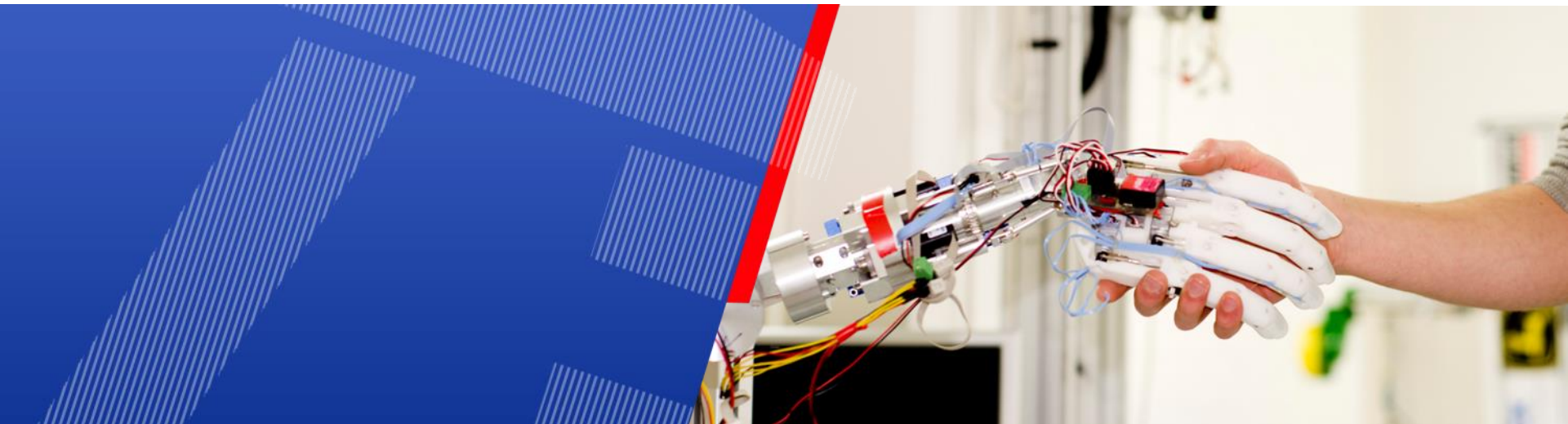


Roboterarchitektur



Prof. Karsten Berns

Robotics Research Lab

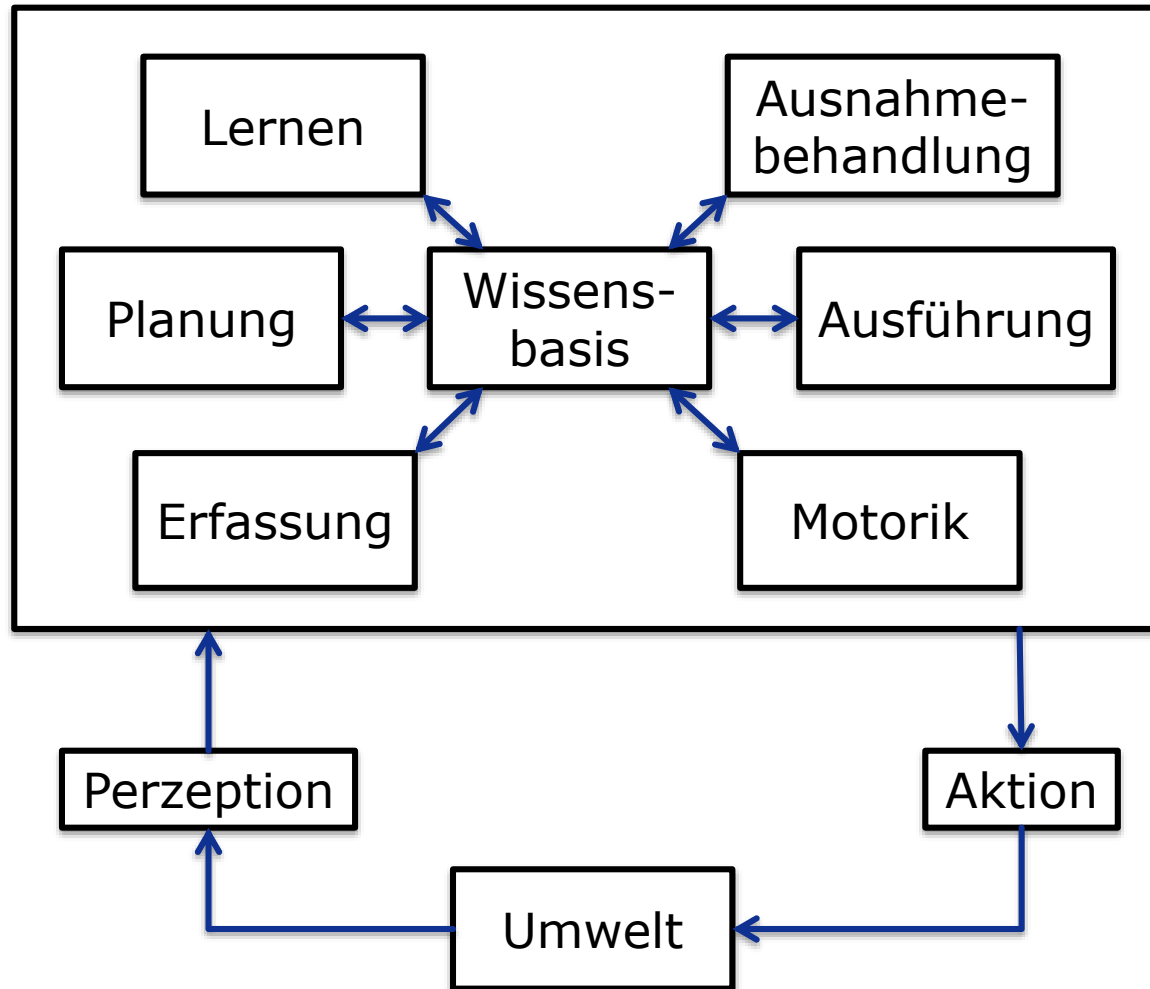
Department of Computer Science

University of Kaiserslautern, Germany

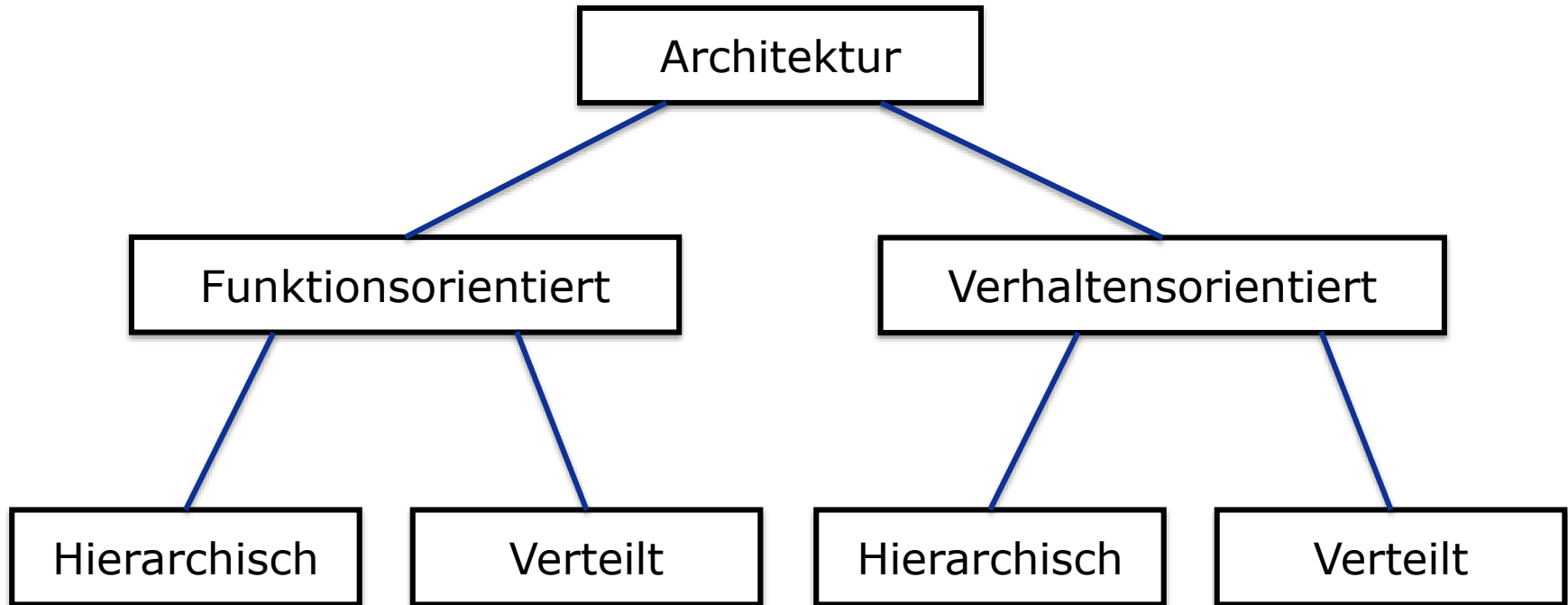
Inhalt

- Grundlegende Fähigkeiten eines Robotersystems
- Schematische Darstellung der 4 Basisarchitekturen
- Hierarchisch funktionsorientierte Architekturen
- Verteilte funktionsorientierte Architekturen
- Hierarchisch verhaltensorientierte Architekturen
- Verteilt verhaltensorientierte Architekturen

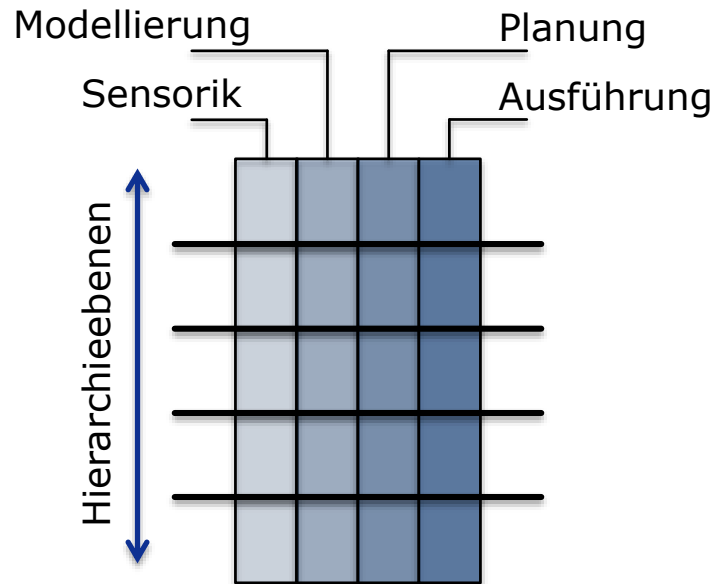
Grundlegende Fähigkeiten eines Robotersystems



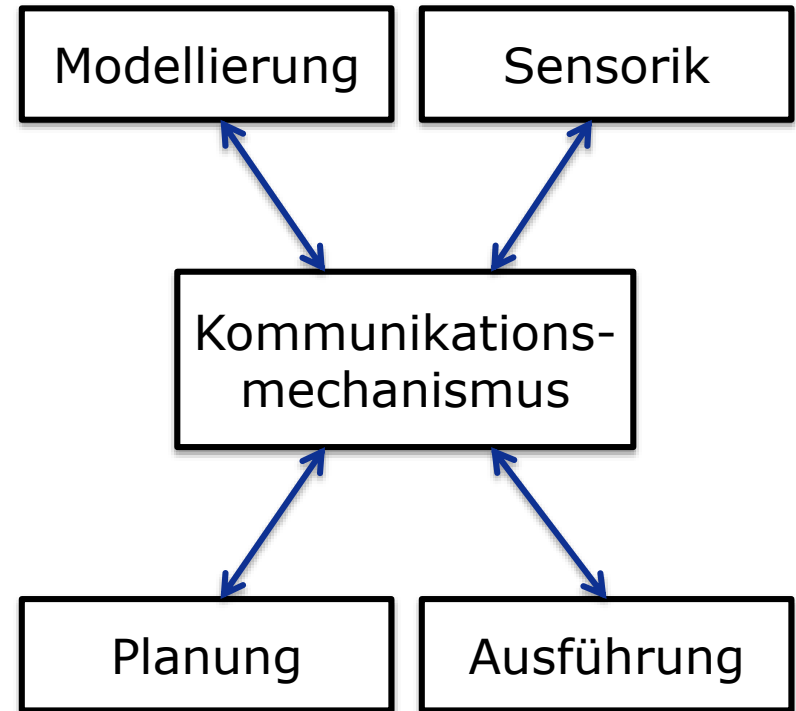
Basisarchitekturen: Klassifikation



Basisarchitekturen: Schematische Darstellung

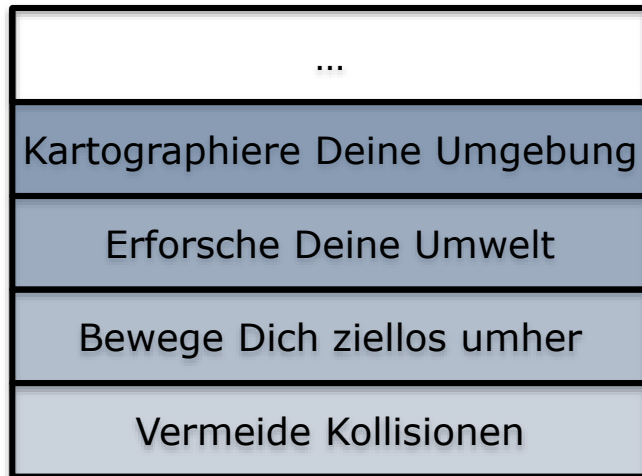


Hierarchisch funktionsorientiert

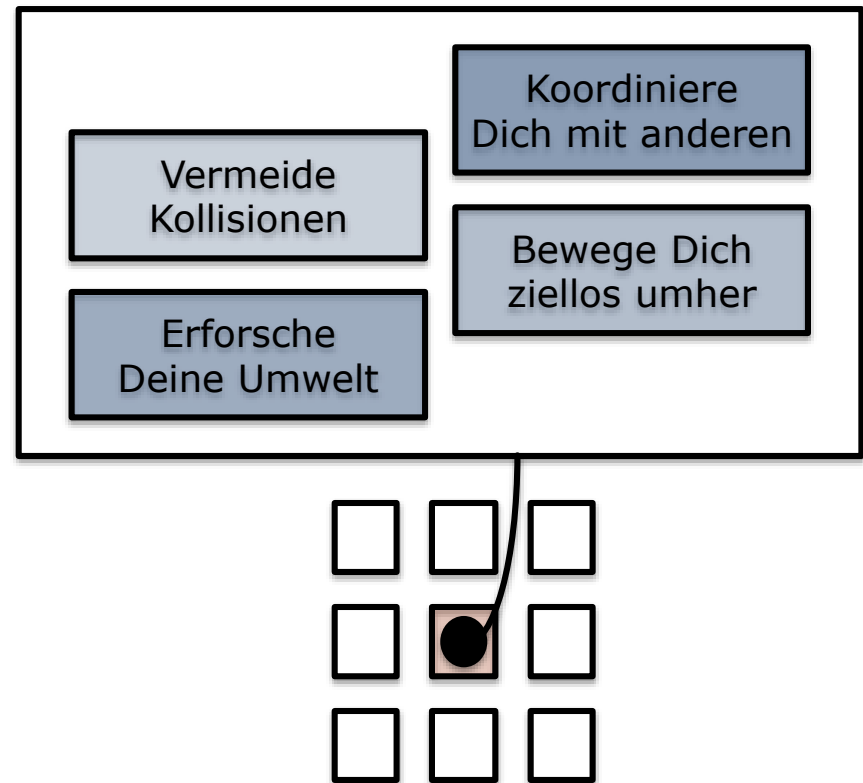


Verteilt funktionsorientiert

Basisarchitekturen: Schematische Darstellung



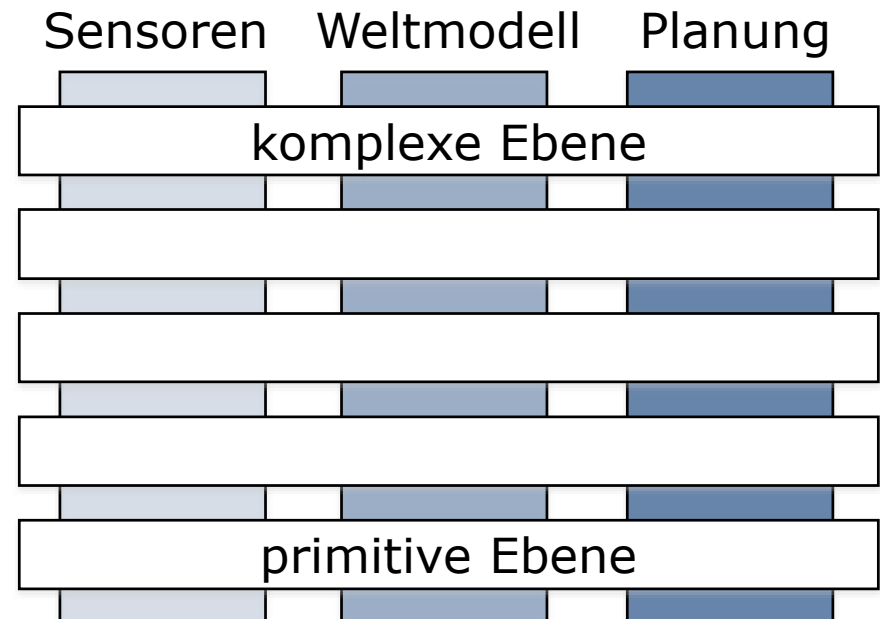
Hierarchisch verhaltensorientiert



Verteilt verhaltensorientiert

Hierarchisch funktionsorientierte Architekturen

- RCS: Real-time Control System
- Referenz-Arch. zur Entwicklung intelligenter Kontrollsysteme
- Kombination aus planenden und reaktiven Kontrollstrukturen
- Hierarchisch geordnete Menge von Knoten
- Beispiel: NASREM-Modell
(Albus, J. S.; McCain, H. G.; Lumini, R; "NASA / NBS Standard Reference Model for Telerobot Control System Architecture" NBS Technical Note 1235, 1987)



Hierarchisch funktionsorientierte Architekturen

- Aufteilung in 4 bis 6 Ebenen
- Je Ebene 3 Module
 - Sensorverarbeitungsmodul G_i
 - Weltmodell- und Referenzdatenmodul M_i
 - Taskzerlegungs-, Planungs- und Ausführungsmodul H_i
- Module sind durch Ebeneneinteilung hierarchisch geordnet

Module

- Planungs-Modul
 - Planung und Überwachung der Aktionsausführung
 - Beachtet Daten aus Weltmodell- und Referenzdaten-Modul
- Sensorverarbeitungs-Modul
 - Verarbeitung und Filterung von Sensordaten
 - Vergleich von Beobachtungen mit internem Weltmodell
 - Ermittlung von Abweichungen
 - Erkennung von Ereignissen, Objekten und Situationen

Module

- Weltmodell-Modul
 - Aktualisierung der Referenzdaten durch Sensordaten
 - Vorhersage von Sensorergebnissen
 - Simulation von Ergebnissen von hypothetischen Plänen
 - Informationen über die Welt
(Variablen, Objekte, Regeln, Karten)
- Benutzerschnittstelle
 - Interaktion mit funktionellen Prozessen oder Wissensbasis

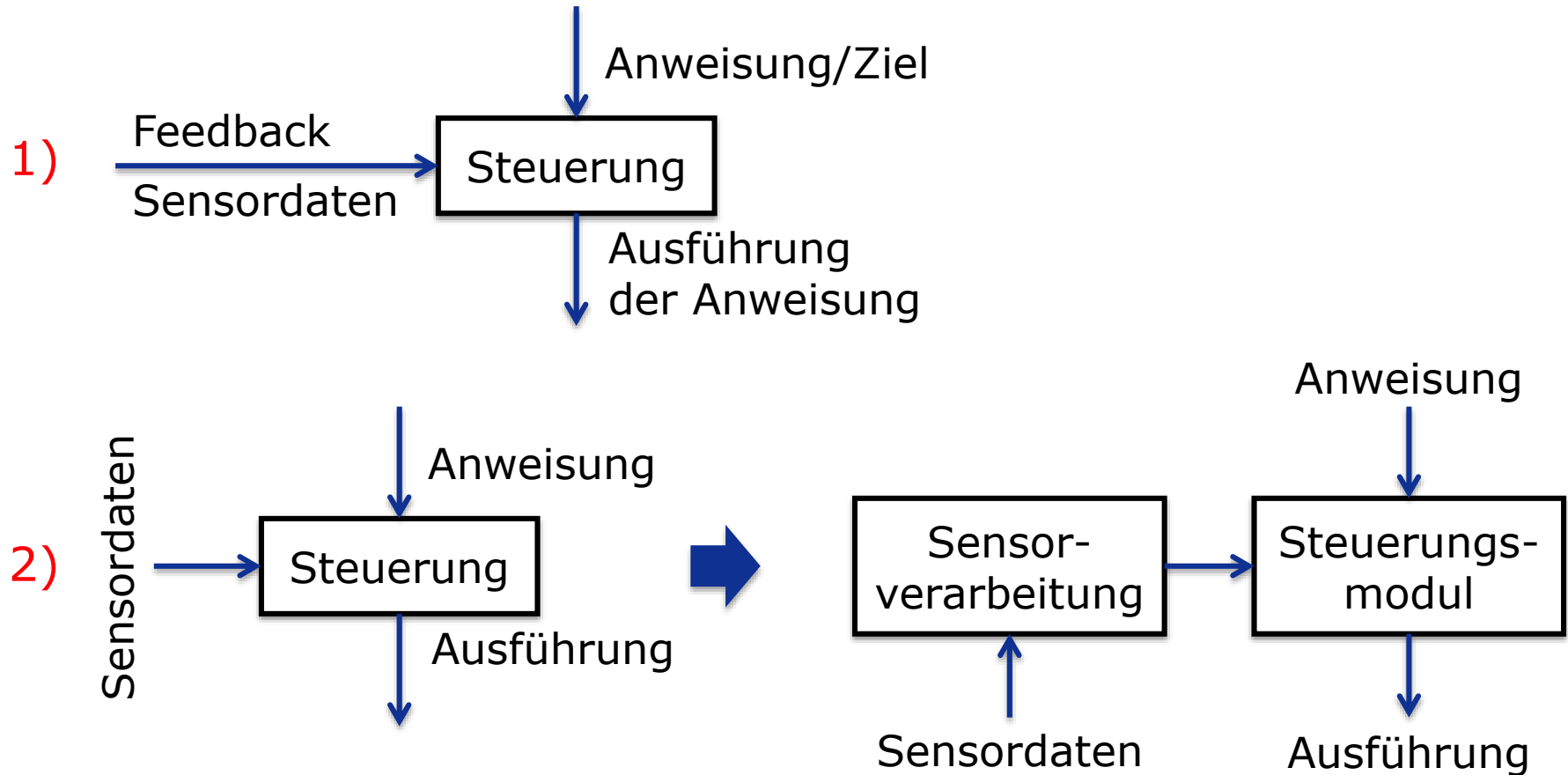
Module: Kommunikation zwischen Modulen

- *H*-Module: Kommandos zu *H*-Modul der tieferen Ebene
- *G*-Module: Status/Sensorinformationen zur höheren Ebene
- Innerhalb der Ebene: Zwischen *G*- u. *M*- sowie *M*- u. *H*-Modul
- Keine speziellen Kommunikationsmechanismen
- Beispiele
 - Point-to-Point-Verbindungen
 - Netzwerkverbindungen
 - Gemeinsamer Speicher

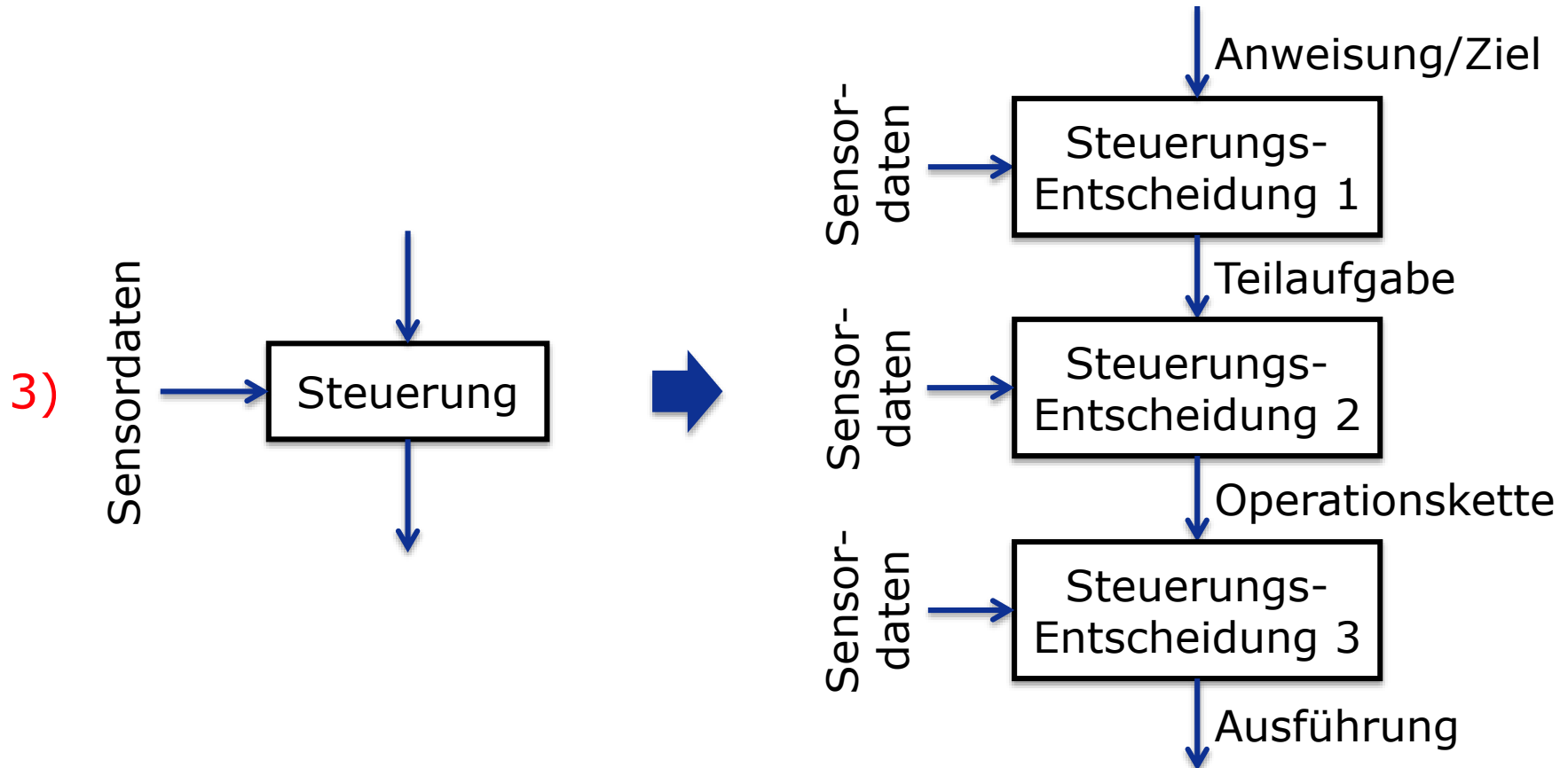
Knoten im RCS

- Implementierung der Funktion innerhalb der Knoten (z.B. als erweiterter endlicher Automat)
- Definierter Anfangszustand
- Eingaben aus min. einem Eingabepuffer
- Zustandswechsel basierend auf gelesenen Informationen
- Zustandsübergang löst Aktionen aus
- Speicherung der Ausgaben in Ausgabepuffern
- Kommunikationssystem verteilt Informationen aus Ausgabepuffern in entsprechende Eingabepuffer
- A-/Synchrone Steuerung der endlichen Automaten
- Verarbeitungsfrequenz abhängig von Ebene

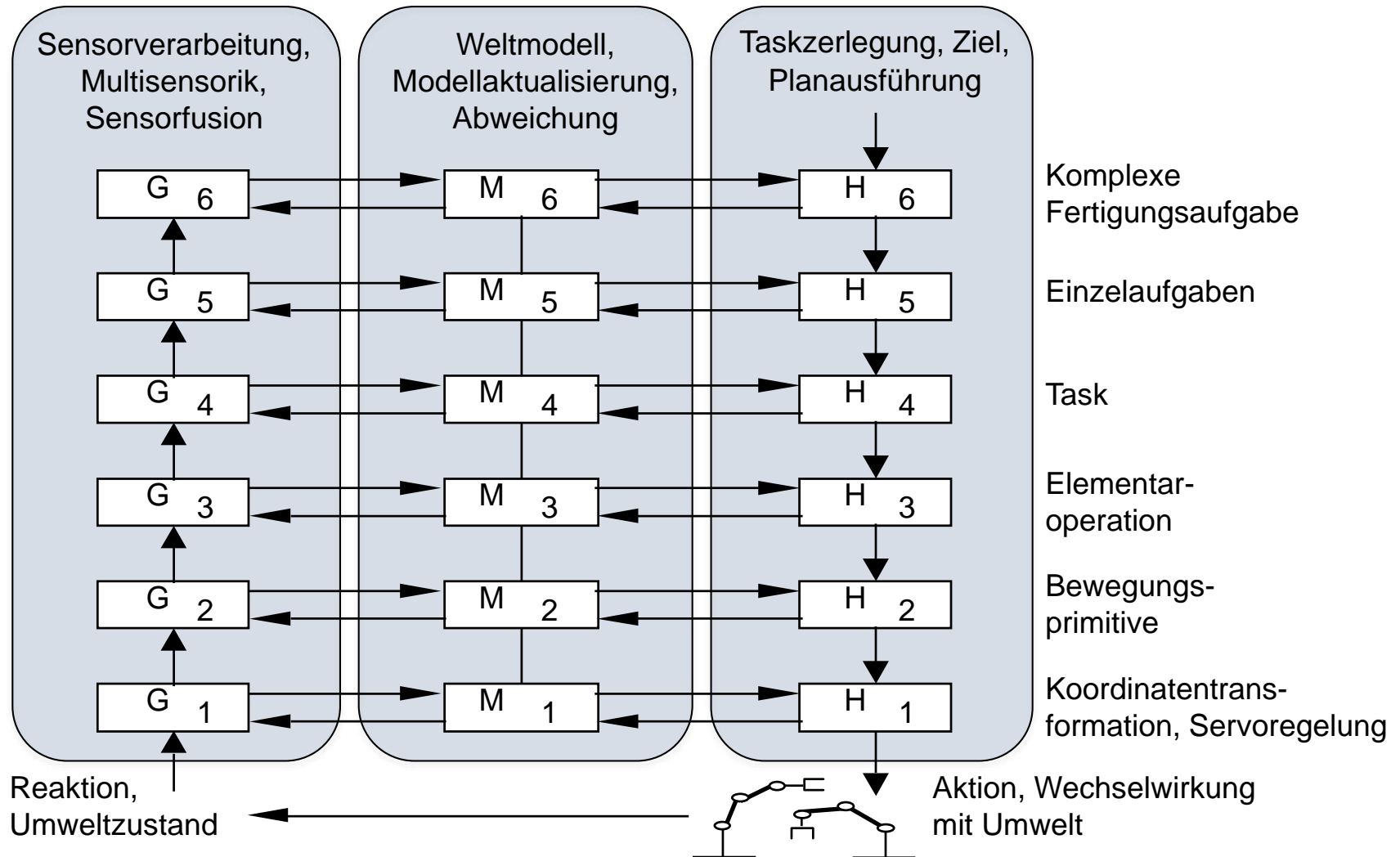
Hierarchisch gegliederte Teilfunktionen



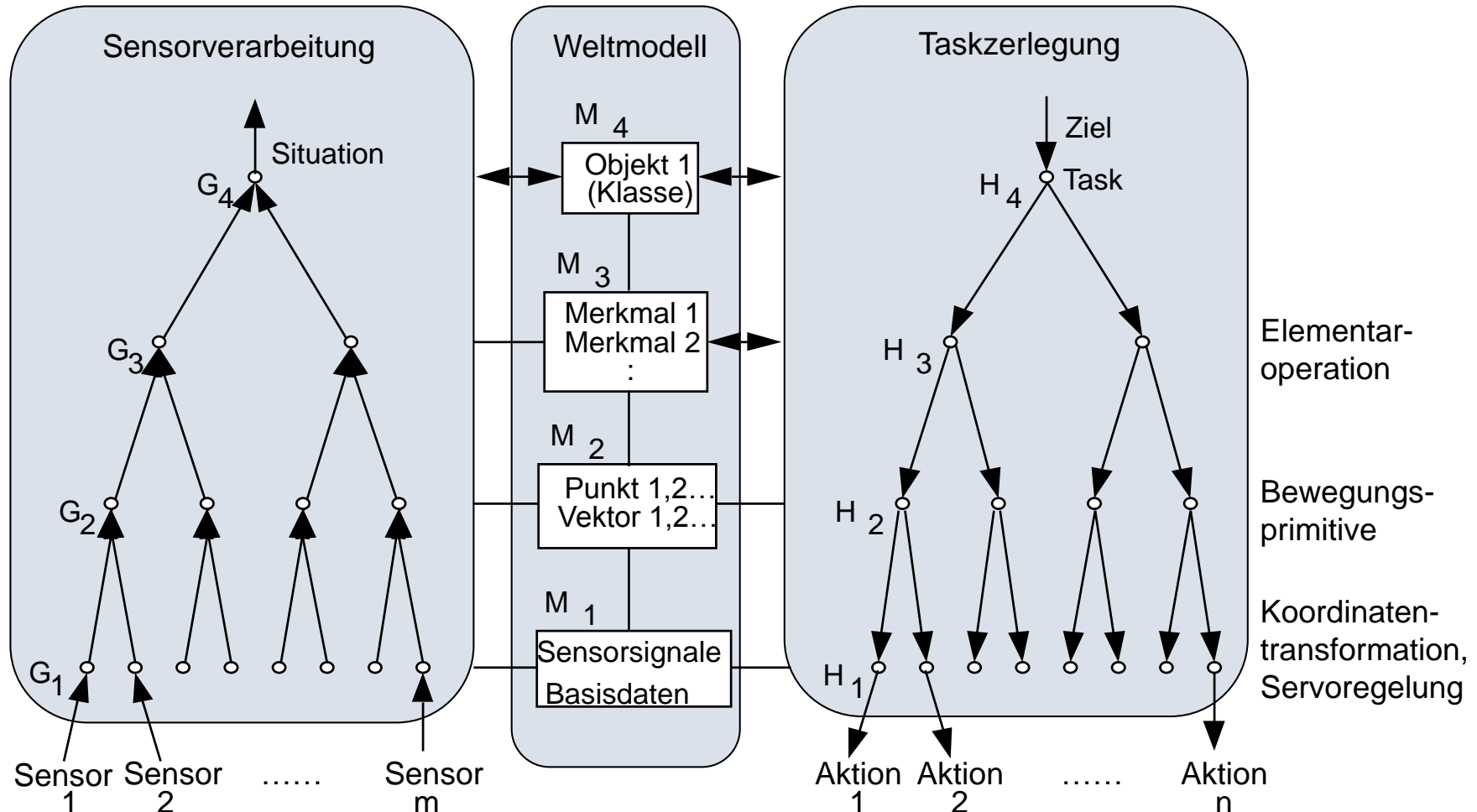
Hierarchisch gegliederte Teilfunktionen



Module einer Roboterarchitektur

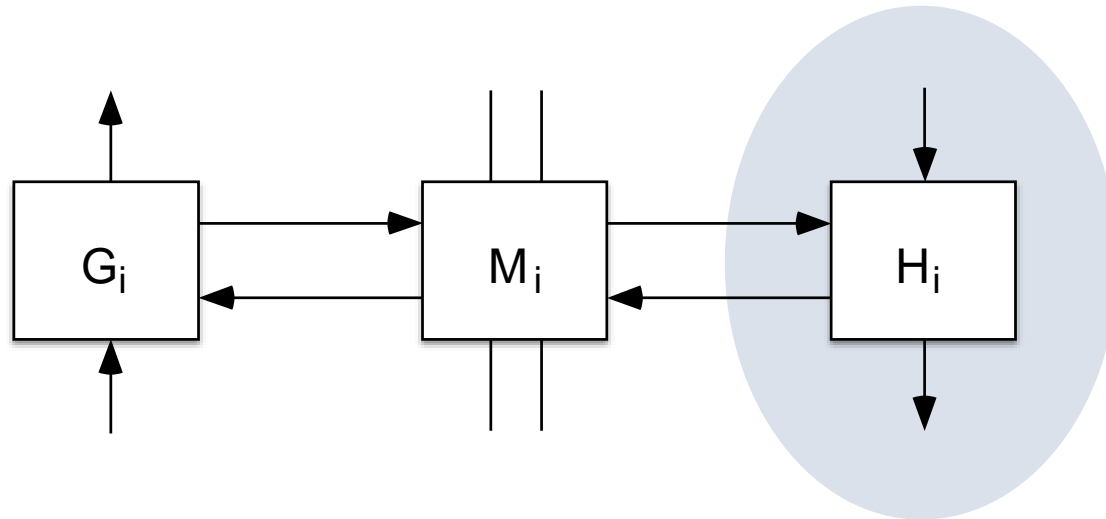


Sensorverarbeitung, Planung und Taskzerlegung

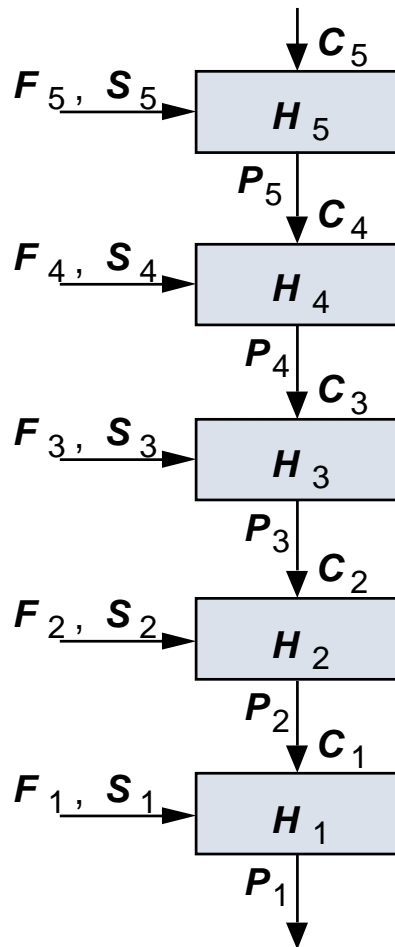


H-Modul

- Erhält Aufgabe aus der oberen Ebene
- Zerlegt Aufgabe in Teilaufgaben mit Daten aus M -Modul
- Aktualisiert M -Modul
- Teilaufgaben werden (einzeln) an untere Ebene gegeben

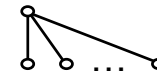


H-Modul: Sensorverarbeitung, Planung und Taskzerlegung



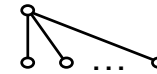
Komplexe Aufgabe

C_5 : Verbinde Teil A mit Teil B



Taskebene

C_4 : Greife Teil B



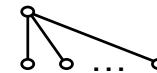
Einfache Operationen
Elementaroperationen

C_3 : Bewege Endeffektor zu
Anrückframe von Teil B



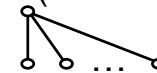
Geom. Trajektorie,
Adaption,
Kraftregelung

C_2 : Frame Teil B
Bestimme Anrückframe



Gelenk-Trajektorie,
Servoregelung

C_1 : Gelenkvektor für Roboter
(TCP in Frame B), ...

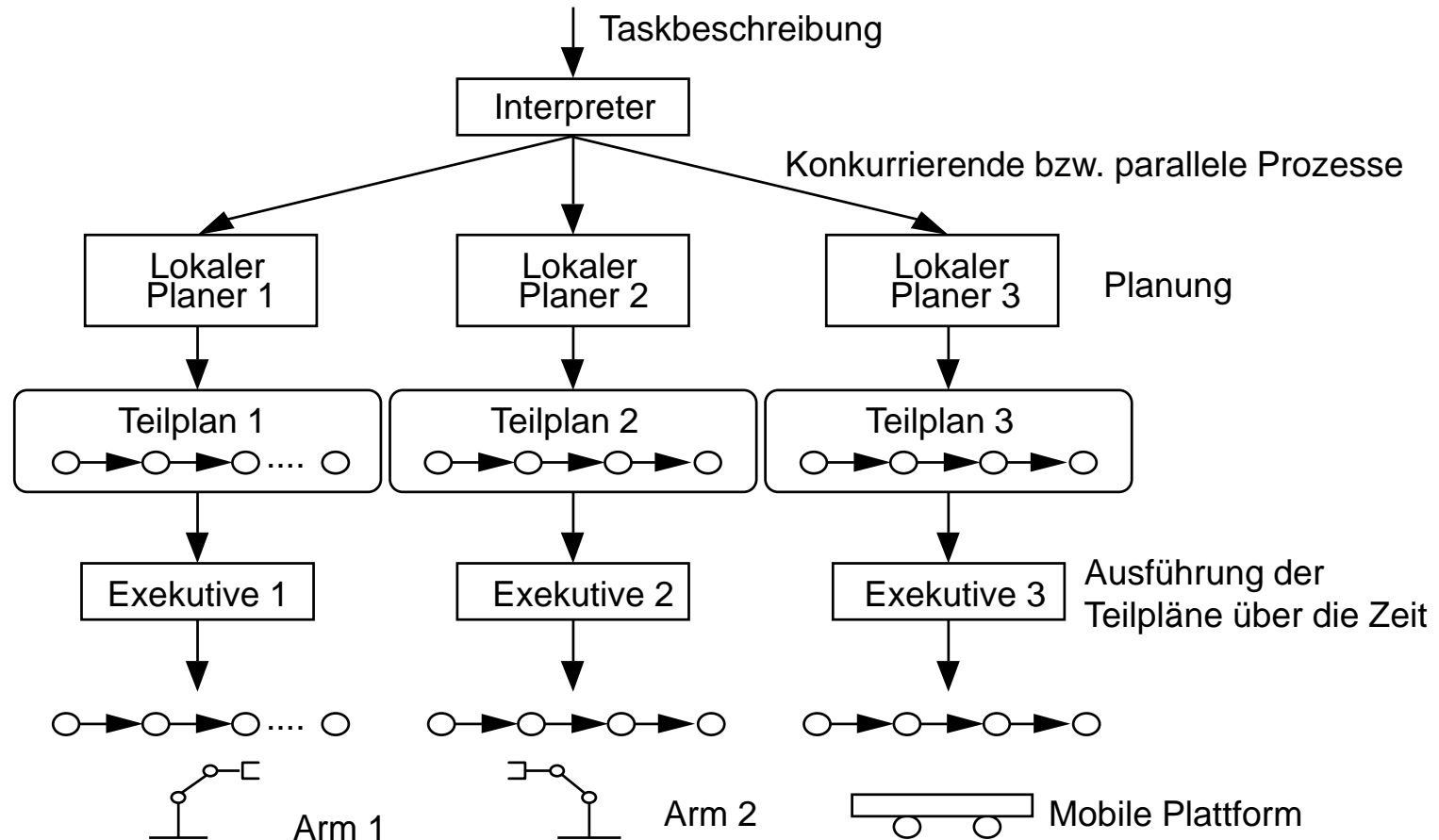


Folge von Stellmomenten

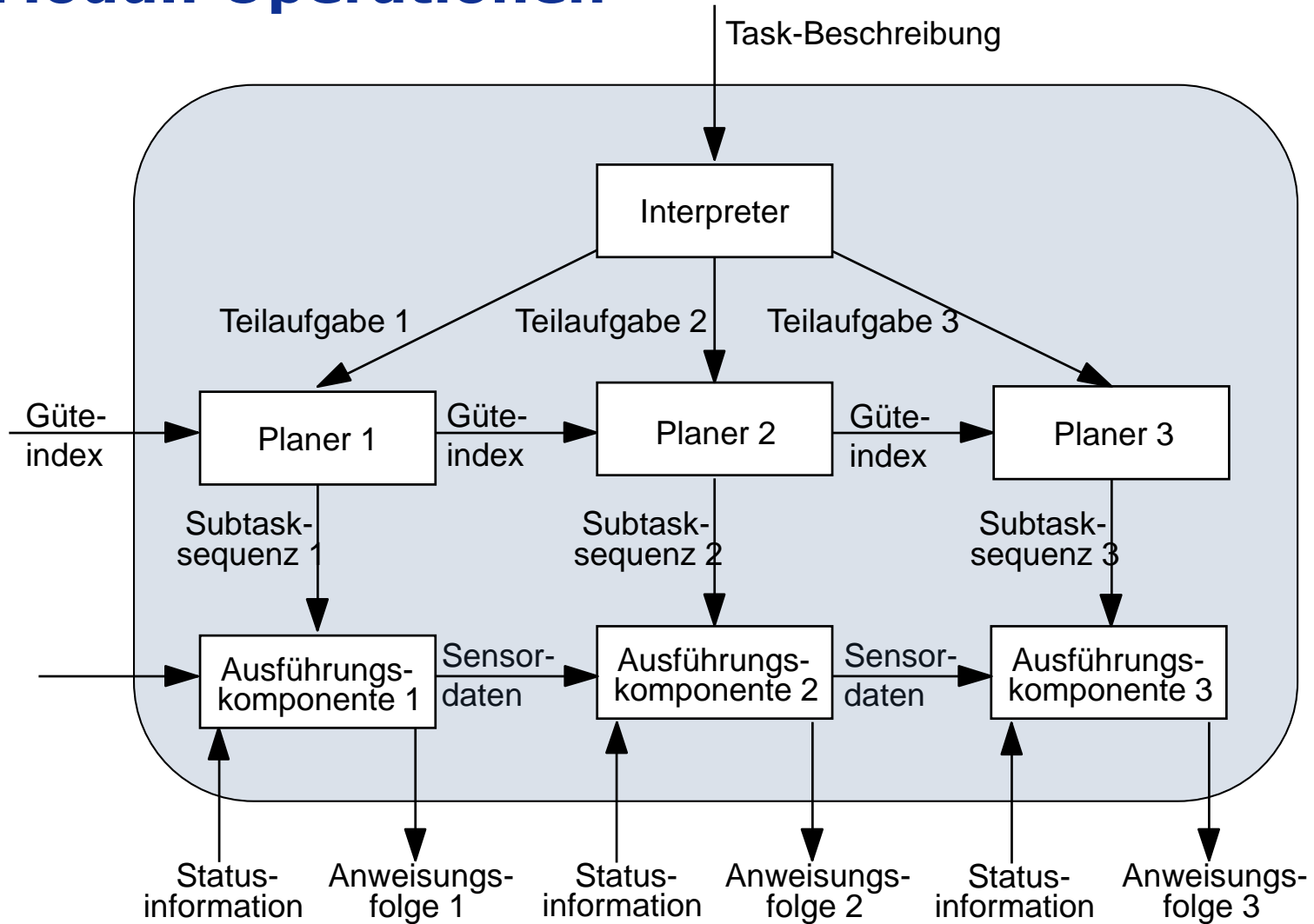
H-Modul: Taskzerlegung über Steuerungsebenen

- Ausgabevektor $P_i \sim C_{i-1}$
- Anweisungsvektor $C_i \rightarrow$ Folge von P_{i+1}
- Sensorvektor F_i
- Eingabevektor $S_i = C_i + F_i$
- Operator H_i
 - $P_i^k = H_i(s_i^{k-1})$
 - $P_i(t) = H_i(S_i(t - \Delta t))$

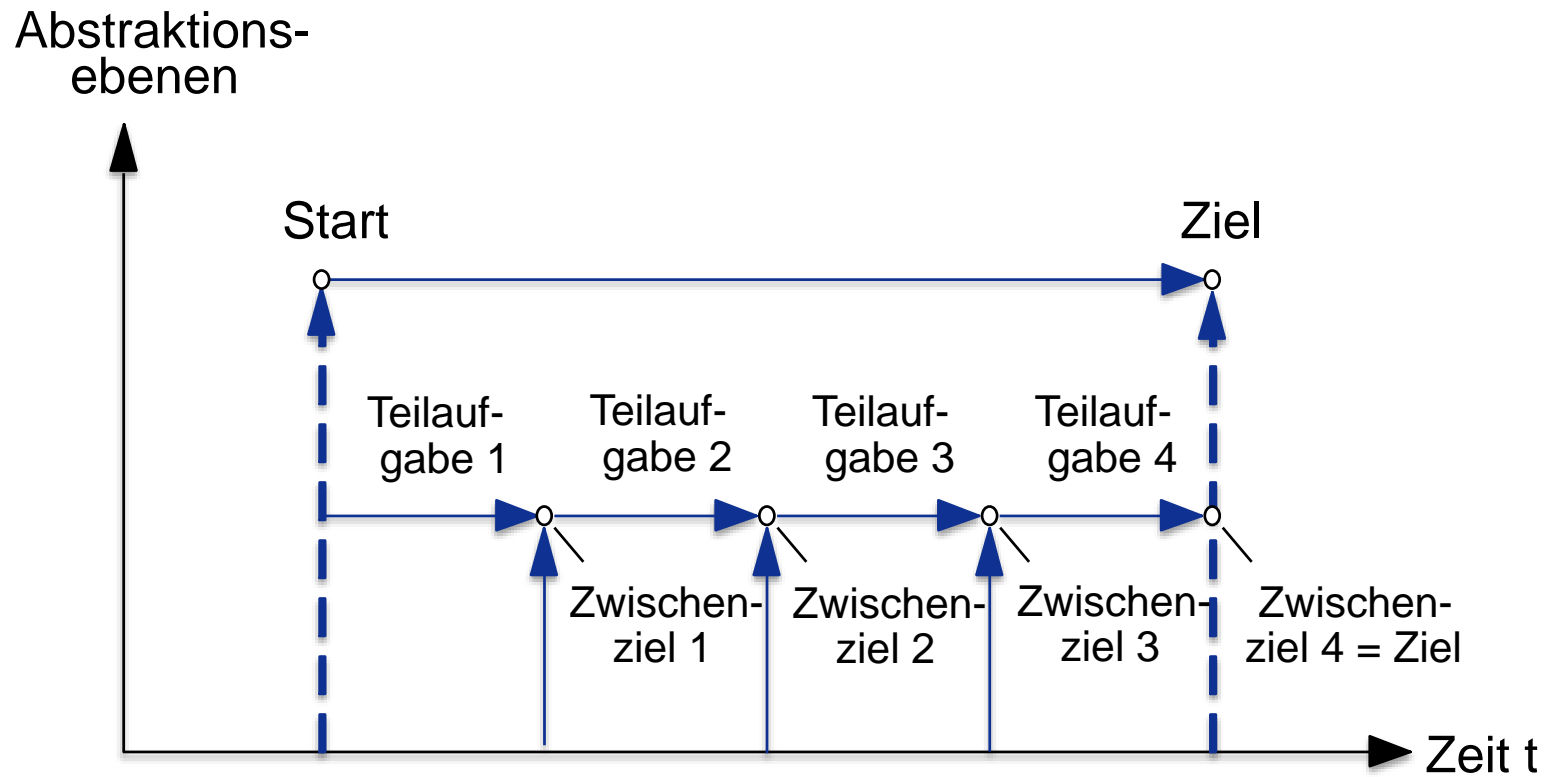
H-Modul: Taskzerlegung und Ausführung



H-Modul: Operationen

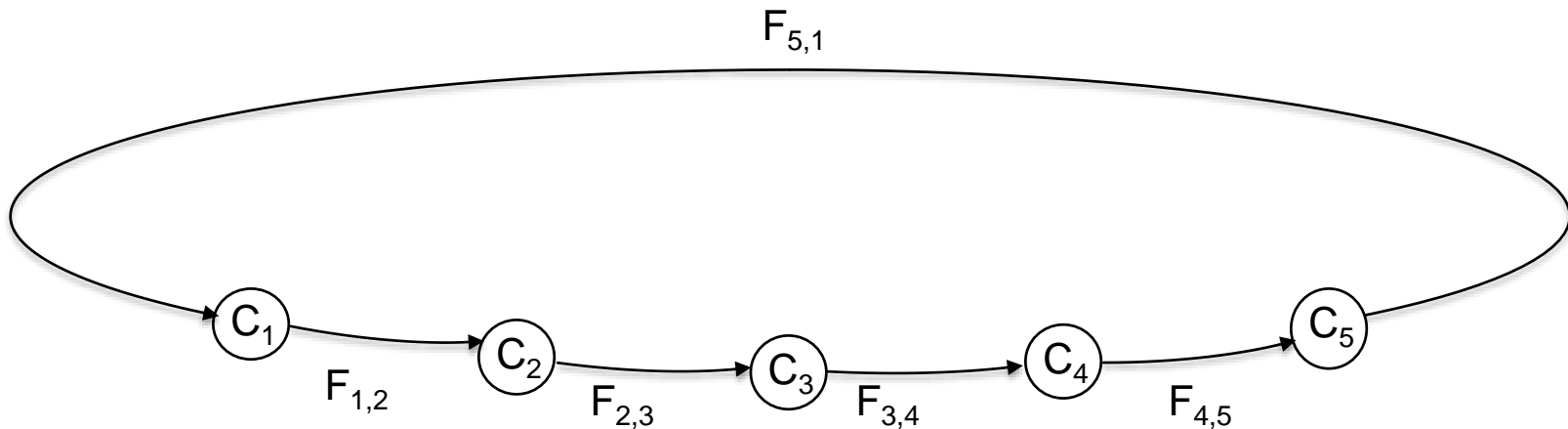


H-Modul: Aufgabenzerlegung in Teilaufgaben

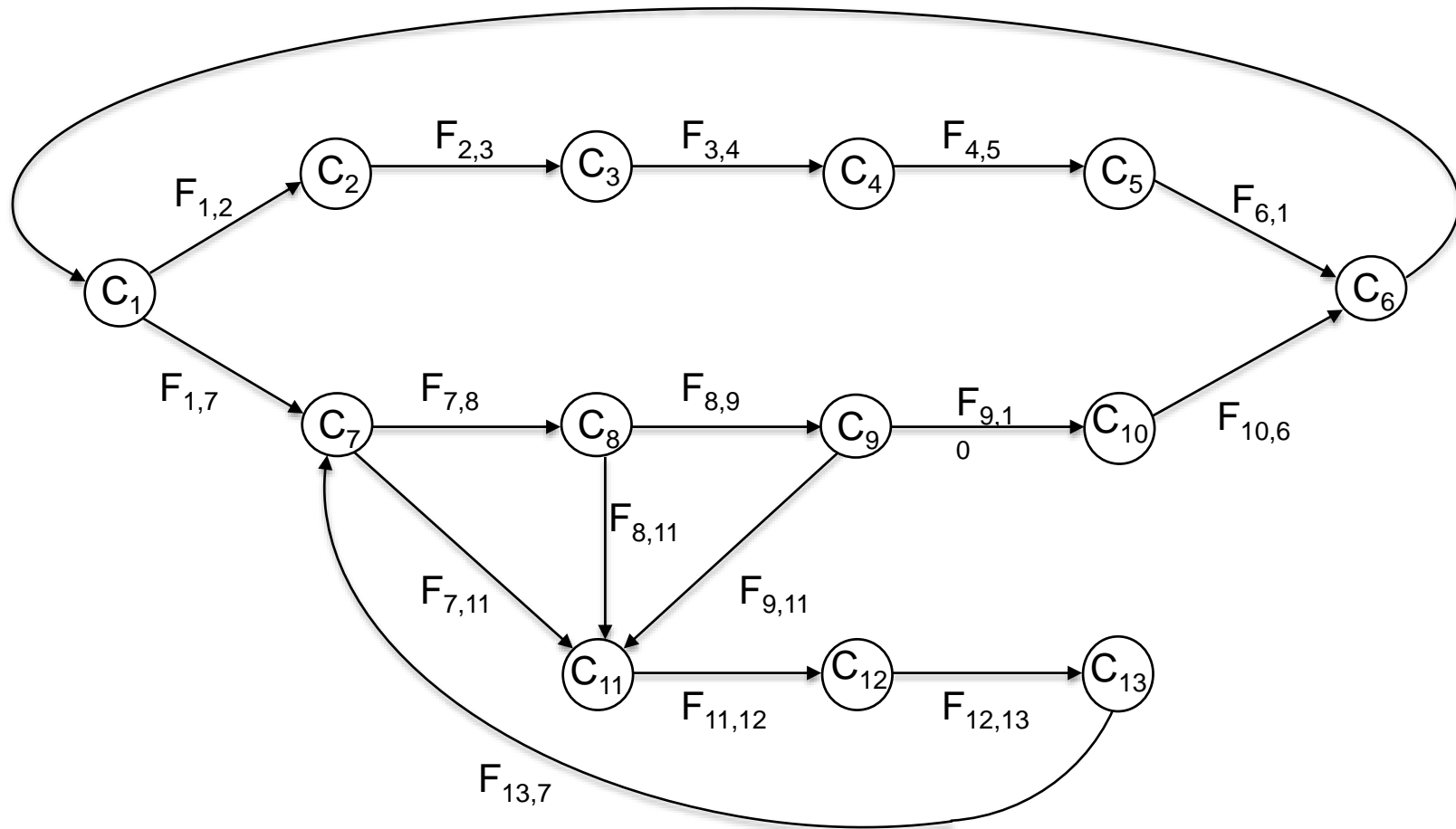


H-Modul: Eingabefolge als linearer Graph

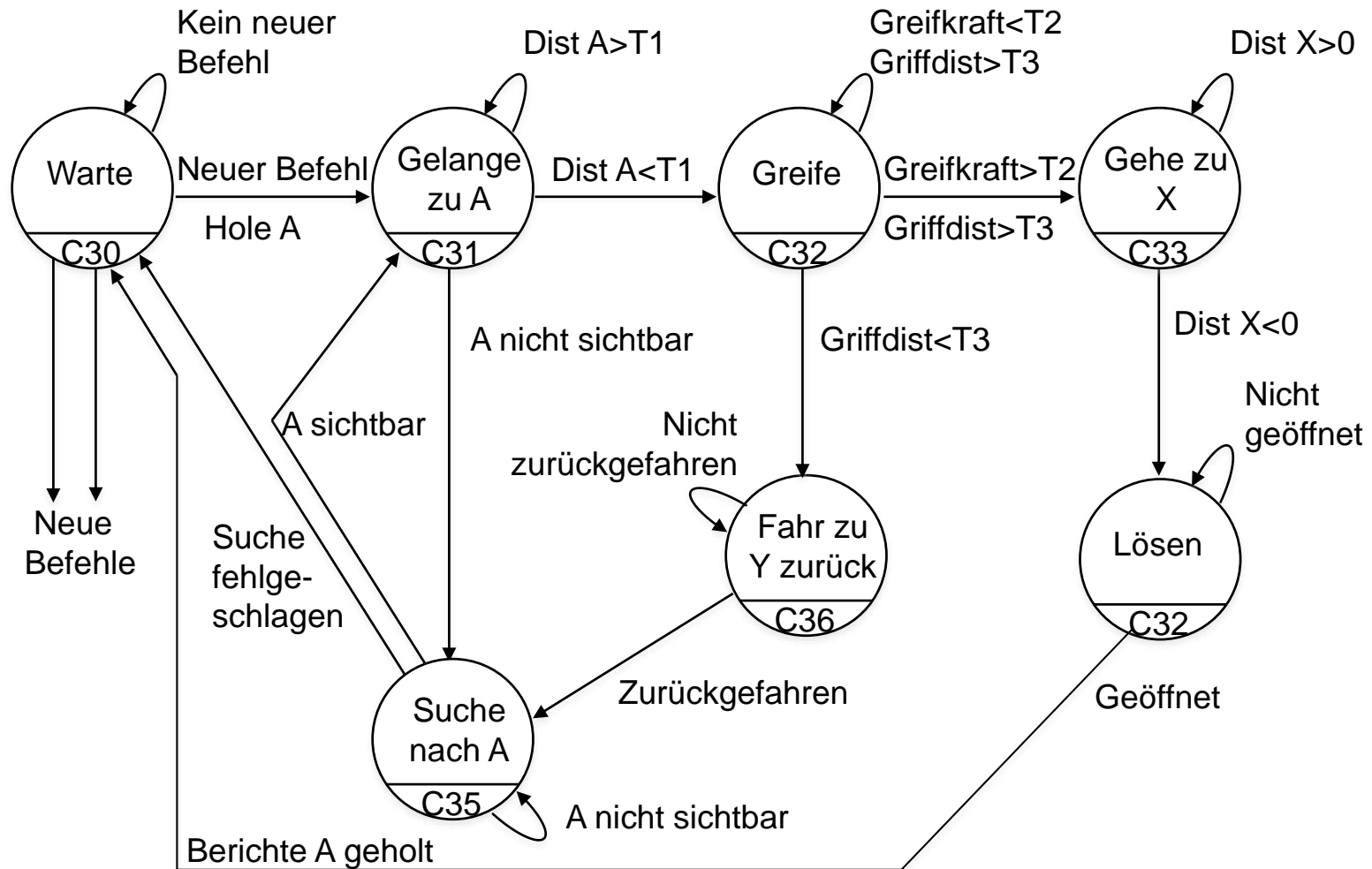
- C_i : Befehl i
- $F_{i,j}$: Transition von i nach j



H-Modul: Eingabefolge als verzweigtes Netz

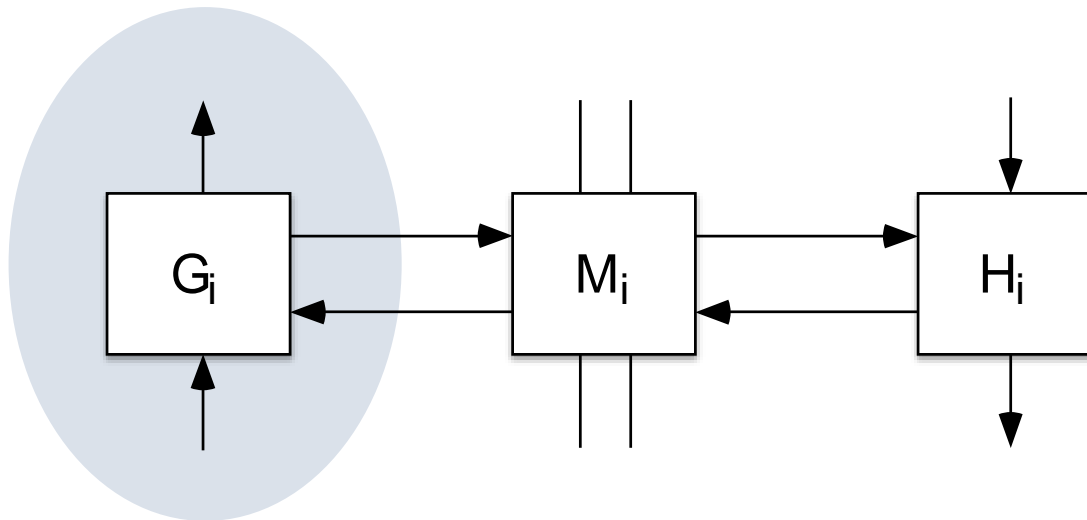


H-Modul: Beispiel Montagesequenz



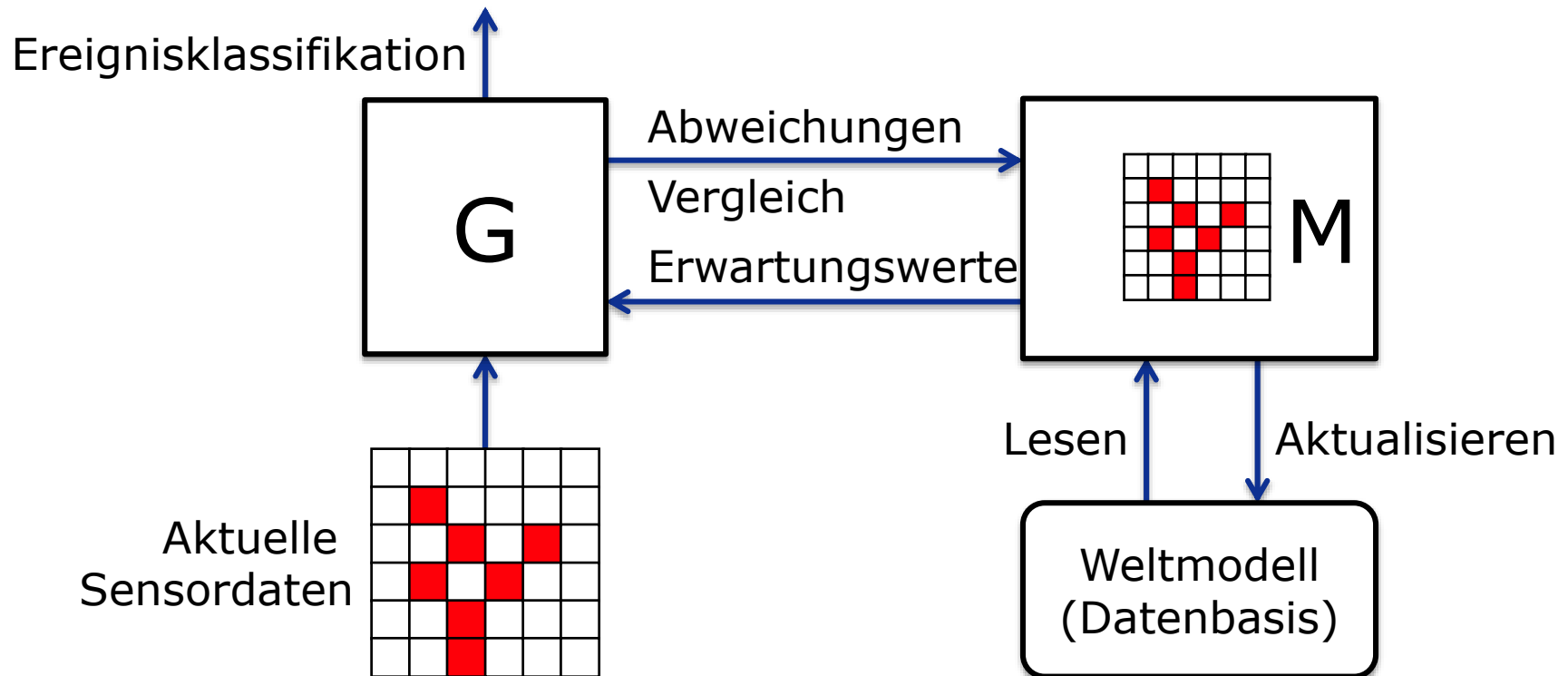
G-Modul

- Erhält Status/Sensorinformationen aus der unteren Ebene
- Verarbeitet diese Daten mit M-Modul
- Aktualisiert M-Modul
- Sendet Informationen an nächst höhere Ebene

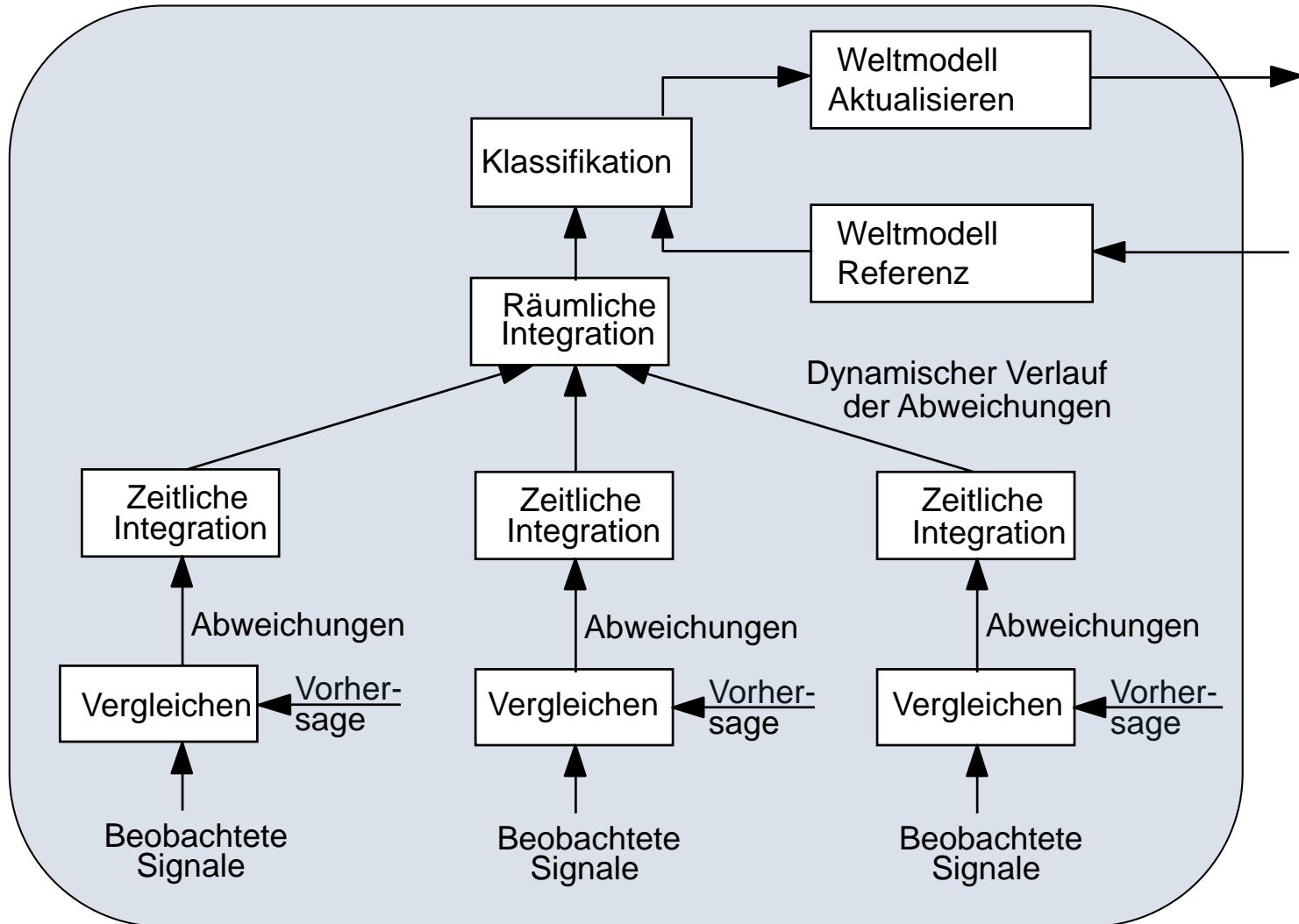


G-Modul: Vergleich Modell mit Messung

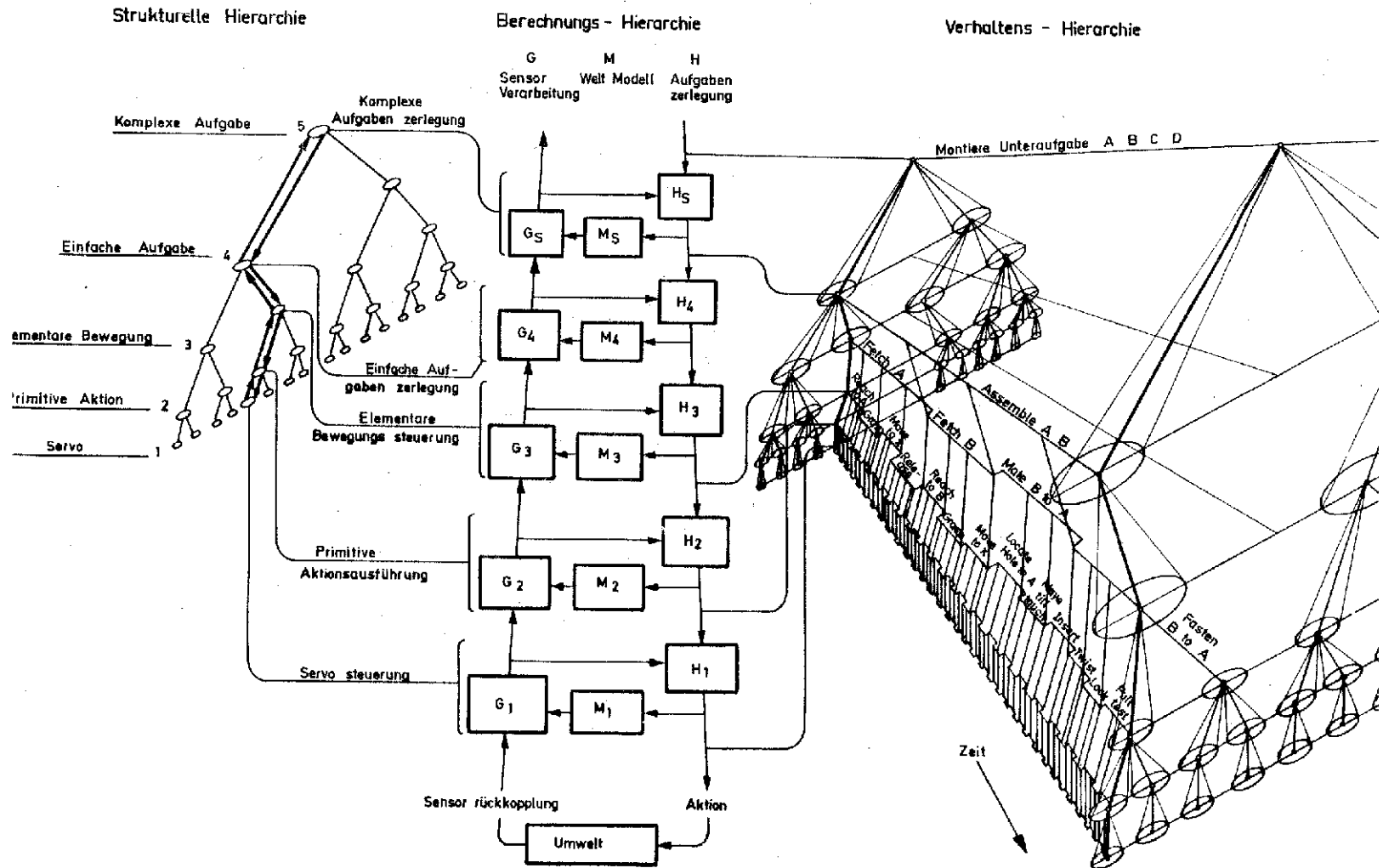
Vergleicht prädiktiv bestimmte Modellzustandsdaten mit gemessenen Ergebnissen



G-Modul: Signalverarbeitung

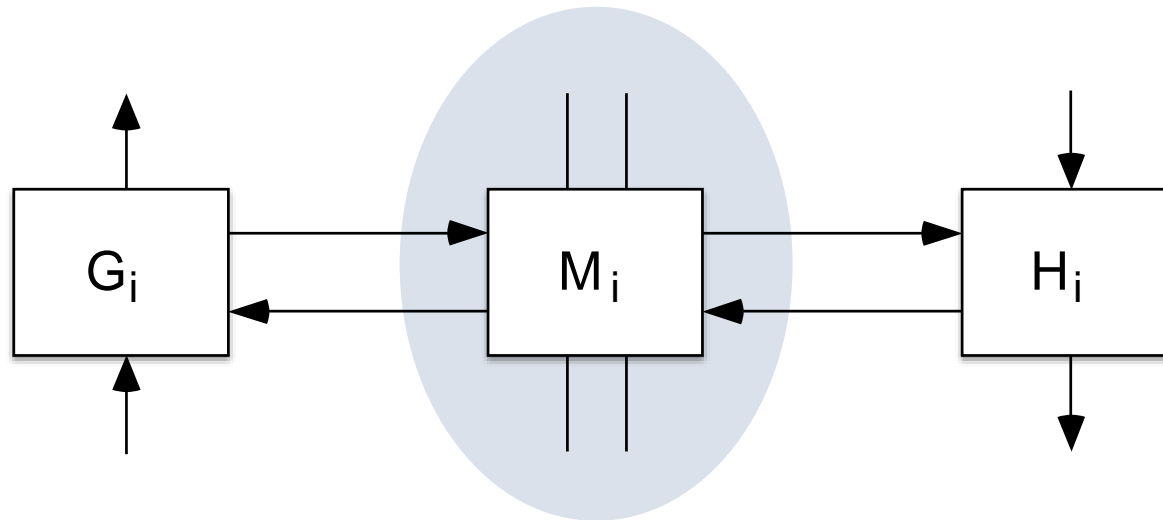


G-Modul: Anwendung

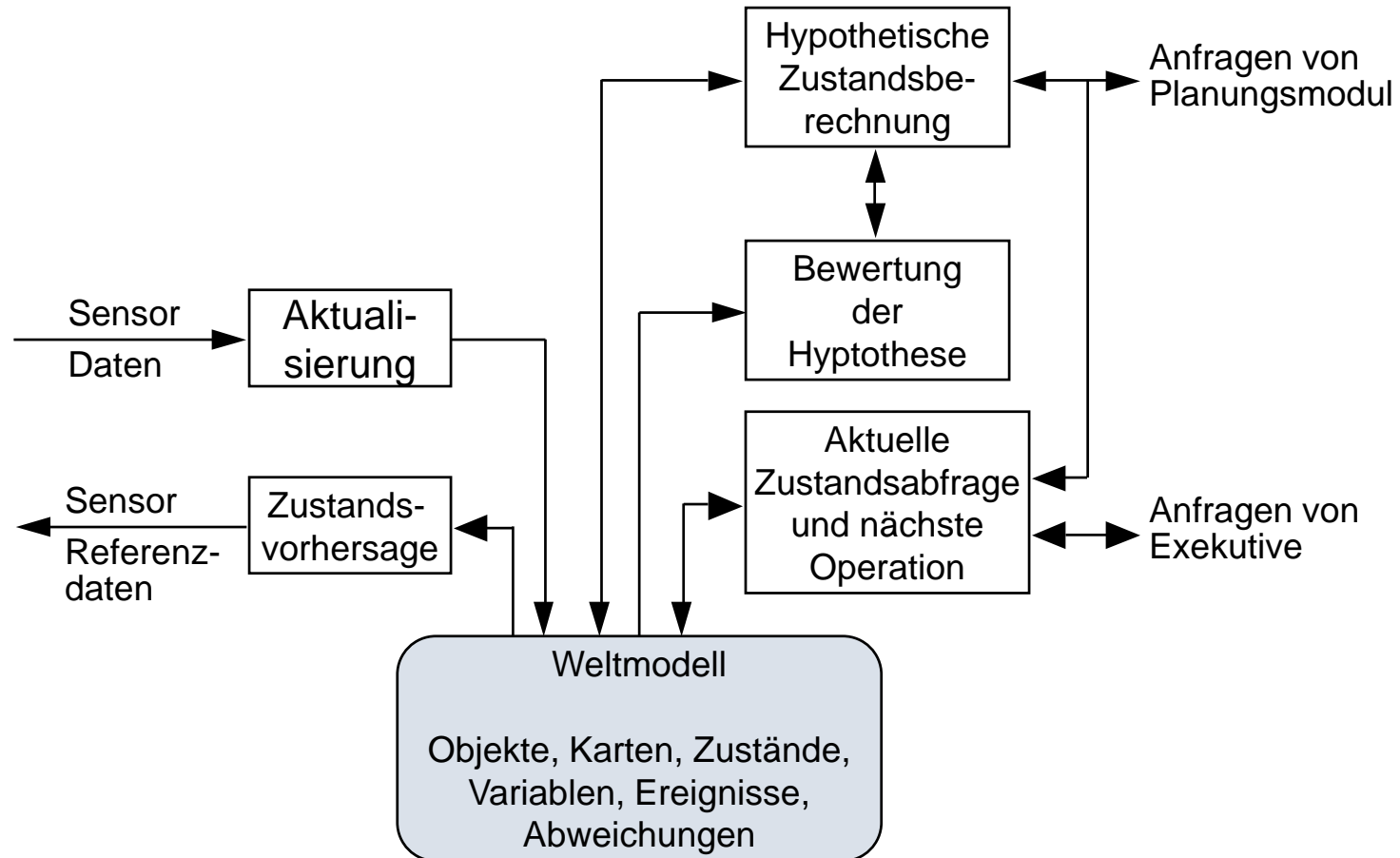


***M*-Modul**

- Beinhaltet Daten mit Abstraktionsgrad der Ebene
- *G*- und *H*-Module extrahieren diese Daten
- Fehlererkennung durch Differenz zwischen Solldaten (*M*-Modul) und Ist-Daten (*G*-Modul)

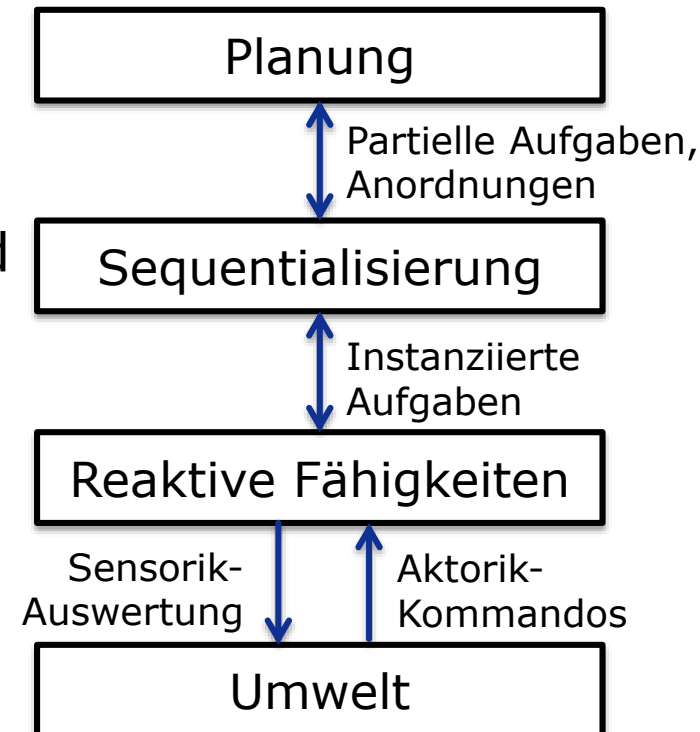


M-Modul: Operationen auf dem Weltmodell



Beispiel: T3 – Intelligent Control Architecture

- Entwickelt von NASA und Metrica Inc. Robotics and Automation Group
- Drei interagierende Ebenen
 - Dynamisch reprogrammierbare Menge reaktiver Fähigkeiten (Skills)
 - Koordination durch Skill-Manager
 - Ablaufsteuerung zur Aktivierung und Deaktivierung von Skills zur Lösung spezieller Aufgaben
 - Reactive Action Packages (RAPs)
 - Planungskomponente zur Bestimmung von Zielen, Ressourcen und Zeitvorgaben (Adversarial Planner (AP))

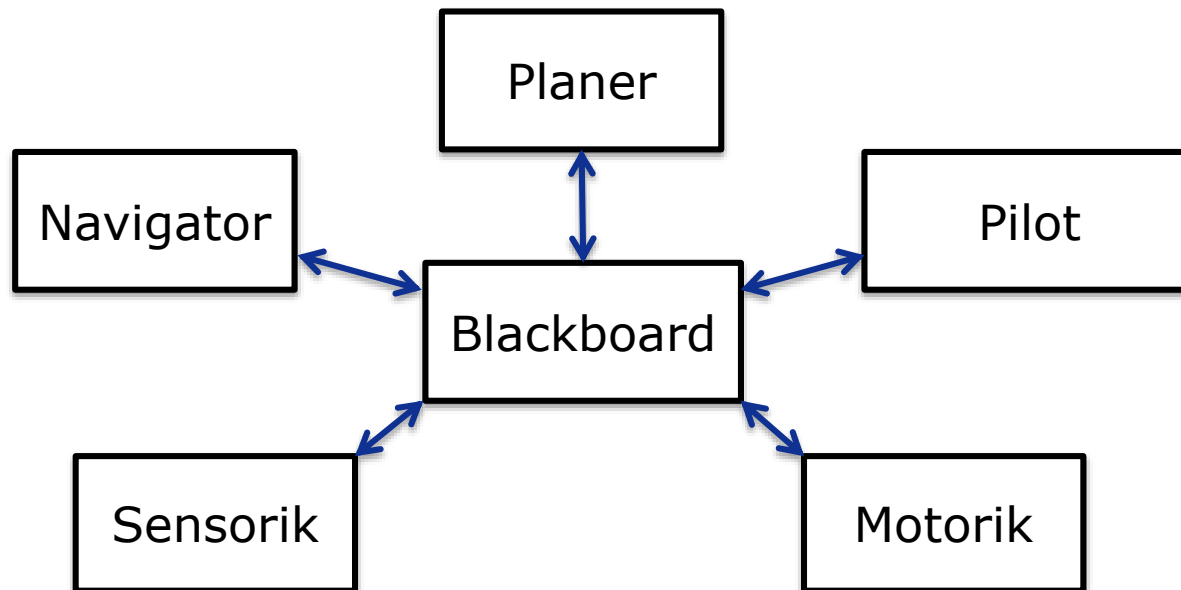


Beispiel: T3 – Ablaufsteuerung

- Aufgabenzerlegung (Folgen von Aktionen oder RAPs)
- Weitere Zerlegung und Ausführung der RAPs auf der Sequentialisierungsebene durch RAP-Interpreter
- Aktivierung von Ereignismonitoren zur Mitteilung bestimmter Ereignisse an die Sequentialisierungs-Ebene
- Aktivierte Skills bewirken das Eintreten der Ereignisse unter Berücksichtigung der Umwelt
- Sequencer-Ebene beendet Aktionen, wenn ...
 - ... überwachte Ereignisse eintreten
 - ... bestimmter Zeitrahmen überschritten wird
 - ... Planänderung von der Deliberation-Ebene gesendet wird

Verteilt funktionsorientierte Architekturen

- Menge spezialisierter Teilsysteme
- Kommunikation über eine zentrale Kompetenz
- Beispiel: Nav-Lab-System



Hierarchisch verhaltensorientierte Architekturen

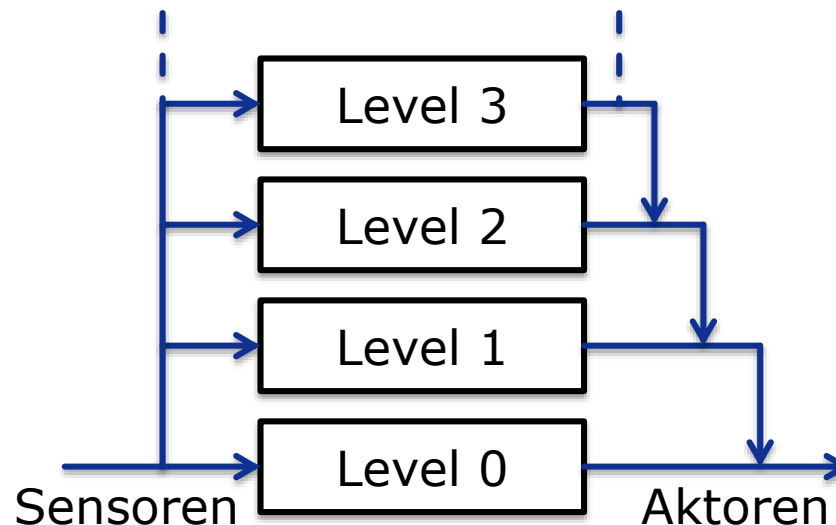
- Steuerung durch Verhaltensmuster/Reflexe
- Muster: Systemreaktion auf Sensorstimuli (Umweltsituation)
- Anordnung in Verhaltensebenen
- Hierarchische Struktur der Kompetenzebenen
- Beispiel: Subsumption Architecture,
Verhaltensschemata als endliche Zustandsautomatennetze

HV-Arch.: Anforderungen an Kontrollsysteme

- Multiple Goals: Roboter kann mehrere, sich teilweise widersprechende, Ziele verfolgen
- Multiple Sensors: Fusionierung mehrere Arten von Sensoren (Problem: Inkonsistenzen)
- Robustness: Robust gegenüber Störungen wie Sensorausfall
- Additivity: Erweiterungsmöglichkeiten von Sensoren und Rechenleistung

HV-Arch.: Subsumption Architecture

- Unterteilung der Architektur in „Levels of Competence“
- Auf jeder Ebene bildet das Verhaltenssystem eine vollständige Steuerung
- Verhalten höherer Ebenen können Ausgaben von Verhalten niedrigerer Ebenen überschreiben



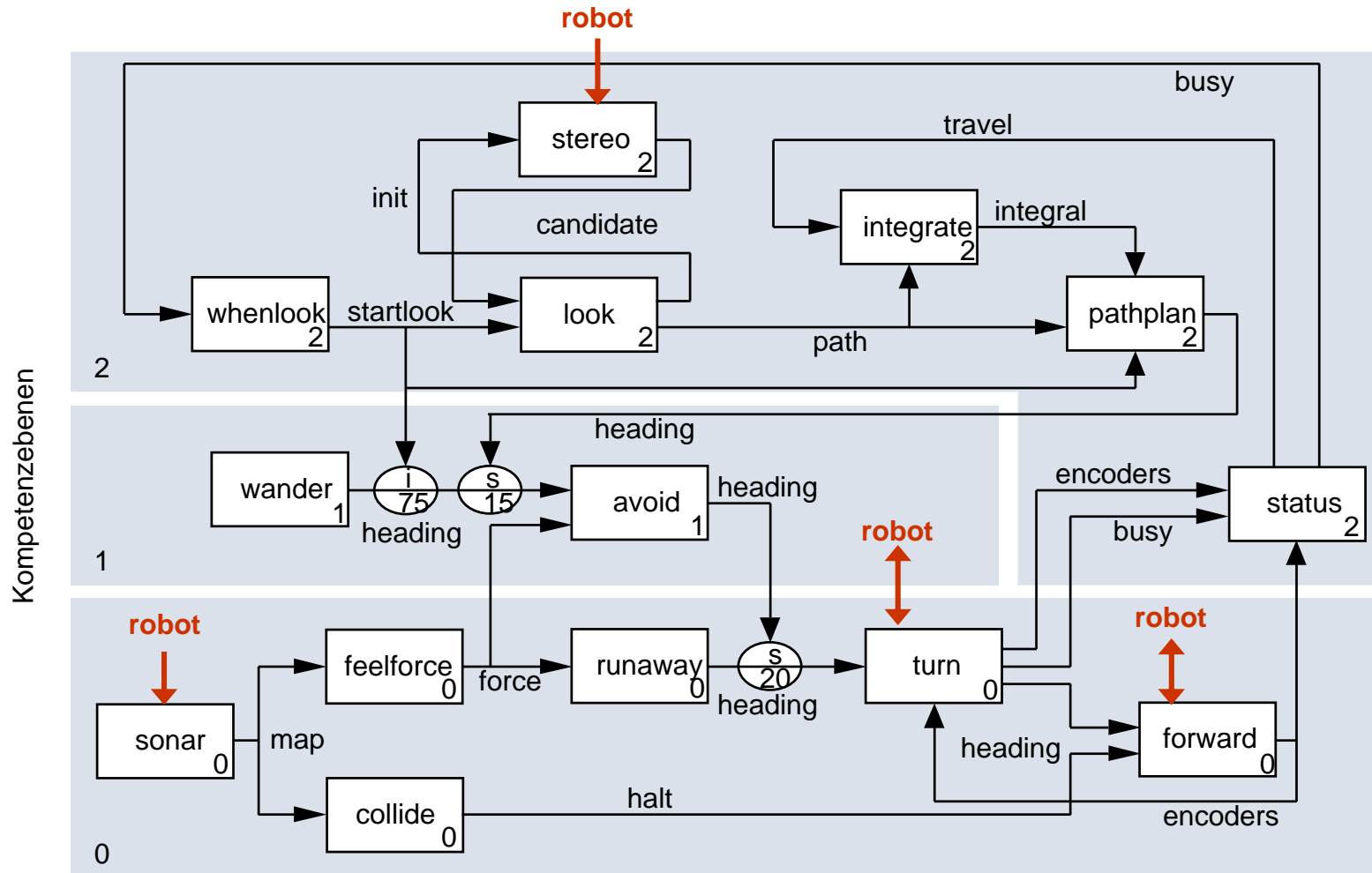
Hierarchischer Aufbau

Subsumption Architecture: Mobot-System

Hierarchische Struktur der Kompetenzebene des Roboters Mobot

1. Vermeidung von Objektkontakten
2. Ziellose Bewegung in der Umgebung
3. Erforschung der Umwelt
4. Kartenerstellung und Wegplanung
5. Erfassung von Veränderungen in der lokalen Umgebung
6. Schlussfolgern über die Umwelt im Hinblick auf erkennbare Objekte und Ausführung objektbezogener Aufgaben
7. Planerstellung und -ausführung,
welche die Umwelt in einer definierten Weise verändern
8. Schlussfolgern über das Verhalten von Objekten in der Umgebung und entsprechende Modifikation von Plänen

Subsumption Architecture: Mobot-System



Verteilte verhaltensorientierte Architekturen

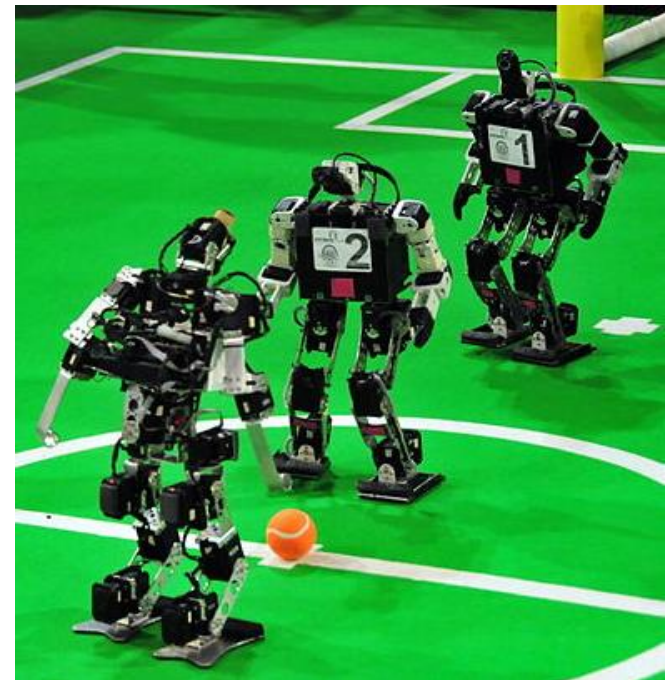
- Unabhängige Teilsystemen mit identischen („konkurrierenden“) Verhaltensmustern
- Koordination erfolgt über ein Verhaltensmuster
- Priorisierung über Arbitrierungsprozess (z.B. abklingende Aktivierung)
- Beispiel: Multi-Agenten-Systeme

Eigenschaften von Multi-Agenten-Systemen

- Selbstorganisation
- Verhandlung
- Assoziative Speicherung von Information
- Erkennung von Mustern

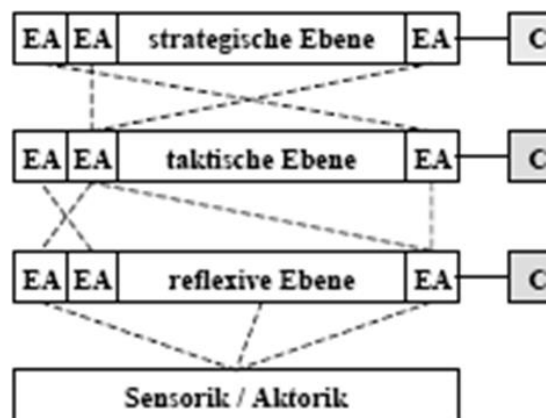
CoMRoS: Multi-Agenten-Roboter-Architektur

- CoMRoS: *CO*operative *M*obile *RO*bots Stuttgart
- Ziele
 - Universelle Architektur für autonome mobile Roboter
 - Vorhersag-/Berechenbarkeit von verteilten Verhandlungen
 - Wiederverwendbarkeit
 - Dynamische Konfigurierbarkeit
 - Fehlertoleranz
 - Sicherheit
- Szenarien
 - Formationsfahren
 - Kooperativer Transport
 - Roboterfußball (RoboCup)



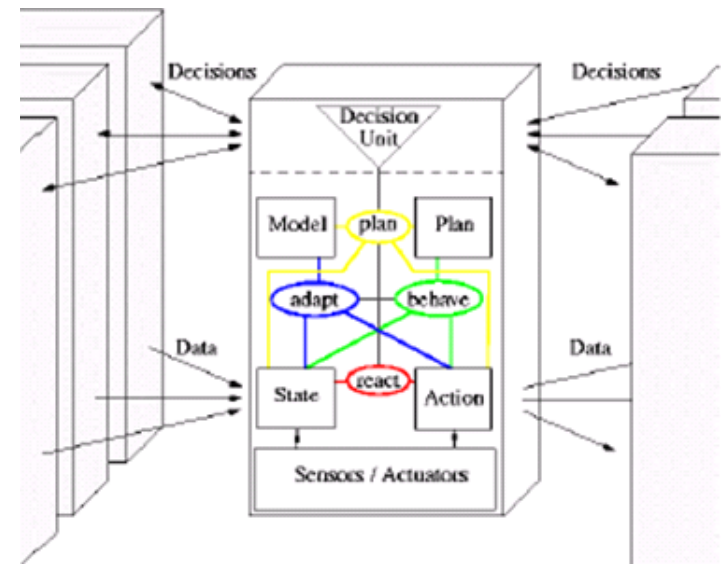
CoMRoS: Multi-Agenten-Roboter-Architektur

- Unterteilung in drei Ebenen
 - Strategische Ebene: Gesamtplanung der Agenten
 - Taktische Ebene: Kontextabhängige Planung, Kontrolle der Ausführung, erhält Anweisungen von strategischer Ebene
 - Reflexive Ebene: Prozesse mit hauptsächlich reaktivem Verhalten, teilweise in direkter Verbindung mit Hardware
- Jede Ebene besteht aus mehreren Software-Komponenten (Elementar-Agenten oder Autonomiezyklen)



CoMRoS: Multi-Agenten-Roboter-Architektur

- **Elementar-Agenten (EA)**
 - Autonom
 - Unterteilbar in Kopf (Entscheidungseinheit) und Körper (Sensor-Aktor-Zyklus)
- **Funktion und Struktur eines EA**
 - Sensor-Aktor-Zyklus
 - Zusammenschluss von Sensorik und Aktorik definiert Autonomie-Bereich des EA
- **Entscheidungseinheit**
 - Steuert abstrakten Kontrollzyklus
 - Führt Verhandlungen zwischen Agenten
- **Interaktion zwischen Entscheidungseinheiten basiert auf dynamisch konfigurierbarem Netzwerk**



CoMRoS: Multi-Agenten-Roboter-Architektur

- Knoten des Sensor-Aktor-Zyklus
 - Weltmodell
 - Pläne
 - Aktionen
 - Zustände
- Knotenverbindung durch vier Zyklen
 - Reagieren: Rückkopplung zwischen Sensorik und Aktorik
 - Verhalten: Definiert das modelltypische Agentenverhalten
 - Anpassen: Aktualisiert die Struktur des Modells
 - Planen: Erzeugt neue Pläne für den Agenten

Nächste Vorlesung ...

Roboterprogrammierung

- Programmierung von Industrierobotern
- Online-/Offline-Verfahren
- Umweltmodellierung