



# An PEARL orientierte Spezifikation verteilter eingebetteter Systeme mit UML- Stereotypen

**Shourong Lu**

FernUniversität in Hagen

Fachbereich Elektrotechnik und Informationstechnik

58084 Hagen

*[shourong.lu@fernuni-hagen.de](mailto:shourong.lu@fernuni-hagen.de)*

# Einleitung

## Übersicht und Probleme von UML für verteilte Echtzeitsysteme

- UML als Standard für den objektorientierten Entwurf
- Intensiver Einsatz zur Entwicklung komplexer Informationssysteme
- Visuelle Modellierung der Zielsysteme mit verschiedenen Diagrammtypen und Erweiterungsmechanismen von UML
- Spezifische Bedürfnisse verteilter Echtzeitumgebungen bleiben unberücksichtigt.
- UML spiegelt ein nur unzureichendes Verständnis der Anforderungen und Merkmale des Echtzeitbetriebes wider.

## Lösungsansatz

- Gründe zur Wahl von Mehrrechner-PEARL als Lösungsansatz:
  1. Hervorragende Eignung zur Spezifikation von Echtzeitanwendungen
  2. PEARL ist eher Spezifikations- als Programmiersprache.
  3. Funktionen zur Spezifikation des Verhaltens verteilter, eingebetteter Systeme können beschrieben werden.
- Definition geeigneter UML-Erweiterungsmechanismen

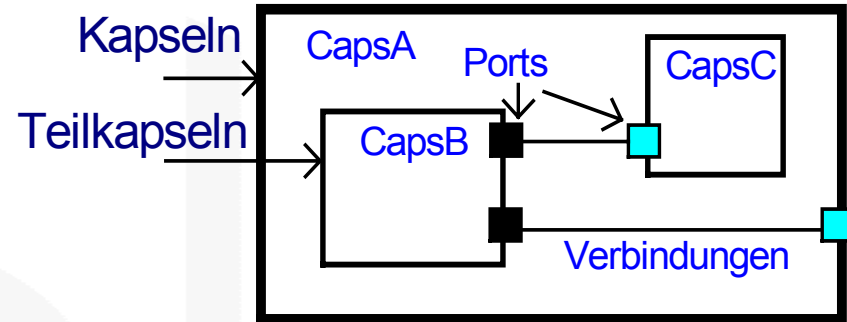
## Drei Erweiterungsmechanismen in UML:

- Mit **Stereotypen** können bestehende Modellierungselemente erweitert oder angepasst, aber auch völlig neue, anwendungsspezifische Elemente geschaffen werden.
- Mit **etikettierten Werten** können Elemente mit neuen Attributen versehen werden, um damit ihre Bedeutung zu verändern.
- Das Konstrukt **Einschränkung** in Modellelementen gibt Semantikspezifikationen bzw. Bedingungen an.

Mit diesen Mechanismen werden die Elemente eines Mehrrechner-PEARL abbildenden UML-Profiles definiert, um das Verhalten verteilter, eingebetteter Echtzeitsysteme genau beschreiben zu können.

## Architekturkonzepte

- **Kapseln (Komponenten)**
  - **Ports**
  - **Verbindungen**
  - **Protokolle**
- Eine Kapsel interagiert mit ihrer Umgebung und mit den Teilkapseln über eine Anzahl von Ports.
  - An Stelle von Methoden haben die Kapseln Ports, die für die Kommunikation nach außen zuständig sind.  
Den Ports werden Protokolle zugeordnet.

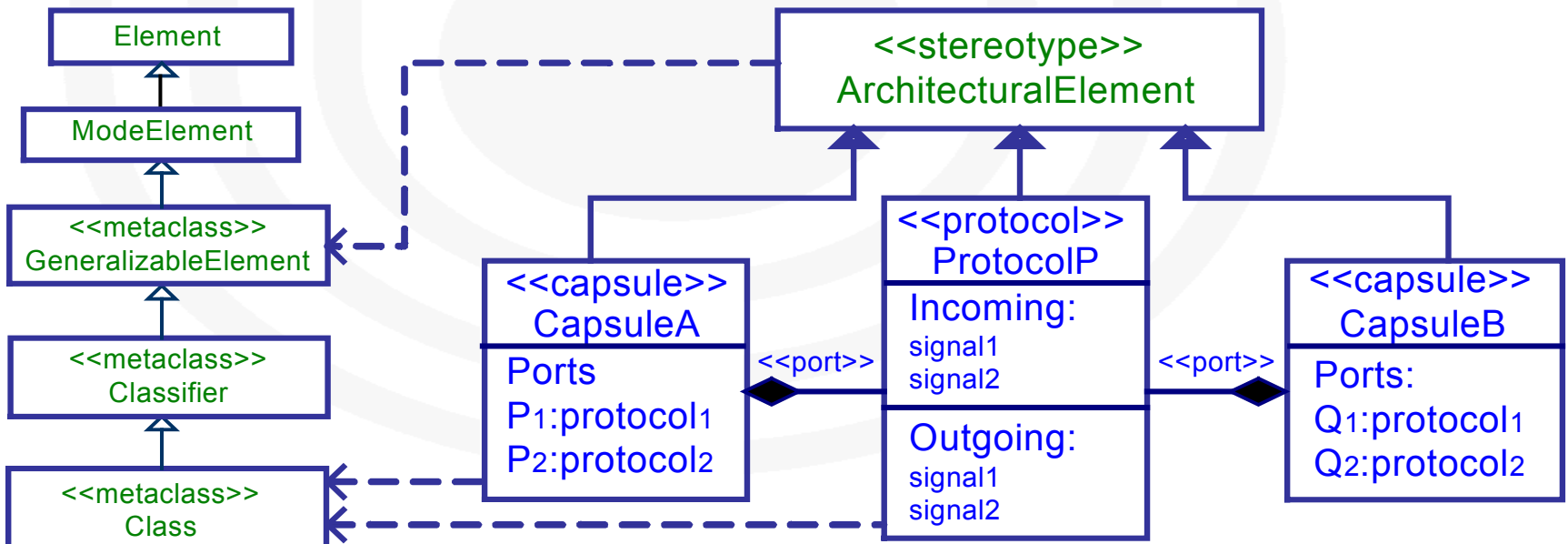


# Mehrrechner-PEARL abbildende UML-Erweiterung

## Kapseln, Ports, Verbindungen und Protokolle

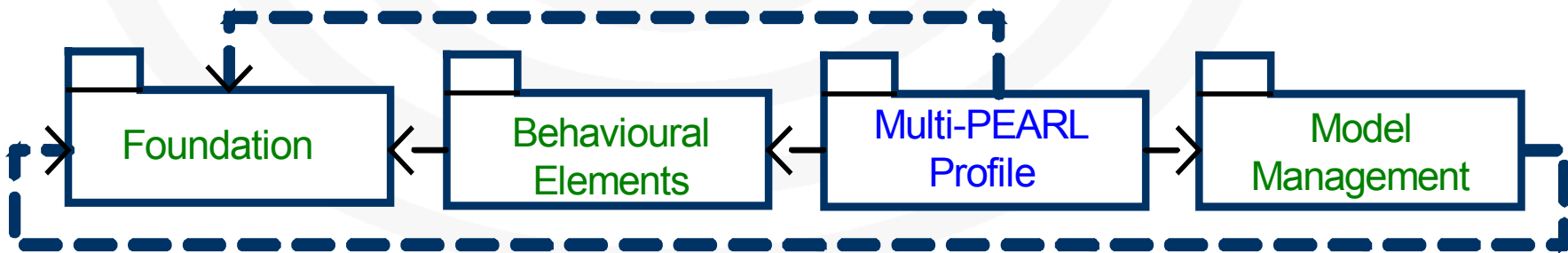
Kapsel, Protokoll und Port sind ihrerseits Stereotypen von „**ArchitecturalElement**“ und gehören der Modellschicht an.

Kapseln können nur mit anderen Kapseln über Ports mit in einem Protokoll vordefinierten Signalen kommunizieren.



## Ein UML-Profilpaket für Mehrrechner-PEARL

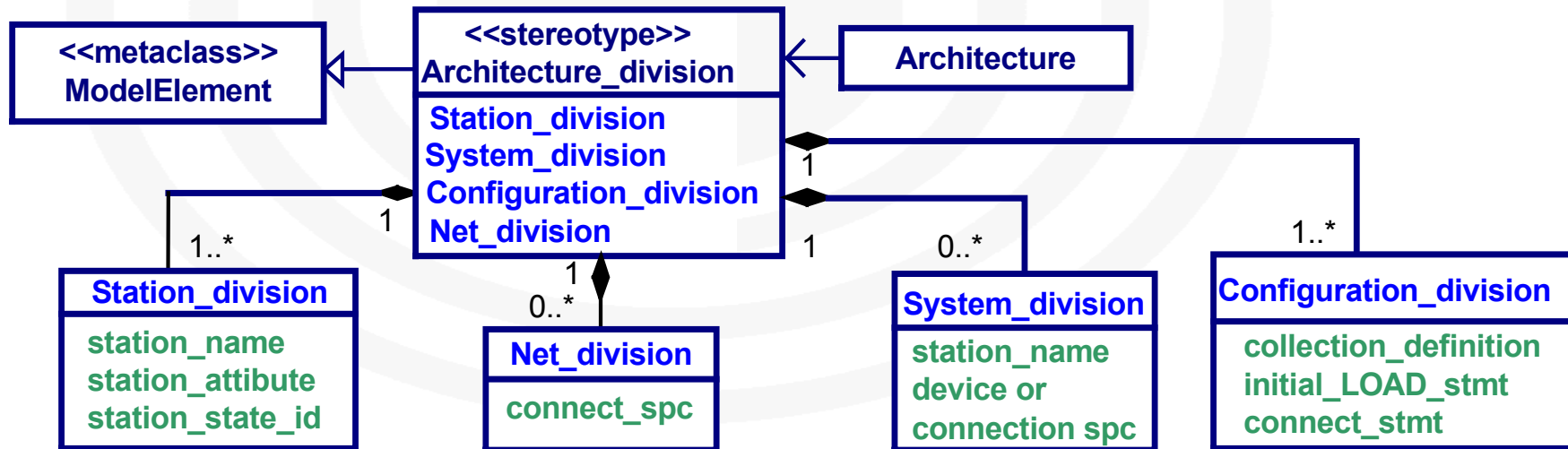
- Das PEARL-Profil ist durch ein UML-Paket dargestellt, das die die PEARL-Konstrukte abbildenden Elemente beschreibt.
- Das PEARL-Profilpaket hängt von dem Verhaltenselement „**Foundation**“ und den Modellverwaltungspaketen ab, da es die Notationen von PEARL definiert und einige Stereotypen generieren muss.



# Mehrrechner-PEARL abbildende UML-Erweiterung

## Einführung eines UML - Klassendiagramms als PEARL-Profilpaket

- Darstellung des **Architekturmetamodells** als eine Architekturbeschreibung von PEARL
- Stereotypisiertes Modell mit Verwendung des „**ModelElement**“ als Basisklasse
- Beschreibung der Systemarchitektur als Menge von Knoten

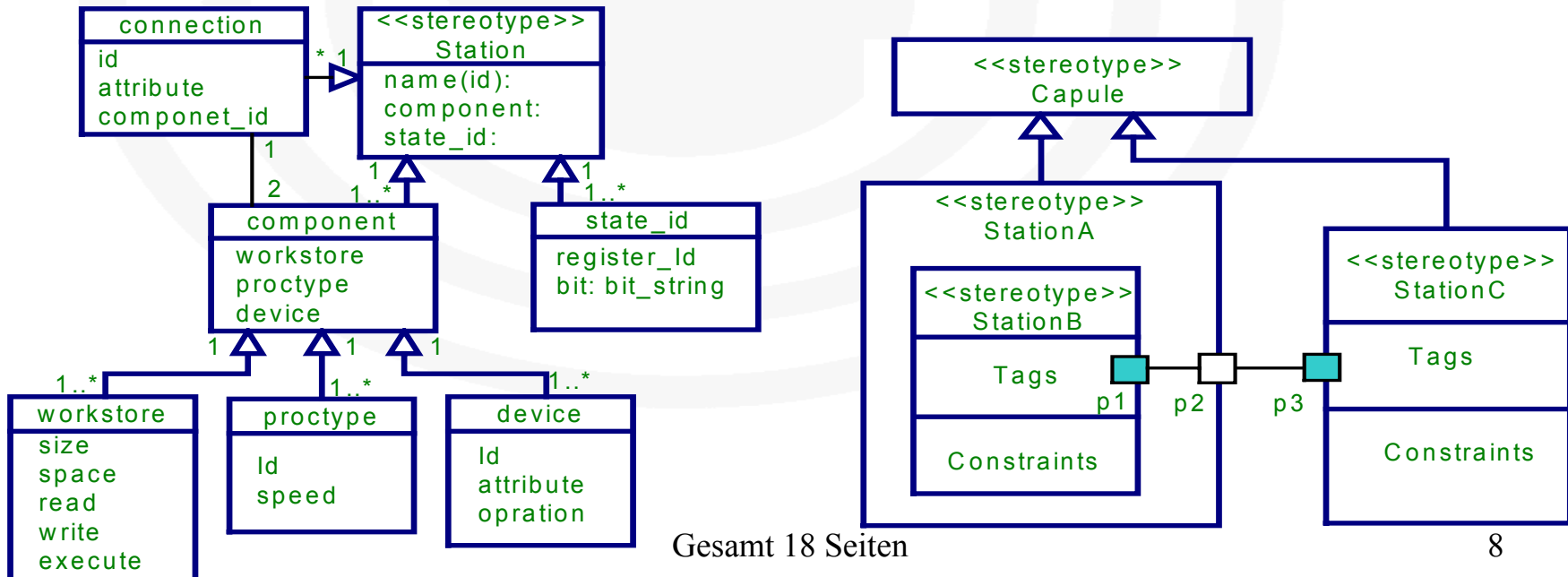


# Mehrrechner-PEARL abbildende UML-Erweiterung

## Stationsteil

- Im Stationsteil werden die Verarbeitungsknoten eines Systems eingeführt.
- Ein System kann mehrere, eindeutig identifizierte und für Rekonfigurierungszwecke mit entsprechenden Zustandsinformationen versehene Stationen umfassen.
- Die Grundkomponenten einer Station sind Prozessortyp, Arbeitsspeicher und Geräte.
- Als Stereotyp von Kapsel enthält Station auch alle deren Eigenschaften und verwendet mithin Ports zur Kommunikation mit anderen Stationen.

## Strukturdiagramm mit Stereotyp für das Konstrukt Station :



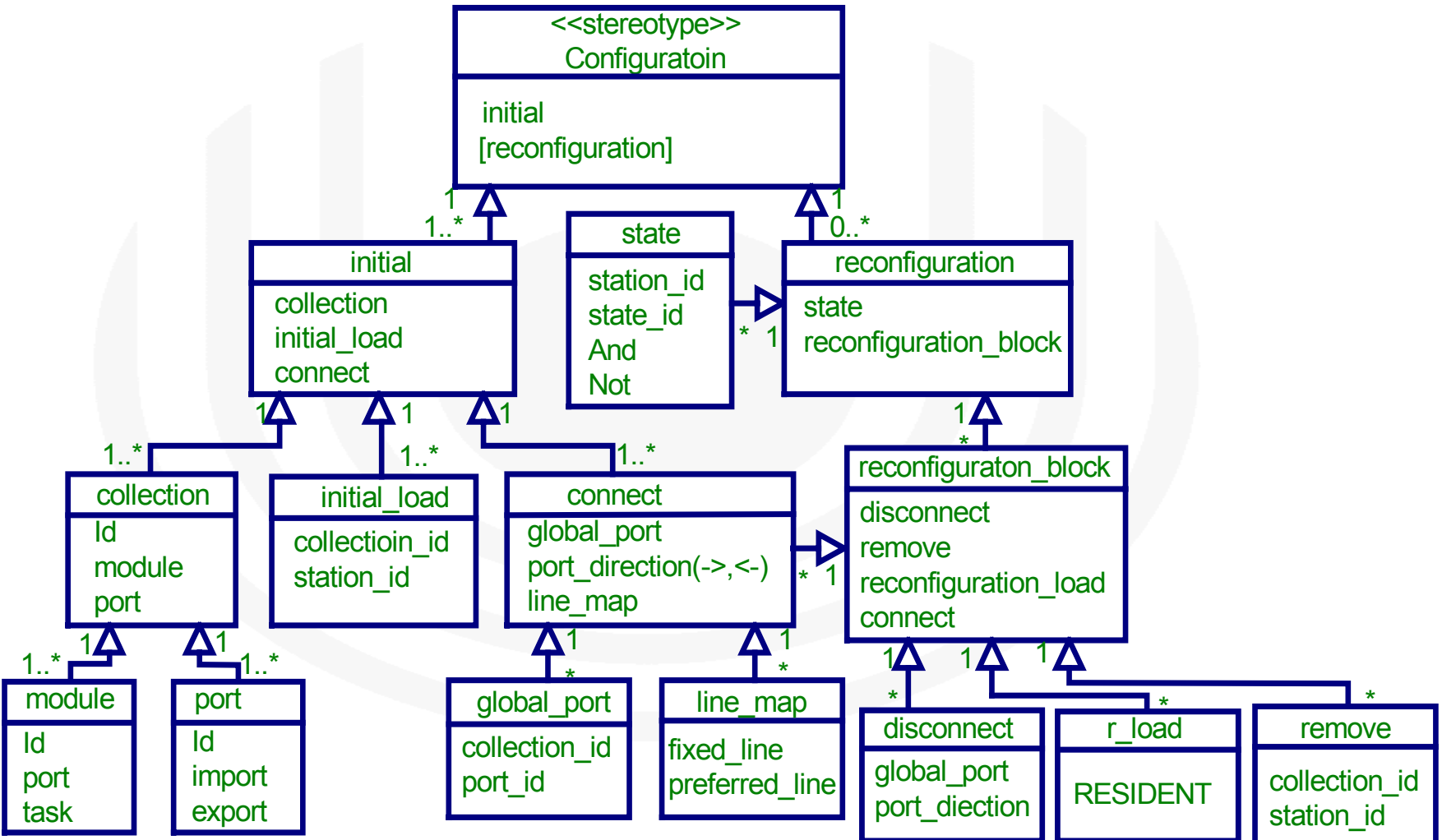


# Mehrrechner-PEARL abbildende UML-Erweiterung

- **Konfigurationsteil in Mehrrechner-PEARL:**
  - Für ausführbare Anweisungen zum Laden und Entfernen von Collections,
  - zum Auf- und Abbau logischer Kommunikationspfade.
- **Collections:**
  - Programme sind als zu Collections gruppierte Module strukturiert.
  - Collections werden im Rahmen der Konfiguration entweder statisch oder dynamisch Systemknoten zugewiesen.
  - Collections stellen die Elemente der dynamischen Rekonfiguration dar.
- **Ports:**
  - Collections kommunizieren untereinander allein durch Nachrichtenaustausch auf der Basis des Portkonzepts.
  - Zu einem Port gesendete oder von dort empfangene Nachrichten sind nur lokal in der eigenen Collection bekannt.

# Mehrrechner-PEARL abbildende UML-Erweiterung

## Klassendiagramm der Konfigurationsstruktur



## Stereotype entsprechend der Collection in PEARL

```
Stereotype CollectionStereotype {          /* Name of the stereotype */
  host Capsule;          /* Capsule can be stereotyped by this stereotype */
  properties {          /* Declaration of the syntactic properties */
    String id;
    [ 'MODULE_enumeration', 'PORT_enumeration' ] role;
  }
  restrictions{          /* Restrictions for stereotype */
    ( ( role = 'MODULE_enumeration' ) implies
      /* MODULE_enumeration must have an identifier */
      (exists M in self.operations |
        M = id_Task )          /* To activate adaption module*/
    )
    ( role = 'PORT_enumeration' ) implies
      (exists I , O in self.operations |
        (I = 'Import') and (O = 'Outport' )
      )
    )
    and (self.subject -> notEmpty ) /* There must be a subject for this Collection*/
    and (exists c in self.subject |          /* This means an associated Class */
      exists C in c.stereotypes | /* with a matching stereotype */
        (C = Collection) and (C.role = 'state') and (C.id = id) )
  } }
}
```

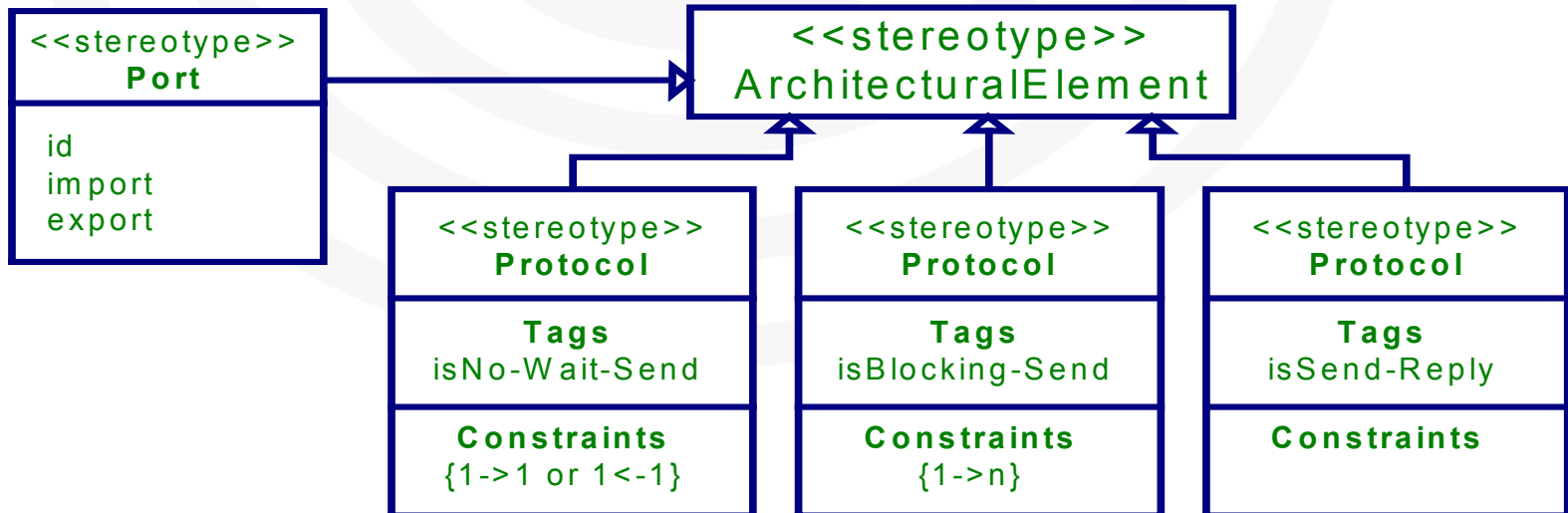
# Mehrrechner-PEARL abbildende UML-Erweiterung

## Protokoll und Port (Mechanismus der Kommunikation im Mehrrechner-PEARL)

Im wesentlichen dienen Protokolle zur Definition von Kommunikationsmustern. Ports werden mit den Typen ein- oder ausgehend versehen, wodurch den über die Ports laufenden Interaktionen Beschränkungen auferlegt werden.

## Stereotypen für die PEARL-Konstrukte Protokoll und Port:

Nach dem Muster der PEARL-Konstrukte Protokoll und Port aufgebaute Stereotypen durch Modifikation des Stereotypen „**ArchitecturalElement**“



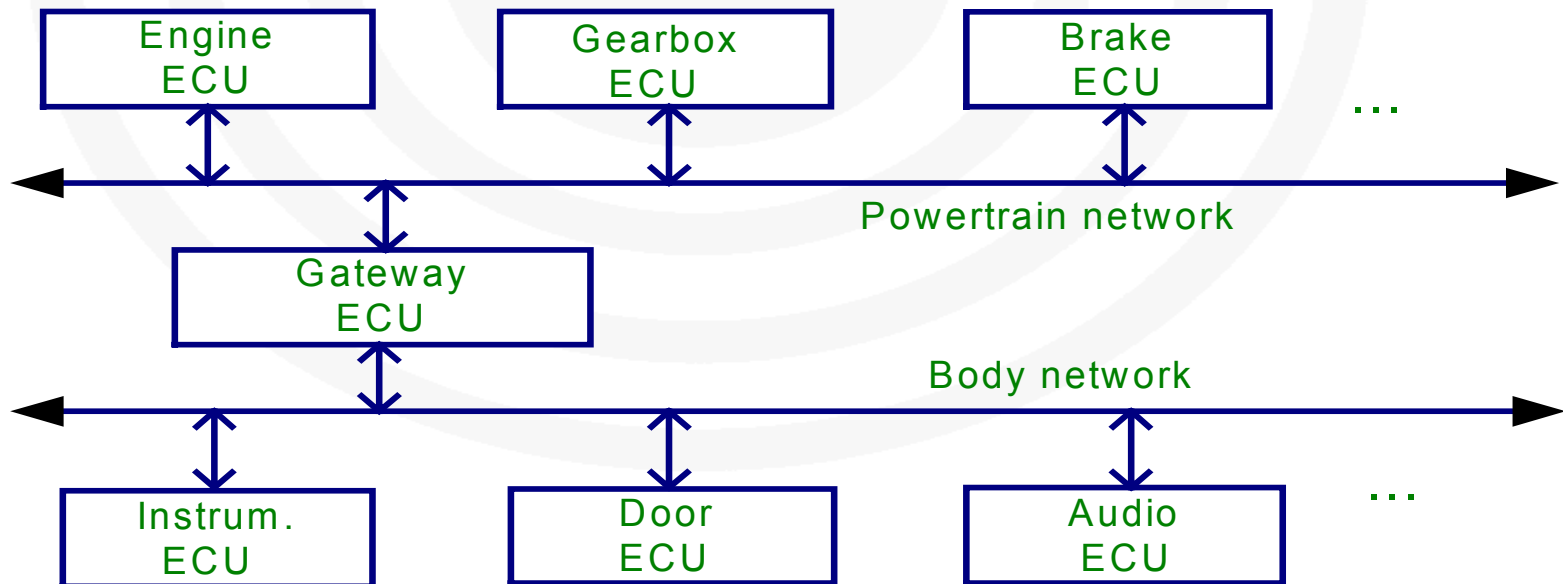
# Modellierung eines verteilten, elektronischen Automobilbordsystems

## Systemarchitektur

Die Systeme bestehen aus einer Reihe elektronischer Steuereinheiten (ECUs).

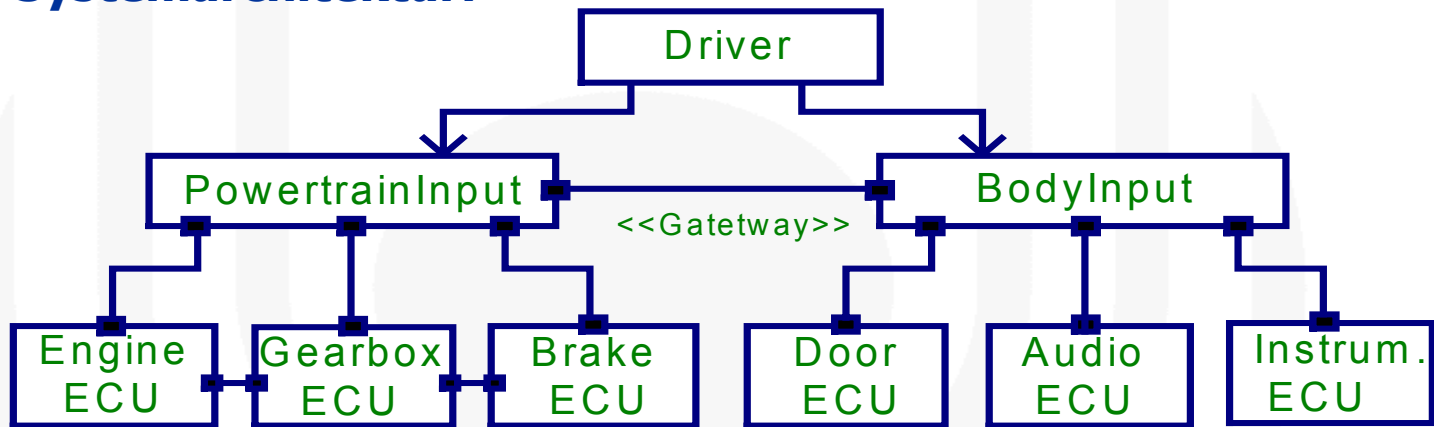
Die ECUs sind durch 2 CAN-Feldbusse miteinander verbunden.

Beide Feldbusse sind über ein Gateway untereinander verbunden.



# Modellierung eines verteilten, elektronischen Automobilbordsystems

Konzeptioneller Entwurf in Form eines Kapselkollaborationsdiagramms aus obiger Systemarchitektur:

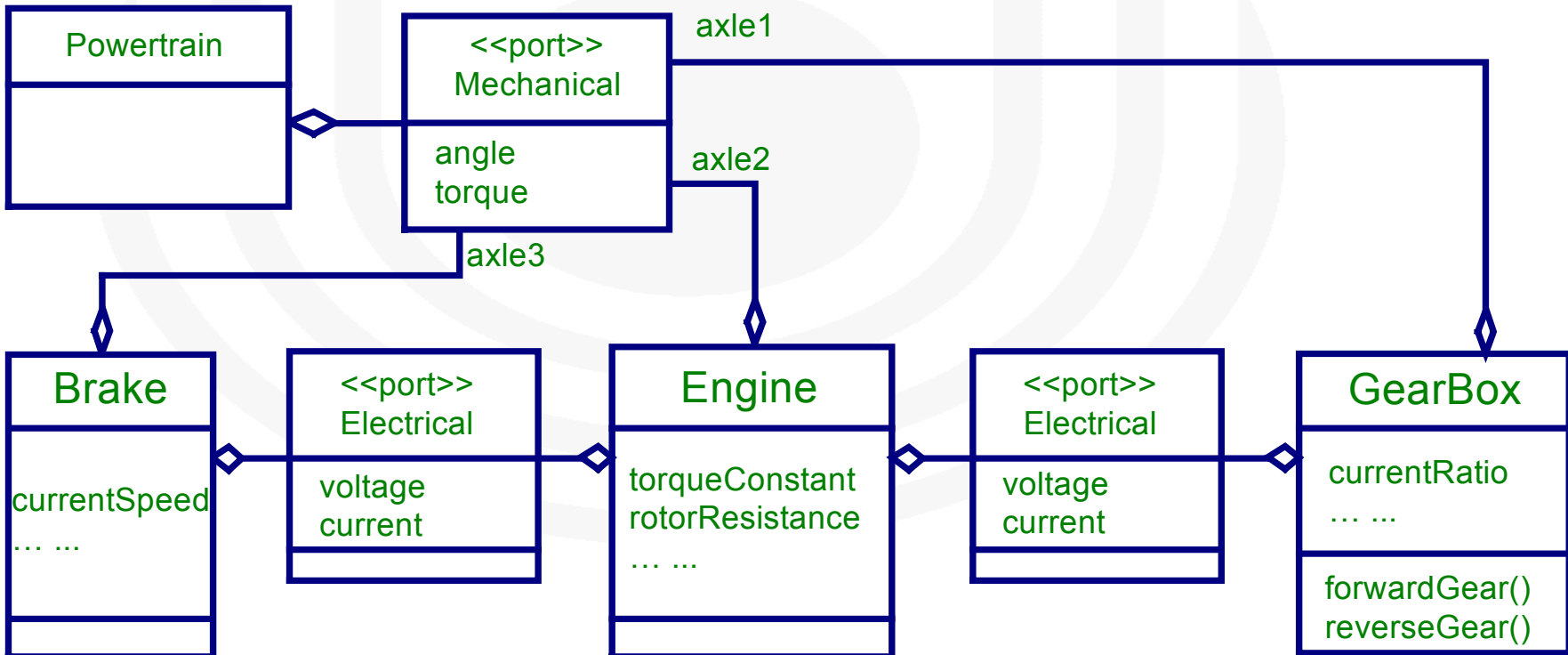


- **Zwei Subsysteme steuern den Antriebsstrang und die Karosserie.**
- **Sie werden durch je eine Kapsel für Antriebsstrang und Karosserieeinwirkung mit einem Verbindungsprotokoll zum Datentransfer zwischen den ECUs dargestellt.**
- **Es gibt Punkt-zu-Punkt- und Mehrpunktverbindungen, die beide mit dem Stereotyp Protokoll modelliert werden können.**
- **Die ECU-Kapsel ist eine spezialisierte aktive Klasse.**
- **Zur Kommunikation bedienen sich Kapseln ihrer Ports.**

# Modellierung eines verteilten, elektronischen Automobilbordsystems

Das Kapselkollaborationsdiagramm wird durch die Definitionen des Klassendiagramms verfeinert.

Ein Klassendiagramm der Architektur des Bordsystems zeigt die folgende Abbildung des Subsystems Antriebsstrang.



# Modellierung eines verteilten, elektronischen Automobilbordsystems

- Zwischen den ECUs für Motor, Getriebe und Bremsen gibt es zwei Ports, die wir mit Port-Stereotypen beschreiben können:
  - ein Port zur Übertragung elektrischer Größen,
  - ein Port zur Angabe der mechanischen Größen Winkelgeschwindigkeit und Drehmoment der Antriebswelle.
- Die drei genannten ECU-Klassen enthalten die Charakteristika der die Komponenten beschreibenden Attribute. Den Klassen sind Invarianten beigegeben, um die ihr Verhalten bestimmenden physikalischen Gesetze zu spezifizieren.
- Durch Modellierung des dynamischen Verhaltens kann jede ECU-Kapsel mit Zuständen und Übergängen spezifizierenden Tabellen versehen werden.
- Ablaufdiagramme zur Beschreibung der Interaktion von Kapseln
- Protokollzustandstabellen zur Darstellung des protokollspezifischen Nachrichtenaustausches



# Schlußfolgerung und weitere Arbeiten

Das Anwendungsbeispiel zeigt, dass sich mit einem objektorientiertem Entwurf unter Verwendung von UML und der an Mehrrechner-PEARL orientierten Stereotypen für verteilte Systeme folgende Probleme überwinden lassen, vor denen die Automobilindustrie zur Zeit noch steht:

- Spezifikationen sind nicht formal genug, um sorgfältig in Hinblick auf Umsetzbarkeit, Kosten und Entwicklungsdauer analysiert werden zu können,
- Entwürfe werden auf zu niedrigem Abstraktionsniveau durchgeführt,
- Steuerungsalgorithmen implementierende Programme sind unstrukturiert,
- auf dem Betriebssystemniveau werden Tasks nach heuristischen Kriterien definiert und das Ablaufzeitverhalten von Systemen wird erst anhand von Implementierungen analysiert.

# Weitere Arbeiten

**Das hier präsentierte UML-Profil gibt die in Mehrrechner-PEARL enthaltenen Kernkonzepte zur Modellierung wieder.**

- **Dieses Profil wird um die Aspekte Echtzeit und Sicherheit erweitert werden,**
- **dann wird es in ein vorhandenes UML-basiertes Entwicklungswerkzeug eingebettet werden und**
- **schließlich soll eine abstrakte Entwicklungsumgebung erstellt werden.**



Vielen Dank  
für Ihre  
Aufmerksamkeit!