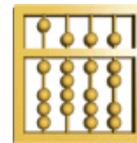

Software Engineering: Software im Automobil

3. Vorlesung: Eingebettete Systeme

Manfred Broy



Technische Universität München
Institut für Informatik



Charakteristika eingebetteter Systeme

Eingebettete Systeme

- nehmen Informationen aus ihrer Umgebung über Sensoren, Nutzerschnittstellen oder über das Netzwerk von anderen informationsverarbeitenden Komponenten auf,
- speichern und verarbeiten diese unter Nutzung gespeicherter Informationen und
- geben Informationen an ihre Umgebung über Aktuatoren, Nutzerschnittstellen oder über das Netzwerk an andere informationsverarbeitende Komponenten ab.
- sind eng gekoppelt an physikalische oder technische Systeme, die sie steuern, regeln und überwachen.
- unterliegen in der Regel Zeitanforderungen.

Zentral: Echtzeit, Interaktion und Reaktivität

Unterscheidung: Isolierte oder vernetzte eingebettete Systeme

- Eingebettetes System
Eine Kombination von Computerhardware und Software, entworfen um eine bestimmte Funktion zu erfüllen
- Eingebettete Hardware
Hardware (Steuergerät) ist in technische Umgebung eingebettet (Sensoren, Aktuatoren, Kommunikationsnetz) und somit untrennbar damit verbunden
- Eingebettete Software
Wird auf eingebetteter Hardware ausgeführt, „liest“ Werte aus der Umgebung (oft Sensorwerte) und „schreibt“ Werte an die Umgebung (oft Aktuatorwerte)

Eingebettete Softwaresysteme im Fahrzeug

- In Fahrzeugen dienen eingebettete Softwaresysteme der Verarbeitung von Informationen mit dem Ziel
 - ◇ Physikalische und technische Vorgänge zu überwachen, zu steuern und zu optimieren (Motor, Bremsen, Schließung, Energiemanagement etc.).
 - ◇ Den Fahrer zu unterstützen (Bremsassistent, Spurerkennung, Parkassistent, Geschwindigkeitsregelung, Mensch-Maschine-Schnittstelle etc.).
 - ◇ Die passive Sicherheit zu erhöhen (Airbag, Gurtstraffer, Precrash, etc.)
 - ◇ Informationen für den Fahrer bereitzustellen (Infotainment).
 - ◇ Höheren Komfort zu schaffen (Sitzverstellung mit Memory etc.)
- Neben diesen Primärfunktionen kommen immer stärker Sekundärfunktionen hinzu, die eine Infrastruktur bereitstellen, die für die Primärfunktionen hilfreich ist (Kommunikationsdienste, Fehlerbehandlung, Diagnose etc.)

- Verknüpfung mit physikalischen und technischen Prozessen
- Reaktivität
- Nebenläufigkeit
- Echtzeit
- Vernetzung untereinander
- Multiplexing auf Bussen und Steuergeräten
- Mensch Maschine Interaktion
- Integration in ein Produkt
- Hohe Zuverlässigkeitsanforderungen

Unterschiedl. Charakteristika der Software im Fahrzeug

- Regelungstechnik (Beispiel: Motorsteuerung)
 - ◇ Sehr genaue Modelle der Regelstrecke
 - ◇ Harte Echtzeit (auch geringfügiges Überschreiten von Zeitschranken unakzeptabel)
 - ◇ Reaktivität
 - ◇ Schwingungsfreies, ruckfreies (sanftes) Verhalten
- Diskrete Zustandssysteme (Schließung)
 - ◇ Reaktivität
 - ◇ Interaktion
- Telekommunikation
 - ◇ Weiche Echtzeit (kein absoluten Zeitvorgaben, leichtes überschreiten von Zeitvorgaben unproblematisch)
- Informatiksysteme (Betriebssystem, Busse/Protokolle)
 - ◇ Typische Systemprogrammierung

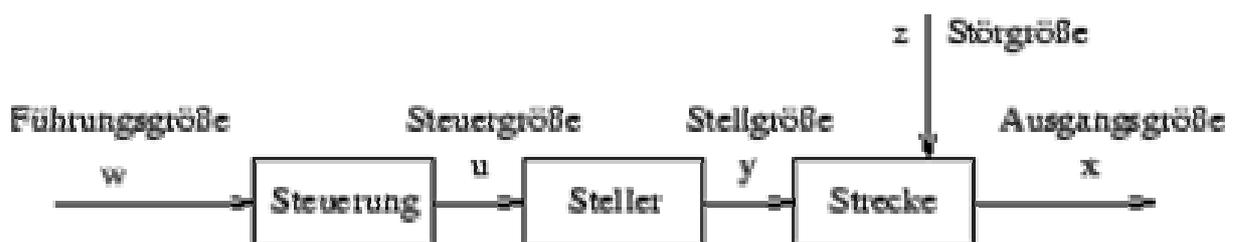
- Bei einer Regelung wird die zu regelnde Größe (**Regelgröße** x) eines Prozesses fortlaufend gemessen und mit einem vorgegebenen Wert (**Soll- oder Führungsgröße** w) verglichen. Besteht zwischen diesen beiden Größen eine Abweichung (Regeldifferenz oder Regelfehler e bzw. Regelabweichung xw), wird abhängig von der Regelabweichung eine den Prozess beeinflussende **Stellgröße** (u) derart verändert, dass die Regelgröße mit der **Führungsgröße** wieder in Übereinstimmung gebracht wird.
- Der Begriff der Regelung oder des Regelkreises bezeichnet diesen geschlossenen Wirkungskreis (Rückkopplung).
- Als Regler wird diejenige Komponente im Regelkreis bezeichnet, die in Abhängigkeit des Regelfehlers die Stellgröße bestimmt. Der Entwurf eines zum Prozess passenden Reglers ist eine Hauptaufgabe der Regelungstechnik.

- In der DIN 19226 ist der Begriff der Regelung wie folgt definiert
Das Regeln, die Regelung, ist ein Vorgang, bei dem fortlaufend eine Größe, die Regelgröße (zu regelnde Größe), erfasst, mit einer anderen Größe, der Führungsgröße, verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird. Kennzeichen für das Regeln ist der geschlossene Wirkungsablauf, bei dem die Regelgröße im Wirkungsweg des Regelkreises fortlaufend sich selbst beeinflusst.
- Teil dieser Definition sind drei Kriterien zur Bestimmung einer Regelung nach einem Entwurf für die DIN 19226 von 1962 und der "Nomenklatur der Regelungstechnik" des Schweizer elektrotechnischen Vereins von 1956
 - ◇ Die Aufgabe einer Regelung ist Befehlsausführung im Allgemeinen; sie soll eine Größe (die Regelgröße) einer vorgegebenen Größe (die Führungsgröße) gleich machen.
 - ◇ Eine Regelung wirkt in einem geschlossenen Kreis mit negativer Rückkopplung.
 - ◇ Eine Regelung besitzt ein Element zum Erfassen der Regelgröße, das von den Einrichtungen, die in die Regelstrecke eingreifen, physisch getrennt ist.
- Der Begriff Regelung ist zu unterscheiden von dem im allgemeinen Sprachgebrauch oft synonym gebrauchten Begriff der Steuerung - bei dieser fehlt die Rückkopplung.

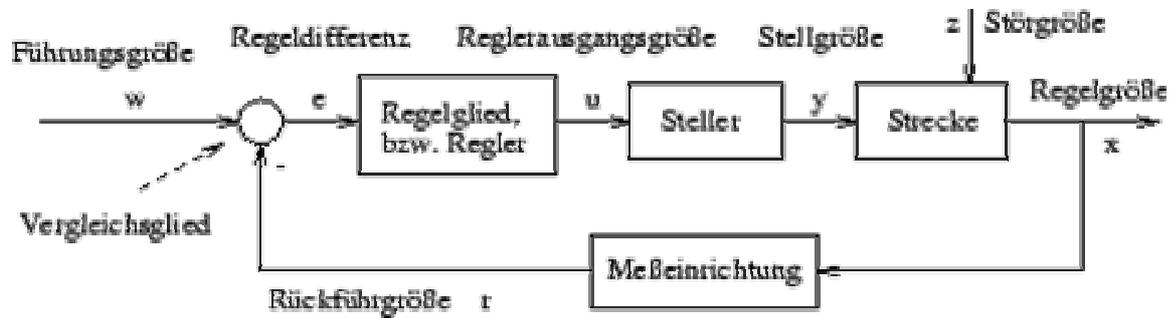
Regelung und Steuerung

- Durch die Regelung wird bei einer nur unvollständig bekannten Strecke, insbesondere bei unbekannter Störgröße, die Regelgröße fortlaufend gemessen und mit der gewünschten Führungsgröße verglichen, um dann mittels der so gebildeten Regeldifferenz über den Regler die Regelgröße an den vorgegebenen Sollverlauf anzugleichen.
- Charakteristisch für eine Regelung ist die sich zwangsläufig ergebende Rückführungsstruktur bzw. der geschlossene Wirkungskreislauf (Regelkreis). Abweichungen der Regelgröße vom gewünschten Sollwert werden selbsttätig ausgeglichen bzw. "ausgeregelt".
- Charakteristisch für eine reine Steuerung ist der offene Wirkungskreislauf. Ein Beispiel ist die gesteuerte Raumheizung.
- Problem: Unbekannte Störgrößen führen zu nicht korrigierbaren Abweichungen der Ausgangsgröße vom Sollwert.
- Es gibt aber auch viele Steuerungen mit Rückführungen. Dabei wird jedoch die Regelgröße nicht fortlaufend erfasst und mit einem Sollwert verglichen, sondern es werden lediglich bestimmte Schaltbedingungen, d.h. im allgemeinen binäre Größen, zurückgemeldet (z.B. Füllstand erreicht, Ventil geschlossen).

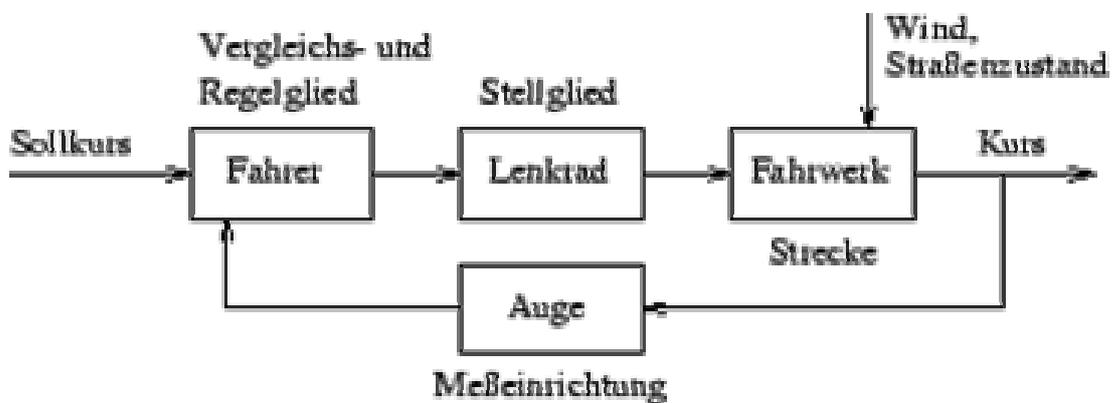
Steuerung



Regelung



Beispiel zur Regelungstechnik



Modellbildung für Regelungsaufgaben

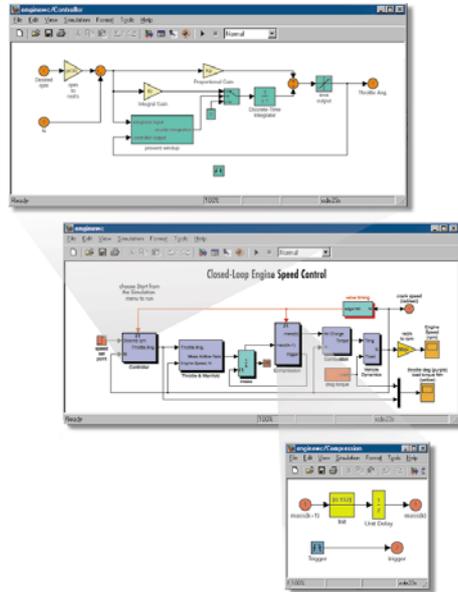
- Die Regelungsaufgaben werden durch Differential- und Integralgleichungen erfasst
- Als Modellierungswerkzeug wird beispielsweise Matlab Simulink verwendet
- Zur Implementierung wird dieses Modell mit Techniken der Numerik und Diskretisierung in Programme umgesetzt

Überblick über MATLAB / Simulink / Stateflow

```
function f = hcurve(t)
    f = sqrt(4*cos(2*t).^2 + sin(t).^2 + 1);
    len = quad(@hcurve,0,3*pi);
```

- MATLAB ist
 - ◇ Von MathWorks.com
 - ◇ Sprache (interaktiv mit Interpreter)
 - ◇ Zahlreiche mathematische und graphische Bibliotheken
 - ◇ Framework (Tool-Architektur)
 - ◇ Flexibel Erweiterbar
 - ◇ Viele Erweiterungen, z.B.
 - Simulink (Blöcke + graphische Simulation)
 - Stateflow (State Charts)
 - Realtime Workshop (Code-Generierung)
- Verbreitete Plattform (z.B. auch Bildverarbeitung)

- Graphische Simulationsumgebung für MATLAB für:
 - ◇ Regelungssysteme
 - ◇ Umgebungsmodelle
- Einsatz im Embedded Bereich
 - ◇ Automotive
 - ◇ Avionik
- sehr untypisiertes Werkzeug (Float und Laufzeit-Strukturen)
- Konsistenzchecks (Verbindungen, „Typen“, ...)

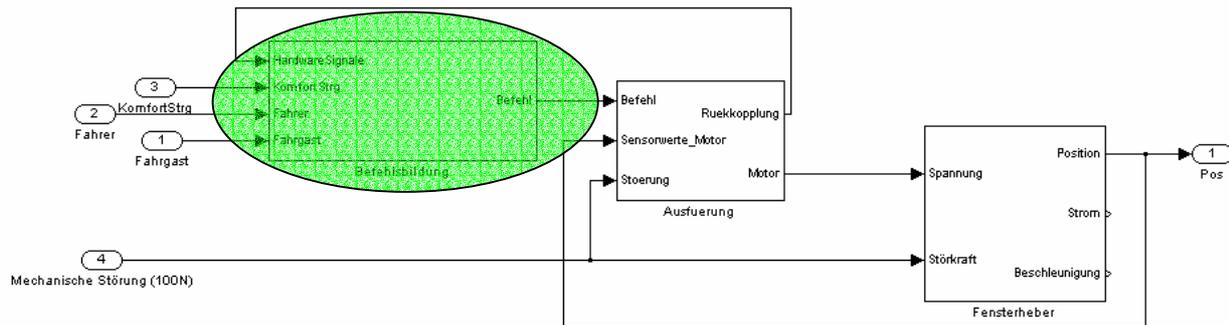


Simulink-Simulation

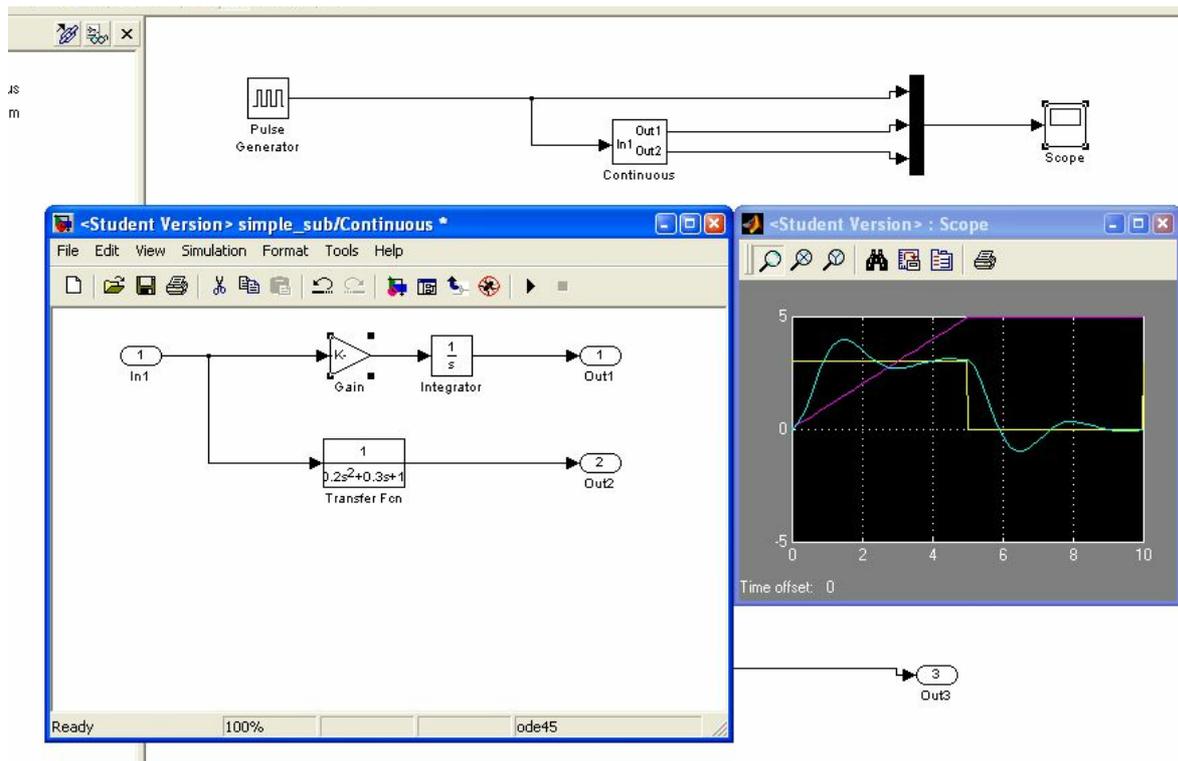
- Ablauf:
 - ◇ Eingänge mit „Generatoren“ belegen, z.B. Pulse Generator
 - ◇ Parameter einstellen, z.B. Höhe, Frequenz
 - ◇ Ggf. Interpolationsverfahren wählen (Euler-Forward/Backward, Fixed Step,..)
 - ◇ Ausgänge mit „Beobachtern“ (Scope) belegen
 - ◇ Simulation starten
- Weitere Features
 - ◇ M-Funktionen (beliebiges MATLAB) einbinden
 - ◇ S-Funktionen (beliebigen Code) einbinden
 - ◇ Protokollierung

Umsetzung in Matlab (1)

- „Befehlsbildung“ und „Ausführung“ stellen den diskreten/kontinuierlichen Anteil der Steuerung dar
- Hier: Vereinfachung durch Nutzung von Informationen aus dem Modell (z.B. Position)
- Auswertung von Sensoren später im Modul Ausführung
- Block „Fensterheber“ repräsentiert die Regelstrecke

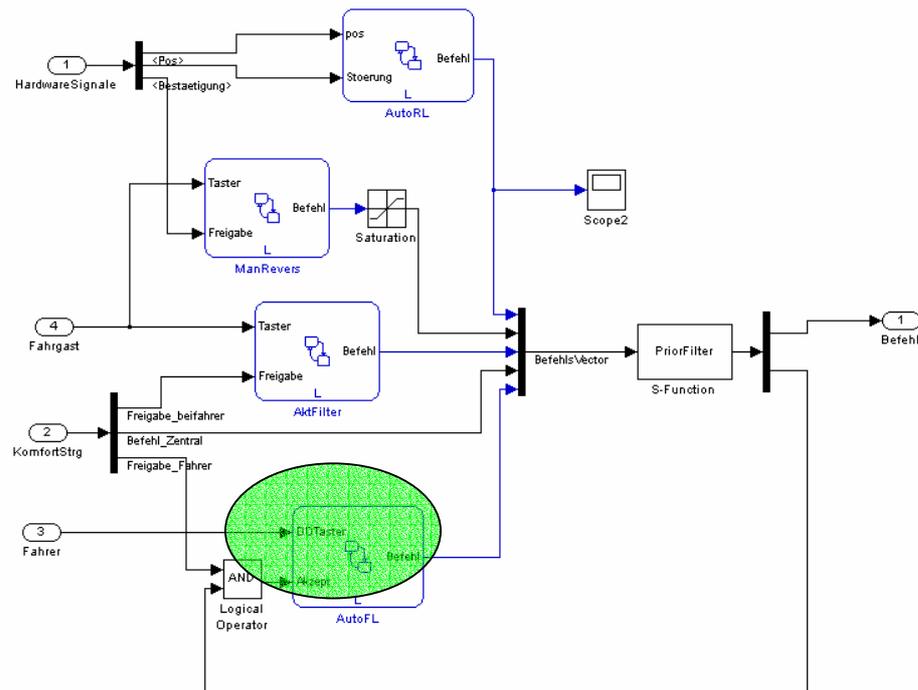


Beispiel: Übertragungsfunktion



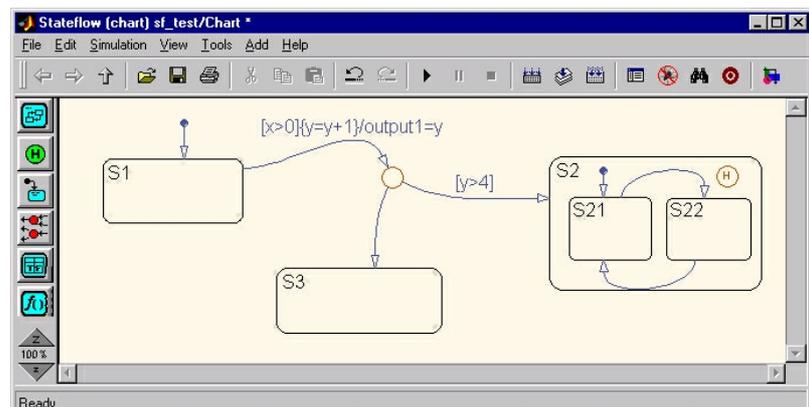
Umsetzung in Matlab

- Ablaufsteuerung zur Bedienung der Fensterheber



Simulink/Stateflow

- Erweiterung zur Modellierung von State Charts in MATLAB
 - ◇ Hierarchische Zustände
 - ◇ Junctions
 - ◇ History States
 - ◇ Events
 - ◇ MATLAB-Funktionen



MATLAB-Erweiterungen

- Simulink Performance Tools (Analysen)
- Realtime Workshop Embedded Coder
- Stateflow-Coder
- xPC Target (In the loop simulation)
- Simulink-Report Generator (Dokumentation)
- DOORS-Kopplung
- Toolboxen (Neuronal Netze, Fuzzy Logic,..)
- Embedded Targets (DSP-Generator,..)
- Bibliotheken („Block sets“)
- AutoFOCUS Importer

MATLAB - Zusammenfassung

- Marktführer in Simulation => viele Möglichkeiten
- Pragmatisches Werkzeug => Semantik schwierig, z.B.
 - ◇ Schleifen in Transitionen möglich
 - ◇ Prioritäten gemäß Layout („Uhrzeigersinn“)
 - ◇ Modellierungsrichtlinien üblich

- Ein Datenmodell („Abstract Data Type“) beschreibt die Datenelemente eines Systems.
- Daten treten in folgenden Rollen auf:
 - ◇ Als Attribute von Zuständen
 - ◇ Als Parameter in Methoden- und Prozeduraufrufen
 - ◇ Als Signale, Nachrichten oder Ereignisse
- Daten werden am besten durch
 - ◇ Datentypen (Namen für Datenmengen)
 - ◇ Und ihre charakteristischen Operationen beschrieben

Sichten auf eingebettete Systeme

- Technisch sind eingebettete Systeme Hardwareeinheiten (Steuergeräte), die über Bussysteme vernetzt sind, mit Sensoren und Aktuatoren und Benutzerschnittstellen verbunden sind, und mit diesen elektronische Signale austauschen. Der Signalfluss repräsentiert auf technischer Ebene das Verhalten.
- Die Software in den Steuergeräten besteht aus Folgen von Anweisungen („Code“), deren Abarbeitung das dadurch festgelegte Verhalten (den Signalfluss) generiert. In den Steuergeräten werden nach festgelegten Verfahrensweisen Codefolgen zur Ausführung gebracht. Diese entstammen teilweise der vorgegeben Infrastruktur (Betriebssystem, Netzwerkmanagement, Systemdienste oder den Anwendungsprogrammen).
- Der Signalfluss repräsentiert zwar das Verhalten eines Systems auf technischer Ebene. Allerdings ist dieses Verhalten in dieser Form kaum zu verstehen. Deshalb wählt man abstraktere Modelle um das Verhalten aus Sicht des Nutzers nachvollziehbar darzustellen.

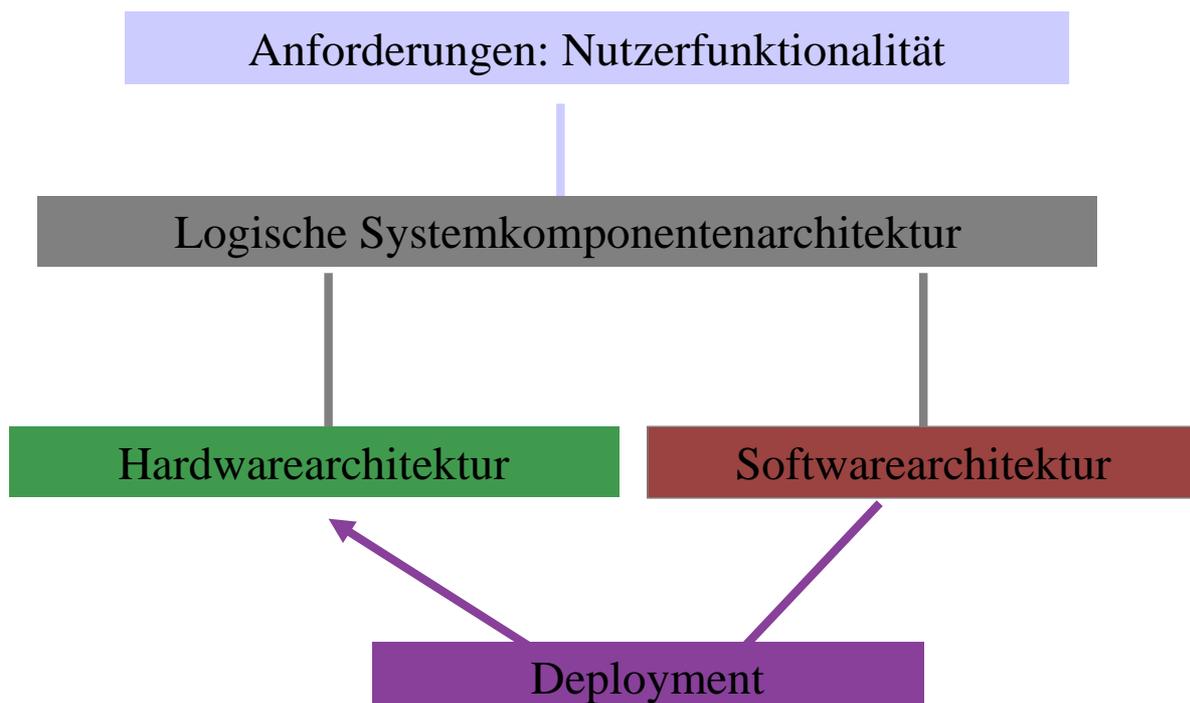
Systemarchitektur

- Die Systemarchitektur ist die strukturierte Darstellung aller wesentlichen Aspekte eines multifunktionalen Systems

Wesentliche Gesichtspunkte:

- Funktionenarchitektur (Nutzungsarchitektur): Strukturierte Darstellung aller Nutzerfunktionen und Abhängigkeiten (Anforderungen)
- Komponentenarchitektur: Zergliederung des Systems in Teilsysteme („Komponenten“), die in ihrem Zusammenwirken die Gesamtsystemfunktionalität erbringen
- Hardwarearchitektur: Aufbau des Systems aus Hardware (Prozessoren, Sensoren, Aktuatoren, Bedieninstrumente etc.) und deren Vernetzung (Bordnetz)
- Softwarearchitektur: Gliederung des Softwaresystems in Softwarekomponenten
- Deployment: Abbildung der Softwareeinheiten auf die Hardware

Systemarchitektur



Die Hardwarearchitektur besteht aus den mechatronischen Geräten wie:

- Prozessoren (Steuergerätehardware)
- Kommunikationsverbindungen (Leitungen, Geräte zur drahtlosen Übertragung)
- Sensoren und Aktuatoren (Geräte zur Wandlung von Meßwerten in elektronische/digitale Daten oder zur Umsetzung von elektronischen/digitalen Daten in Steuergeräten)
- Geräte der Mensch-Maschine-Schnittstelle MMI - Man Machine Interface (Bedien- und Anzeigeeinstrumente)

Mikrokontroller: Steuergeräte/Prozessoren (ECU)

- Ein Mikrokontroller ist eine Hardwareeinheit, die der Ausführung eines Programms (Software) dient.

Ein Mikrokontroller besteht aus:

- Mikroprozessor
- Programm- und Datenspeicher, Register, Stack
- Internem Bussystem
- Taktgenerator
- Überwachungsschaltungen zur Ausführung des Programms
- Ein/Ausgabeeingänge und Steuergeräte für die Ein/Ausgabe

- **Zeitanforderungen**
 - ◇ Systemreaktion müssen in bestimmten Zeitrahmen erfolgen, eng auf die zeitlichen technischen oder physikalischen Prozesse ausgerichtet
 - ◇ Die Festlegung der Zeitanforderungen ist Teil des Requirements Engineerings
- **Zeitverhalten**
 - ◇ Das realisierte System zeigt ein Zeitverhalten, das sich aus den Reaktionszeiten und der Ausführungsgeschwindigkeit der Systembestandteile ergibt
 - ◇ Durch Scheduling und Multiplexing wird das Zeitverhalten nichtdeterministisch
- **Verifikation des Zeitverhaltens**
 - ◇ Es ist sicherzustellen, dass das Ist-Zeitverhalten dem Soll entspricht

Ereignis- versus Zeitsteuerung

Ereignissteuerung:

Zeitliche Steuerung durch Ereignisse (Interrupt) gesteuert

- keine präzise zeitliche Spezifikation der Schnittstellen
- unbestimmte Latenzzeiten
- geeignet für sporadische Daten
- nicht deterministisch
- nur statistisch abschätzbar

Zeitsteuerung

Zeitliche Steuerung durch Voranschreiten der Zeit

- Schnittstellen sind zeitliche Firewalls
- definierte Latenzzeiten
- Regelalgorithmen stabil realisierbar
- Nachrichten sind Zustandsmeldungen
- deterministisch
- statisch berechenbar

Plattform

- Eine Plattform besteht aus einer Reihe von Diensten im Sinne einer Betriebsumgebung, auf denen eine Anwendung aufgesetzt werden kann.

Auf den Steuergeräten läuft typischerweise Systemsoftware, die die technischen Funktionen in der Form von Diensten erschließen.

- Betriebssystem
- Protokolle der Bussysteme
- Treiber für Sensoren, Aktuatoren und MMI

Wir unterscheiden:

- Hardwareplattform: Mikrokontroller, Sensorik, Aktuatorik und hardwaretechnische Kommunikationsverbindungen
- Systemplattform: Betriebssysteme, Kommunikationsdienste (i.d.R. unabhängig von der konkreten Hardwareplattform).

Plattformunabhängigkeit und Portierbarkeit

Bei der Übertragung von Software aus einer Plattform in ein andere ändert sich das Verhalten durch unterschiedliche Gegebenheiten bei den Betriebsmitteln und Laufzeiten.

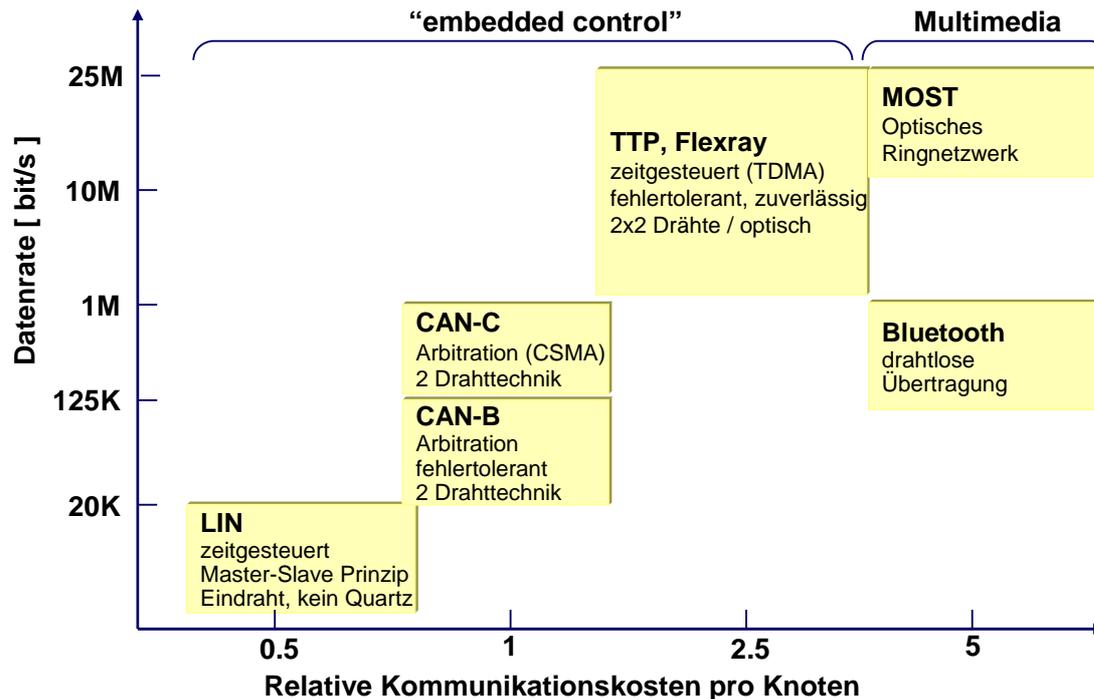
Ein Ziel ist es,

- die Unterschiede der Software im Verhalten auf unterschiedlichen Plattformen so gering wie möglich zu halten.
- trotz dieser Unterschiede im Verhalten das Gesamtverhalten im Sinne der Anforderungen zu garantieren.

- Das Kommunikationssystem umfasst alle Hard- und Software, die dem gesicherten Austausch von Daten zwischen den einzelnen Teilen der auf mehrere Steuergeräte verteilten Anwendungen ermöglicht.

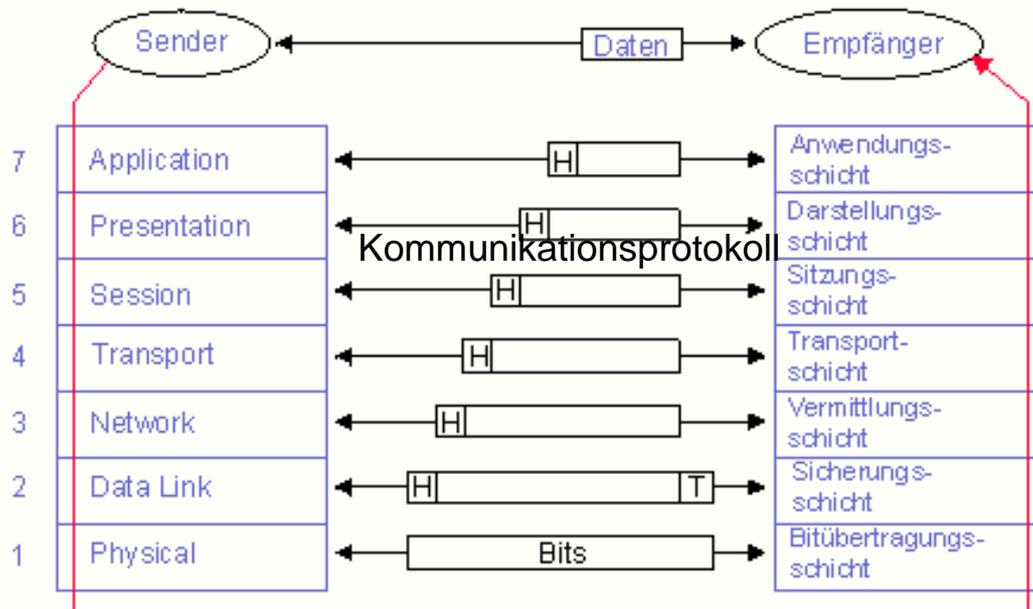
Bussysteme

- Bussysteme dienen der Kommunikation zwischen den Steuergeräten. Ein Bus ist ein Kommunikationsmedium, das auf Basis einer technischen Verbindung (Leitung, Draht, drahtlose Übertragung) einen Kommunikationsdienst bereitstellt.
- Wir unterscheiden zwischen den internen Bussen auf Mikrocontrollern, die Prozessor, Speichermedien und Ein/Ausgabeeinheiten verbinden und den externen Bussen, die die Steuergeräte, Sensoren, Aktuatoren und Bedieninstrumente verbinden
- Bei externen Bussen unterscheiden wir zwischen der technischen Verbindung und der Übertragungsdisziplin (Übertragungsprotokoll)



Protokolle

- Bussysteme verwenden in der Regel Kommunikationsprotokolle
- Ein Kommunikationsprotokoll definiert eine Konvention zum Nachrichtenaustausch in elektronischen Systemen.
- In einem Kommunikationsprotokoll sind alle Regeln für den Nachrichtenaustausch festgelegt.
- Dies erfolgt in der Regel in einem Schichtenmodell, in dem die unterschiedlichen Aspekte in Schichten strukturiert sind.
- Die Schichten bauen aufeinander auf und definieren so immer abstraktere Sichten auf die Funktionen des Protokolls.



FlexRay - Das Protokoll

- Echtzeitfähiges Kommunikationssystem konzipiert fürsicherheitsrelevante Anwendungen
- zeitgesteuerter Teil (Zeitmultiplex-Verfahren, konstante Slotlänge)
- ereignisgesteuerter Teil (Minislotting-Verfahren)
- fehlertolerante, clusterweite Zeitbasis (nutzbar von der Applikation)
- deterministisch im statischen Teil (eindeutig vorhersagbares Systemverhalten)
- redundante Übertragungskanäle (physikalische Trennung)
- Bus Guardian (elektrische Trennung)

Da die direkte Programmierung eines ECUs viel zu aufwändig wäre und die Software zur Hardware-abhängig läuft auf der Hardware ein Betriebssystem. Dies ist Software, die folgende Aufgaben erfüllt:

- Systemstart
- Betriebsmittelverwaltung
 - ◇ Die Steuerung der Ausführung der Anwendungsprogramme (Scheduling, Unterbrechungskonzept, Alarmbehandlung)
 - ◇ Speicherverwaltung
- Kommunikationsdienste

Echtzeit

Die Steuerung der Systeme in einem Fahrzeug findet in Echtzeit statt

- Bestimmte Steueraufgaben erfordern das Einhalten genauer Zeitvorgaben („Echtzeitanforderungen“)
- Auf der Hardware laufen die Prozesse in vorgegebenen Zeitrahmen (Takt) ab. Die Echtzeitanforderungen müssen unter Beschränkung durch diese Zeitrahmen und durch vorhandenen Betriebsmittel erfüllt werden
- In der Regel konkurrieren unterschiedliche Aufgaben („Tasks“) um die Betriebsmittel, die diesen durch Systembetriebsverwaltungsfunktionen („Betriebssystem“) scheinbarweise zugeteilt werden.
- Dadurch entstehen in den unterschiedlichen Betriebssituationen „zufällige“ Verzögerungen und ein „nichtdeterministisches“ Verhalten der Systeme

- Aktuatoren und Sensoren sind mechatronische Systeme, die aus elektrischen, hydraulischen, pneumatischen und mechanischen aber zunehmend auch aus elektronischen Bestandteilen aufgebaut sind.
- Sensoren dienen der Erfassung von Messwerten und der Umsetzung in elektronische Signale
- Aktuatoren dienen der Regelung von physikalischen Vorgängen durch die Umsetzung von elektronischen Signalen

Deployment

- Ein multifunktionales eingebettetes System besteht aus Hardware, Software und Mechanik.
- Die Software realisiert Funktionen und wird in Form von Code auf Steuergeräte aufgebracht; dieser Code wird auf dem Steuergerät ausgeführt.
- Die Hardware (Steuergeräte) bildet ein durch Kommunikationseinrichtungen (Bussysteme) verbundenes Netzwerk
- Auf einem Steuergerät können mehrere Software-realisierten Funktionen zur Ausführung kommen. Dann muss allerdings die Leistungsfähigkeit der Hardware dafür ausreichen.
- Die Abbildung der Software, die die einzelnen Funktionen realisiert, auf die Hardware nennt man Deployment.

Scheduling

Sowohl auf den Steuergeräten, wie auch auf den Busses werden unterschiedliche Aufgaben („Tasks“) zeitlich verschränkt ausgeführt.

- Die Auswahl, welche Task zu welchem Zeitpunkt auf einem Gerät zur Ausführung kommt, nennt man Scheduling.
- Das Scheduling bestimmt entscheidend das Echtzeitverhalten
- Wir unterscheiden:
 - ◇ Statisches Scheduling/Zeitsteuerung: Für bestimmte (in der Regel periodisch auftretende) Tasks werden zu statisch festgelegten Zeitpunkten Betriebsmittel bereitgestellt.
 - ◇ Dynamisches Scheduling/Ereignissteuerung: Für die Tasks werden zu dynamisch abhängig von der Betriebssituation („Last“) unter Berücksichtigung von Prioritäten Betriebsmittel bereitgestellt

Schichtenarchitekturen

Die Schichtenbildung wie im ISO/OSI Schichtenmodell ist typisch für die Arbeitsweise und Systemstrukturen der Informatik

- Durch Schichtenbildung wird von realisierungsnahen Details abstrahiert
- Es werden stufenweise höherwertige Funktionen und Dienste auf Grundfunktionen und -diensten aufgebaut
- Dies dient der Komplexitätsreduktion durch Verbergen von Implementierungsdetails und der Flexibilität der Änderung und Wiederverwendbarkeit von Lösungsdetails (auch bei Portierung)

Nutzungssicht: Die Nutzungsarchitektur

- In der Nutzungsarchitektur werden alle vom System angebotenen Funktionen strukturiert aufgelistet und beschrieben.
- Die strukturierte Auflistung kann in Form eines Featuretrees (siehe auch PLA - Produktlinienansatz) erfolgen oder in Form eines Funktionsnetzes. Dabei sind insbesondere die funktionalen Abhängigkeiten zu dokumentieren.
- Die Funktionen können im Detail durch Use Cases beschrieben werden
- Wir können bezogen auf die Stakeholder zwischen Funktionen für den Fahrer/Beifahrer und Funktionen für Wartung unterscheiden

Ziel- und funktionsorientiertes RE/SE

- **Grundlagen:**
 - ◇ Multi-funktionale Systeme
 - ◇ Strukturierte Darstellung der Systemfunktionen
 - ◇ Abstraktionsebenen
 - ◇ Funktionenhierarchie
 - ◇ Übergang zur Architektur
- **Vision:** Ziel- und funktionsorientiertes Software Engineering
 - ◇ Funktionen als Anforderungsbausteine
 - ◇ Funktionen als Aspekte
 - Schnittstellen
 - Architekturen

In Fahrzeugen finden sich heute bis zu 2500 unterschiedliche Funktionen, die durch Software realisiert sind

Deshalb ist eine strukturierte Sicht auf die Nutzerfunktionalität (Funktionsnetzhierarchie) unabdingbar

Wesentliche Aspekte der Nutzungssicht

- Funktionsnetzhierarchie:
 - ◇ Gibt an, welche Funktionen in welchen Anwendungsfeldern verfügbar sind.
 - ◇ Welche Abhängigkeiten bestehen.
- Detaillierte Funktionsspezifikation
 - ◇ Gibt für jede einzelne Funktion die logische Funktionsweise an.
- Mensch Maschine Schnittstelle
 - ◇ Legt fest, wie ein Nutzer auf eine Funktion konkret zugreifen kann.

Achtung:

Nicht nur Fahrer sind Nutzer, auch Wartung etc.

Die Mensch Maschine Schnittstelle (Man Machine Interface - MMI) dient der Interaktion menschlicher Nutzer oder Betreiber mit dem System

Das MMI besteht aus

- allen optischen, akustischen oder haptischen Instrumenten und Anzeigen zur Vermittlung von Informationen und
- allen Einrichtungen (Hebel, Köpfen, Schieber, akustische Eingabe, Sensoren) zur Erfassung von Nutzerkommandos.

Modellierung von IT Systemen

- IT Systeme haben eine Schnittstelle (Systemgrenze nach außen), Struktur (einen inneren Aufbau) und ein Verhalten
- Es gibt zwei wesentliche Ansätze zur Modellierung des Verhaltens von IT Systemen
 - ◇ Zustandsmodelle: Ein System wird beschrieben, indem die Menge seiner Zustände beschrieben wird und seine Aktivitäten als Übergänge zwischen Zuständen
 - ◇ Ablaufmodelle: Ein System wird beschrieben, indem die Menge seiner Aktivitäten beschrieben wird und die Menge der Folgen von Aktivitäten, genannt Abläufe, die in dem System auftreten können.
- Beide Ansätze ergänzen sich. Jedes Zustandsmodell kann zu einem Ablaufmodell erweitert werden und umgekehrt.

- *Verteilung*: Ein verteiltes System besteht aus einer Familie eigenständiger Komponenten mit einer räumlichen oder konzeptuellen (strukturellen oder logischen) Verteilung. Die Verteilung kann nach unterschiedlichen Gesichtspunkt erfolgen, wie etwa nach räumlichen Gesichtspunkten oder etwa aufgabenorientiert.
- *Parallelität*: Die Komponenten eines Systems führen gewisse Aktionen zeitlich nebeneinander (gleichzeitig - „simultan“) durch.
- *Kommunikation, Reaktion und Interaktion*: Das System tauscht mit seiner Umgebung oder die Komponenten des Systems tauschen untereinander oder mit ihrer Systemumgebung Informationen aus (durch das Senden und Empfangen von Nachrichten und Signalen oder durch Zugriff auf gemeinsamen Speicher).
- *Koordination*: Gewisse zeitliche oder kausale Abhängigkeiten, Beziehungen und Einschränkungen bestehen zwischen den Aktionen der unterschiedlichen Komponenten und sind im Sinne der Aufgabenstellung erforderlich.
- *Quantitative Zeit*: Die zeitlichen Beziehungen zwischen den und die Zeitdauer der Aktionen sind für Systemwirkung u.U. entscheidend (Echtzeitanforderungen)

Sichten auf ein System

- **Datensicht**:
Beschreibt die Daten eines Systems, ihre Struktur und ihre charakteristischen Funktionen
- **Schnittstellensicht**:
Beschreibt die Wechselwirkung zwischen einem System und seiner Umgebung.
- **Ablaufsicht**:
Beschreibt die Abläufe (Folgen von Aktionen oder Ereignissen) eines Systems
- **Zustandssicht**:
Beschreibt die Zustände, die ein System einnehmen kann und die auftretenden Zustandsänderungen
- **Verteilungssicht**:
Beschreibt die Gliederung („Verteilung“) eines Systems in Teilsysteme
- **Interaktionssicht**:
Beschreibt die Interaktion (den „Dialog“, Informationsaustausch) zwischen Systemteilen

Innensicht und Außensicht von Systemen

- Abhängig von der Zielsetzung bei der Modellierung eines Systems oder Teilsystems ist die Innensicht oder die Außensicht angemessen:
- Innensicht (Glassboxsicht, Whiteboxsicht):
Hier wird der innere Aufbau und seiner innere Funktionsweise eines Systems oder Teilsystems betrachtet, beispielsweise seine Untergliederung in Teilsysteme, seine lokalen Zustände und und Zustandsübergänge, seine internen und externen Aktionen.
- Außensicht (Blackboxsicht, Schnittstellensicht):
Hier wird der äußere Aufbau und seiner äußere Funktionsweise (Wirkung nach außen) eines Systems oder Teilsystems betrachtet, beispielsweise seine Reaktionen auf Ereignisse der Umgebung, die Ein-/Ausgaben, die externen Aktionen.
- Die Außensicht entsteht in der Regel als Abstraktion („Kapselung“) der Innensicht!

Beispiel Fensterheber - informelle Spezifikation

- Das Fenster wird über einen einfachen Wippschalter gesteuert, der drei Stellungen einnehmen kann: Hochfahren, Runterfahren, Arretiert
- Die Fenstersteuerung verfügt über folgende Funktionen:
 - ◇ Normalschließen/-öffnen
 - ◇ Einklemmschutz
 - ◇ Komfortschließen/-öffnen
- Wird die Wippe in Öffnen oder Schließstellung festgehalten, so fährt das Fenster bis in seine Endposition, falls nicht die Wippe vorher losgelassen wird.
- Wird die Wippe nur kurz angetippt, so schließt/öffnet das Fenster völlig, soweit der Vorgang nicht durch erneutes Antippen der Wippe oder durch Ansprechen des Einklemmschutzes unterbrochen wird
- Spricht beim Schließen des Fensters der Einklemmsensor an, so öffnet sich das Fenster völlig (unabhängig von der Stellung der Wippe), das Fenster geht in einen Nothalt, der erst verlassen wird, wenn die Zündung aus und wieder eingeschaltet wird.

Datenmodell Fensterheber

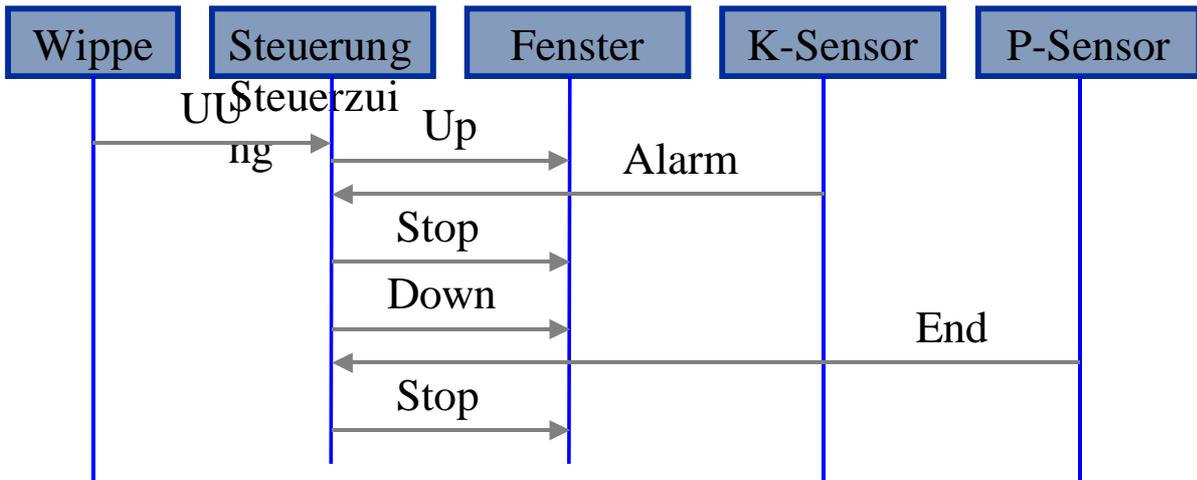
- Bereits im Datenmodell entscheiden wir über die Anstraktionsebene. So können wir die Stellung der Wippe wie folgt vereinbaren (für Down, Stop, Up):
- Type Wippe = {D, S, U}
- Wollen wir aber das kurze Antippen der Wippe nicht im Steuergerät, sondern im Schalter modellieren können wir auch wie folgt die Signale definieren, die von der Wippe ausgehen:
- Type Wippe = {DD, D, S, U, UU}

Strukturmodell Fensterheber

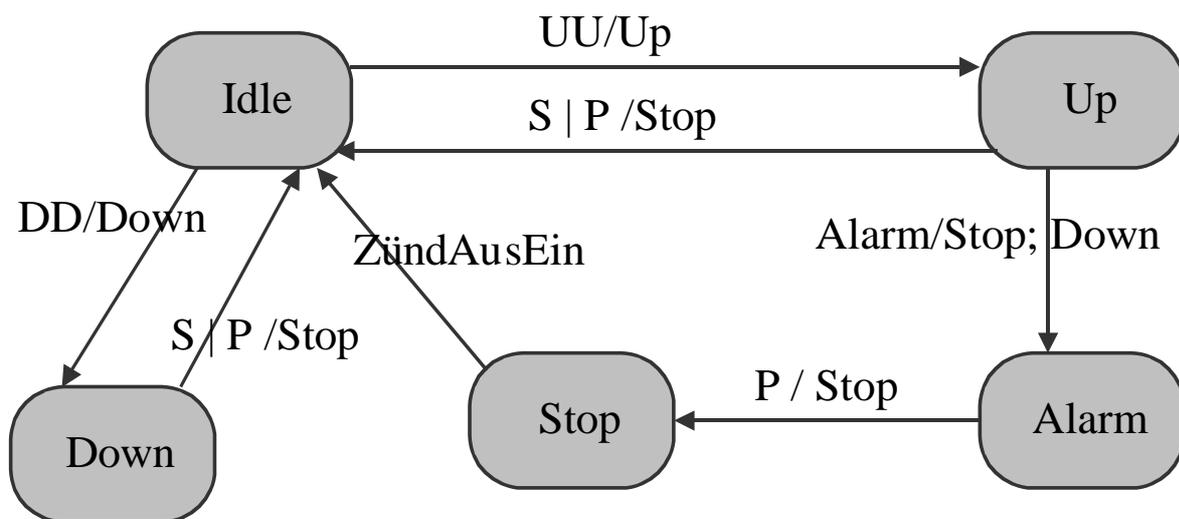
- Das Strukturmodell zeigt die Gliederung in Komponenten



Interaktionsdiagramm Fensterheber: Fall Einklemmschutz



Zustandssicht: Steuergerät Fensterheber



- Betrachten wir ein System mit seiner Umgebung oder ein verteiltes System, so können wir den Nachrichtenaustausch zwischen dem System und seiner Umgebung durch die Darstellung der Interaktion beschreiben.
- Zur Darstellung der Interaktionssicht bieten sich Interaktionsdiagramme an. Diese beschreiben exemplarisch in Szenarien Interaktionsmuster für das System.
- Die Interaktionssicht nehmen wir ein, um
 - ◇ Die Interaktion zwischen einem System und seiner Umgebung
 - ◇ Die Interaktion zwischen den Komponenten eines Systems

zu beschreiben

Vorteile der Modellierung

Eine Modellbildung auf RE Ebene erlaubt:

- Präzise Strukturen zur Beschreibung der Anforderungen
- Überprüfung der informellen Anforderungsdokumente auf Vollständigkeit und Konsistenz („Anforderungvalidierung“)
- Überprüfung der verhaltensorientierten Anforderungsmodellierung auf wesentliche Eigenschaften („Anforderungsverifikation“)