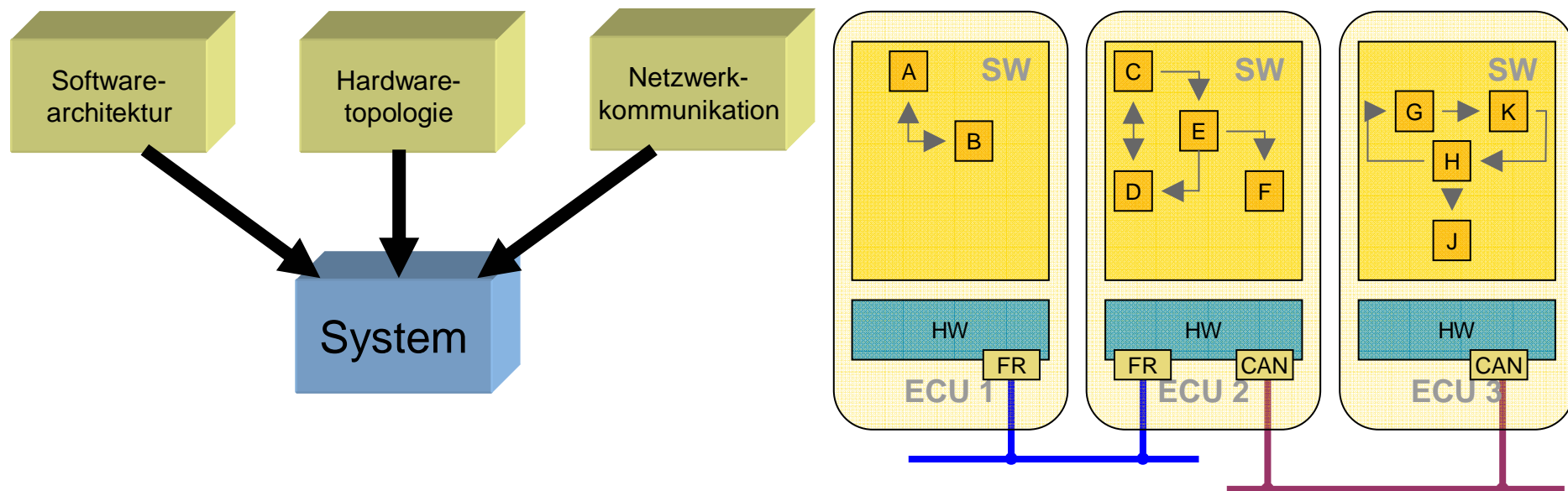
# Knoten	~ 70	>100
# Busse	+ 3	



Systementwurf und –integration

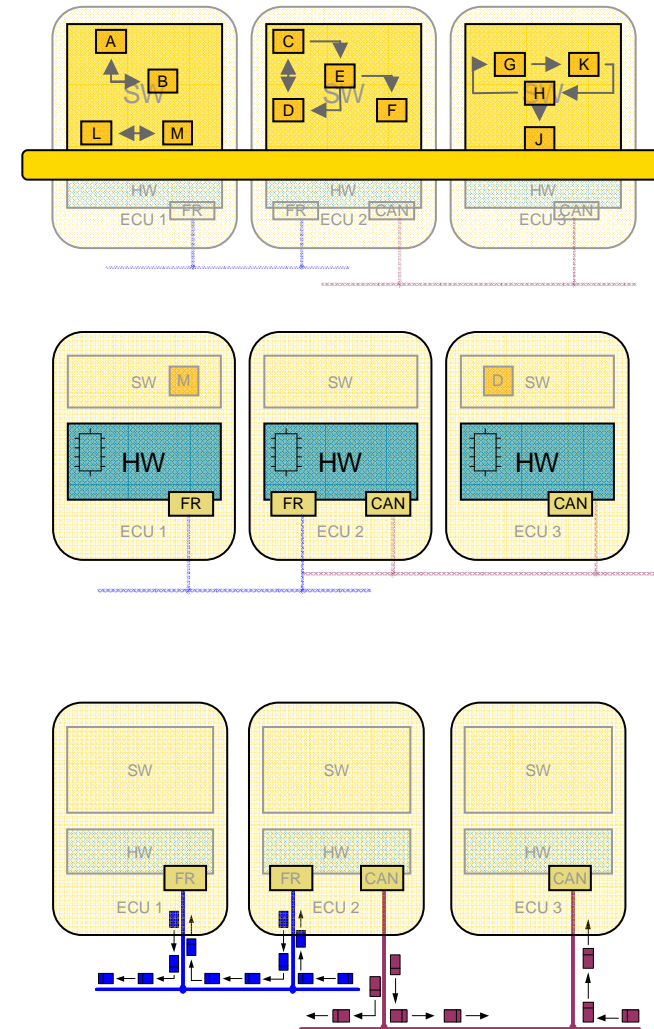
Systemkomplexität

Separation of concerns als Lösungskonzept

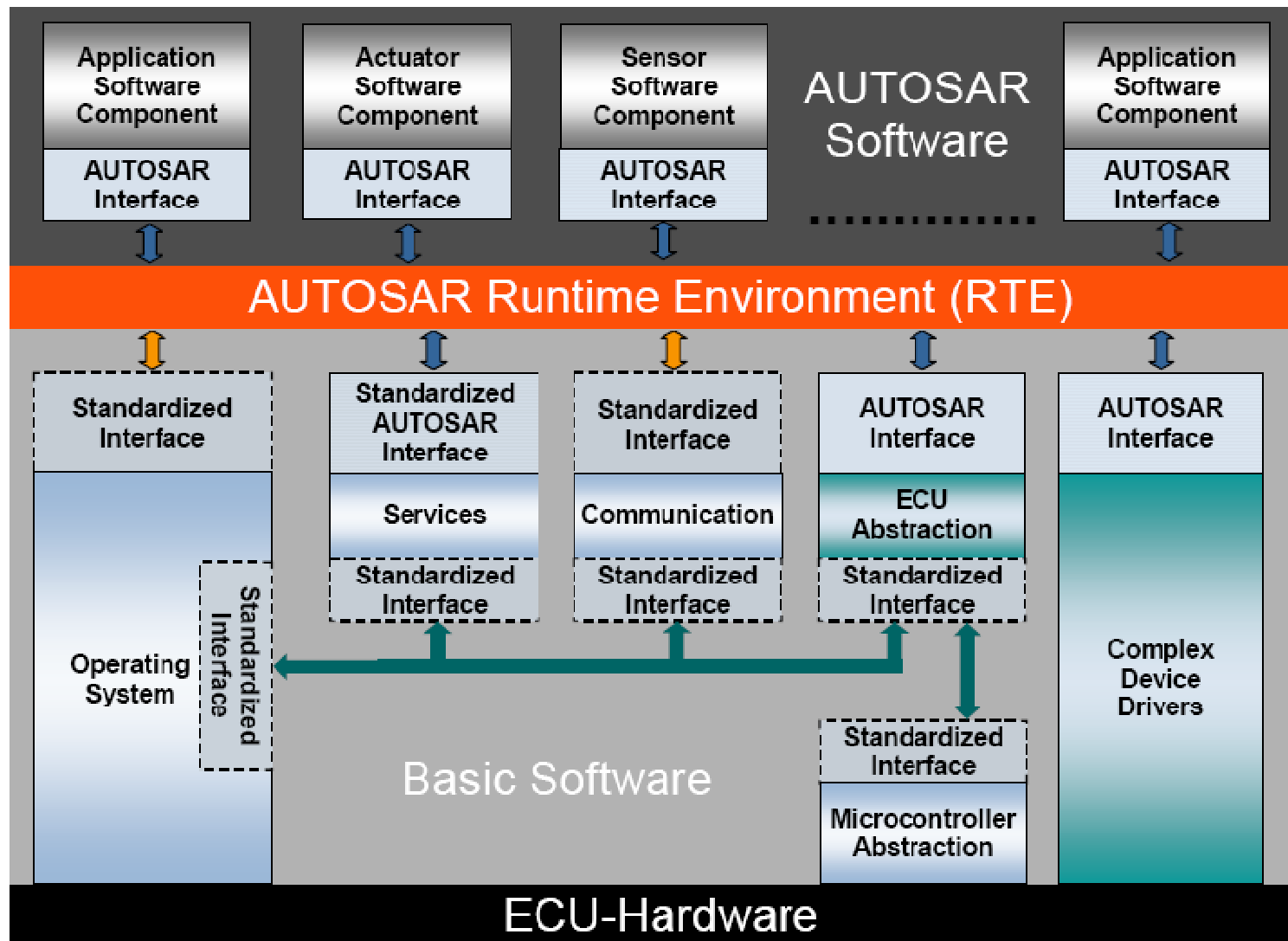


AUTOSAR: Ein Überblick

- **Softwarearchitektur**
 - Kommunikationsabstraktion
 - Verschiebbarkeit von Funktionen
- **Hardwaretopologie**
 - HW-Ressourcen
 - Kommunikationscontroller
 - fixe mappings (z.B. Sensorik)
- **Netzwerkkommunikation**
 - Kommunikationsmatrix
 - Frametypen

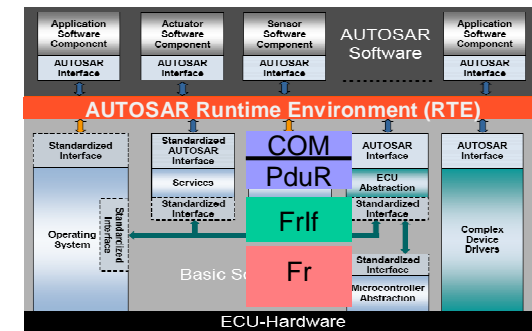


AUTOSAR: Ein Überblick



Kommunikationsstack

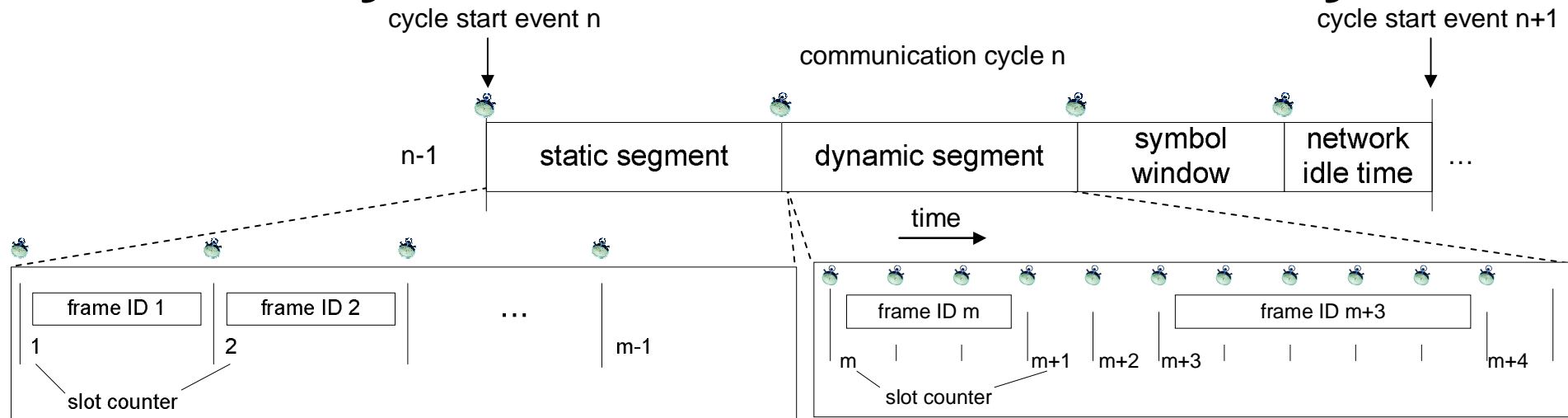
- physikalischer Buszugriff, HW-Treiber
- Busparameter: Zykluszeit, Segment-timing, ...
- Routing-Unterstützung
- Mapping: Netzwerk- auf Applikationssignale
- Abstraktion
 - interne vs. externe Kommunikation
 - Puffermodi



Vernetzungskomplexität

Integrationsaufwand beherrschen

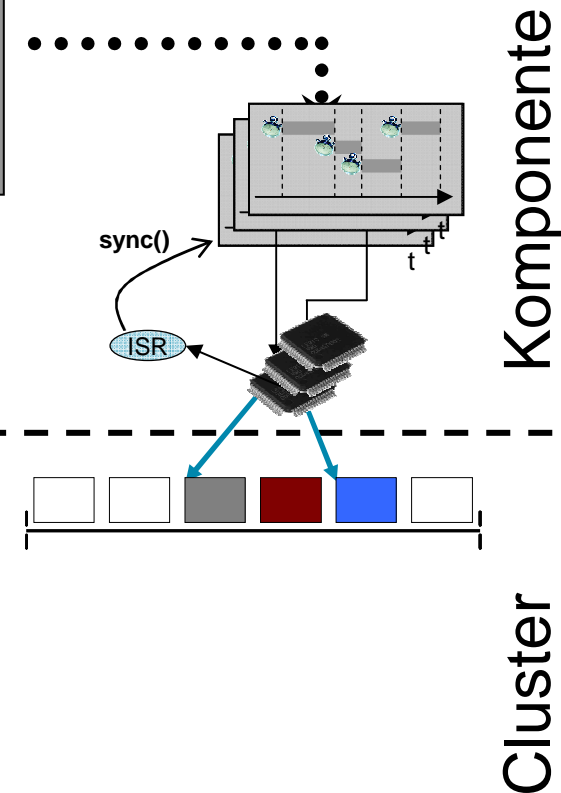
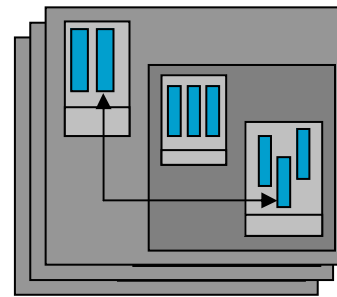
FlexRay als deterministisches Bussystem



- Kommunikationszyklen
- (F) TDMA
 - statisches Segment
 - dynamisches Segment
- Zugriffsmechanismen
 - Slot-based
 - Cycle-based
- verteilte Uhrensynchronisation
- Skalierbarkeit
 - 100% Last
 - konstanter Integrationsaufwand

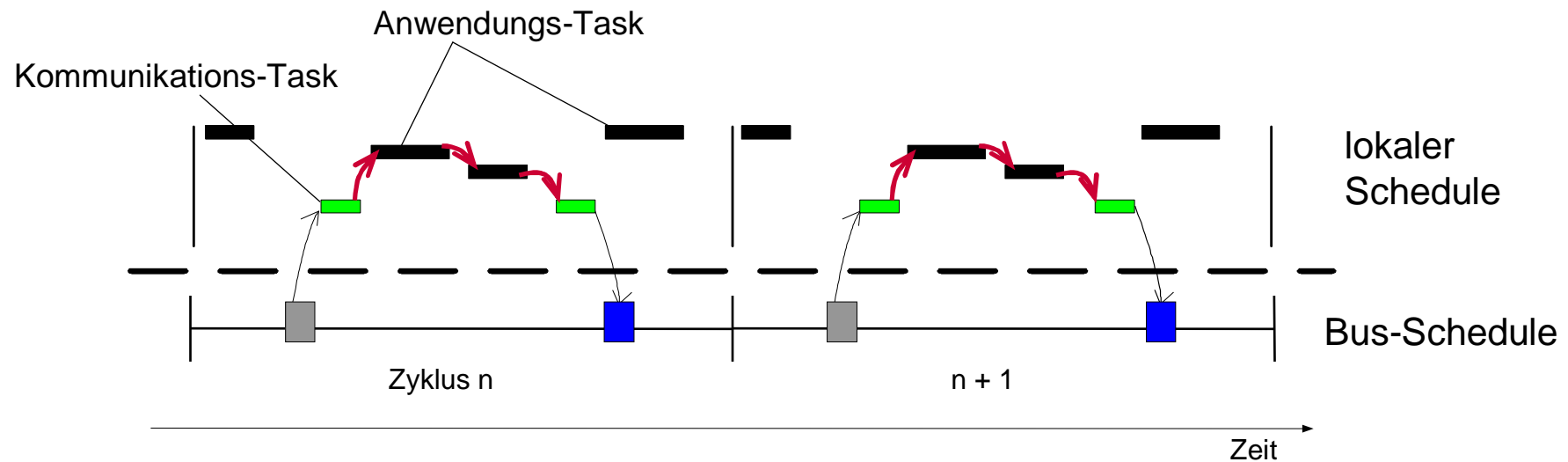
zweistufiger Entwicklungsansatz

- SW-Architektur
- lokaler Ablaufplan
- Kommunikationsabstraktion
- lokale Synchronisation
- composability



- periodischer Ablaufplan für Signalversand
- Kommunikationsmatrix
- Netzwerkparameter und -topologie

timing-Anforderungen: in-cycle response



kausale / temporale Abhängigkeiten

 timing-Modell auf Entwurfsebene

Timing-Modell: Kernkonzepte

- **Systemattribute**
 - SW-Architektur (Taskmodell, Runnables, ...)
 - Trigger-Konzepte
- **Timing-Attribute**
 - properties (Bus-Schedule, Task-Laufzeiten, OS overhead ...)
 - constraints (HW-Beschränkungen, Abhängigkeiten)
 - requirements (Sensor/Aktor-Deadline, Mess-/Steuerfreq., ...)
 - produce/consume
- **satisfy**

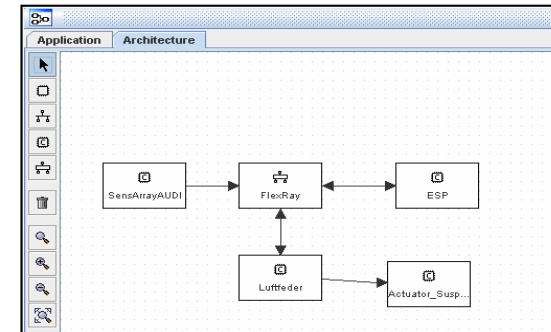
Scheduling-Analyse: SYMTA VISION

Systemmodell

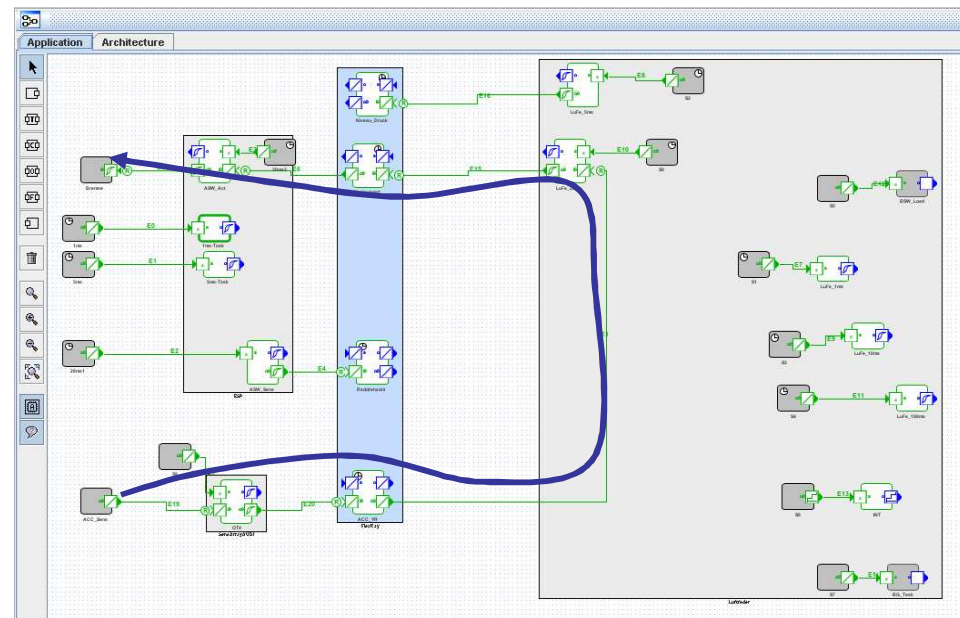
- Topologie
- Busparameter
- lokale Taskmodelle

Analyse-Gegenstand

- End-to-end Pfad / Timing Chain



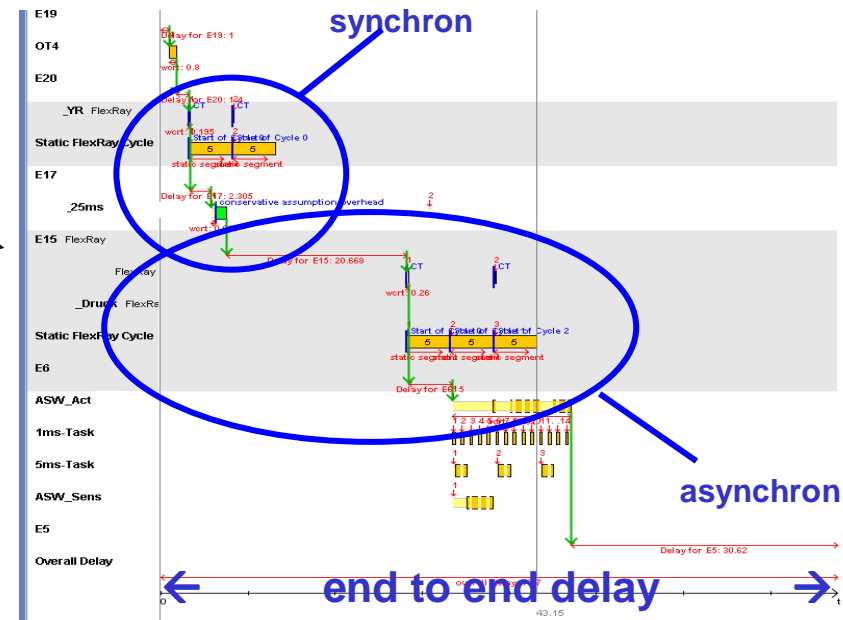
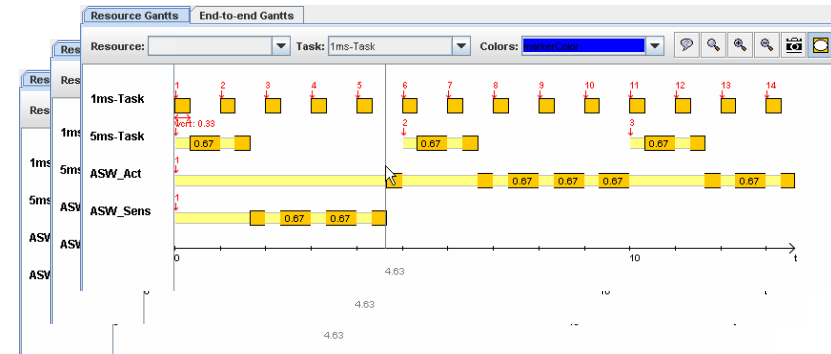
Topology



Application

Scheduling-Analyse: SYMTA VISION

- Task- und Frame-Laufzeiten
- Scheduling-Diagramme pro ECU & Bus
- End-to-end Timing Visualisierung
- Vergleich verschiedener Synchronisierungs-Optionen



Fazit

Ausblick und Zusammenfassung

Zusammenfassung

➤ AUTOSAR

- einheitliche SW-Architektur → Wiederverwendung
- Standardisierung → Versionsstabilität notwendig
- Werkzeugunterstützung

➤ FlexRay

- veränderter Entwicklungsansatz
- höherer Entwurfsaufwand
- vereinfachte Systemintegration

➤ timing

- steigende Anforderungen
- AUTOSAR-Unterstützung erforderlich

Ausblick

- **werkzeuggestützte Erfassung für timing-Modell**
 - SW-Architektur: AUTOSAR-Import
 - timing-Attribute: vorgehengestützt

- **(lokales) Scheduling**
 - heuristischer Ansatz

- **Forschungsprojekte: TIMMO**



Vielen Dank!