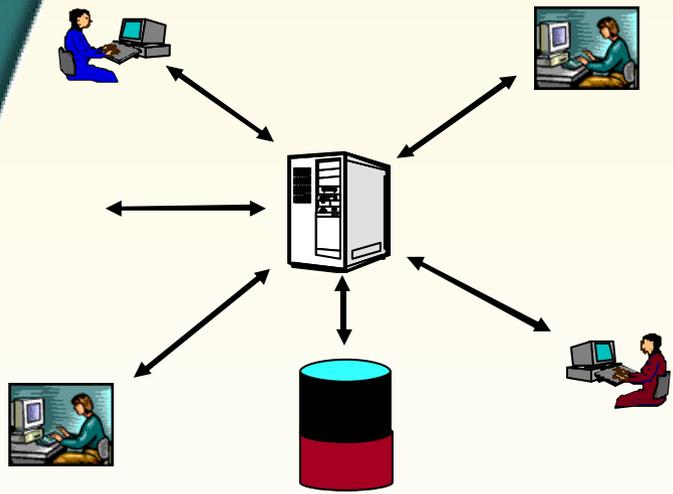


# ARTiSAN Software Tools



Der neue Modellierungsstandard UML 2: Besser geeignet für Eingebettete Systeme?

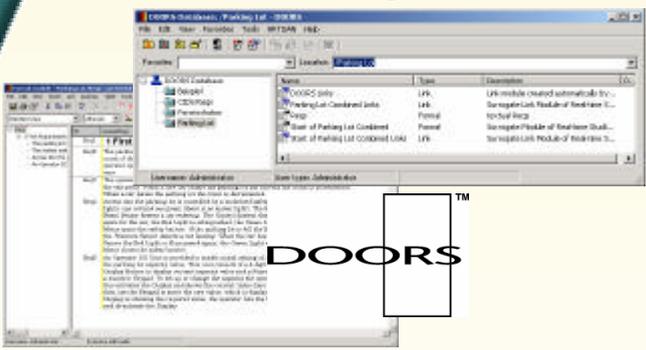
Andreas Korff, Principal Consultant

Workshop „Eingebettete Systeme“ der GI-Fachgruppe „Echtzeitsysteme und PEARL“, Boppard, 25./26.11.2004

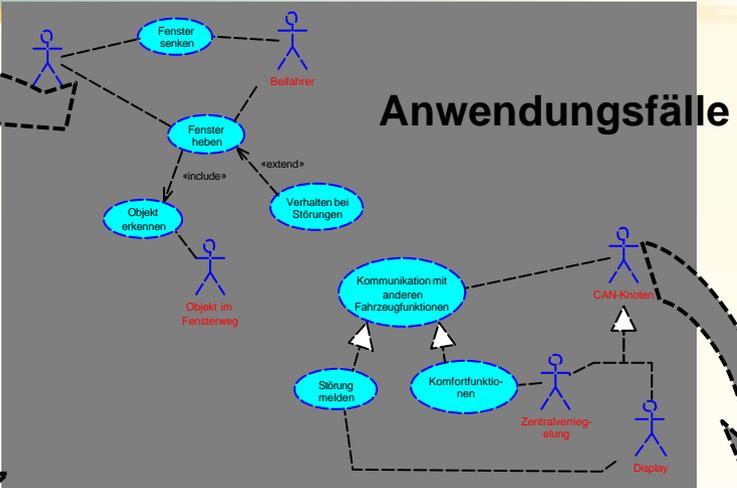
# Wer ist ARTiSAN?

- CASE-Tool-Hersteller, gegründet 1997
  - Hauptsitz in Europa: Cheltenham, UK
- Fokussiert auf das Marktsegment Real-time/Embedded Systems
- Pragmatischer Ansatz zur Anwendung der objektorientierten Technologie für die Entwicklung von Echtzeitsystemen
  - Mit Entwicklungsprozessvorschlag RtP
  - Spec-konforme UML-Erweiterungen für alle Sichten der Systementwicklung

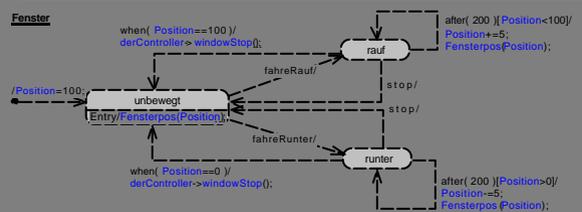
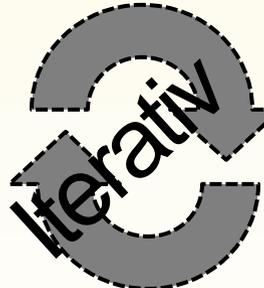
## Textuelle Requirements



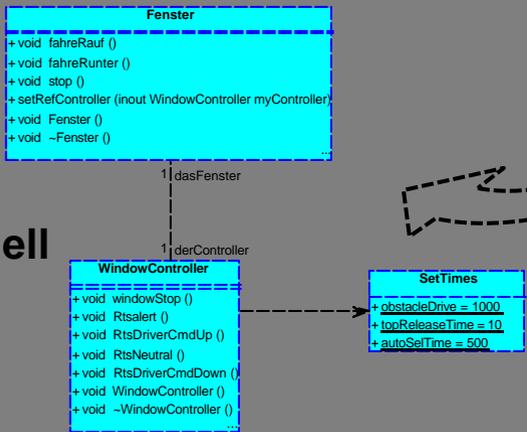
DOORS



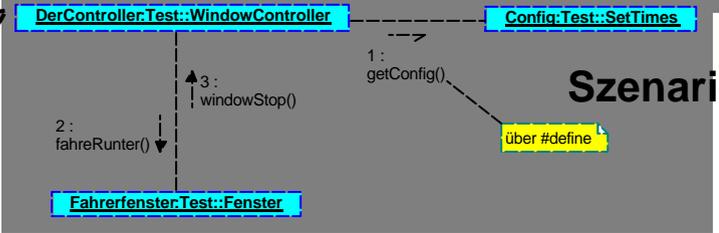
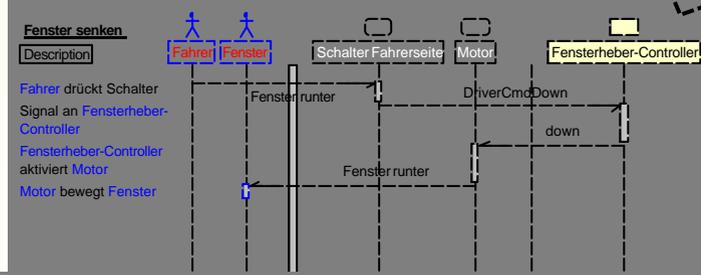
## Anwendungsfälle



## Objektdynamik

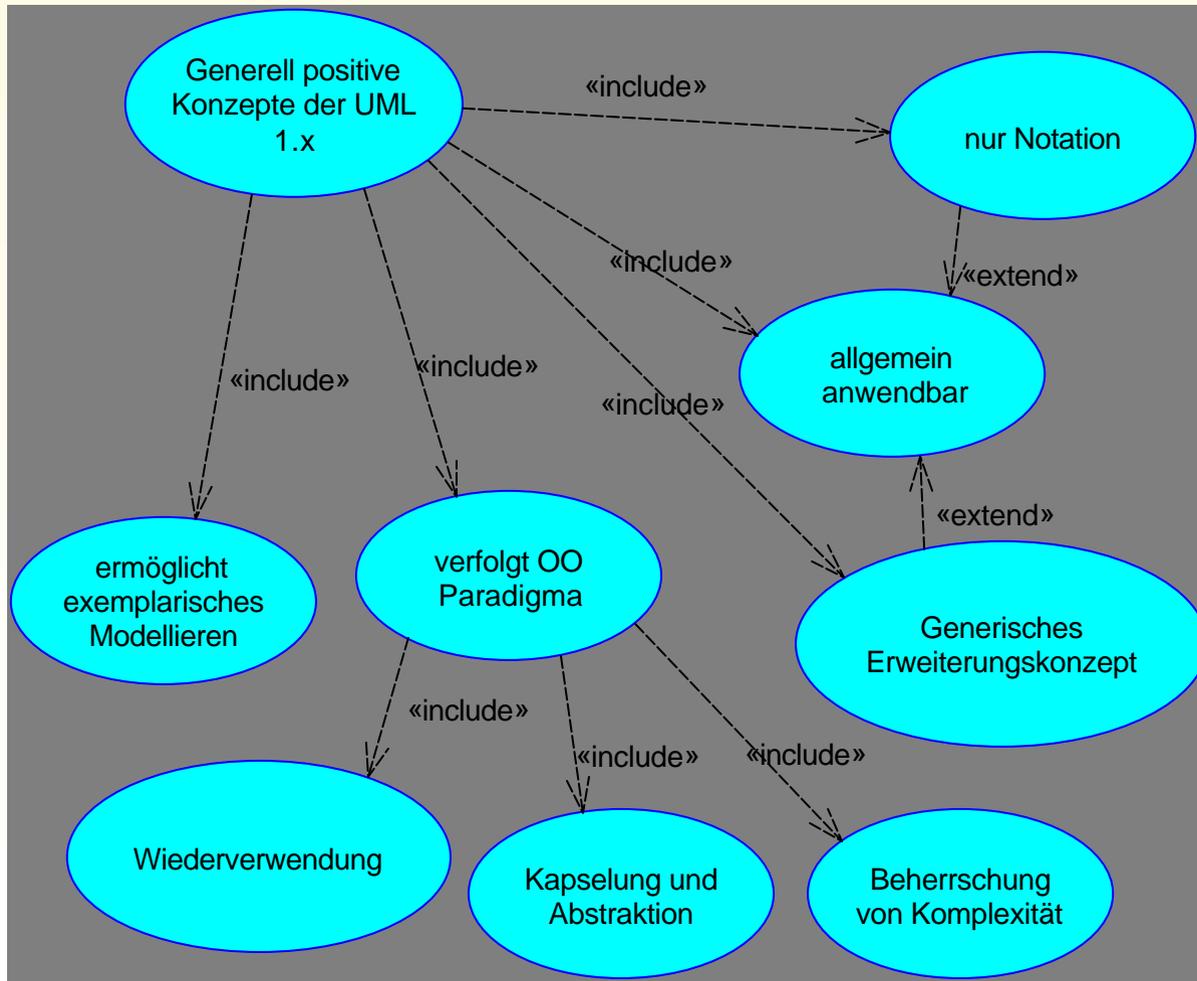


## Klassenmodell



## Szenarien

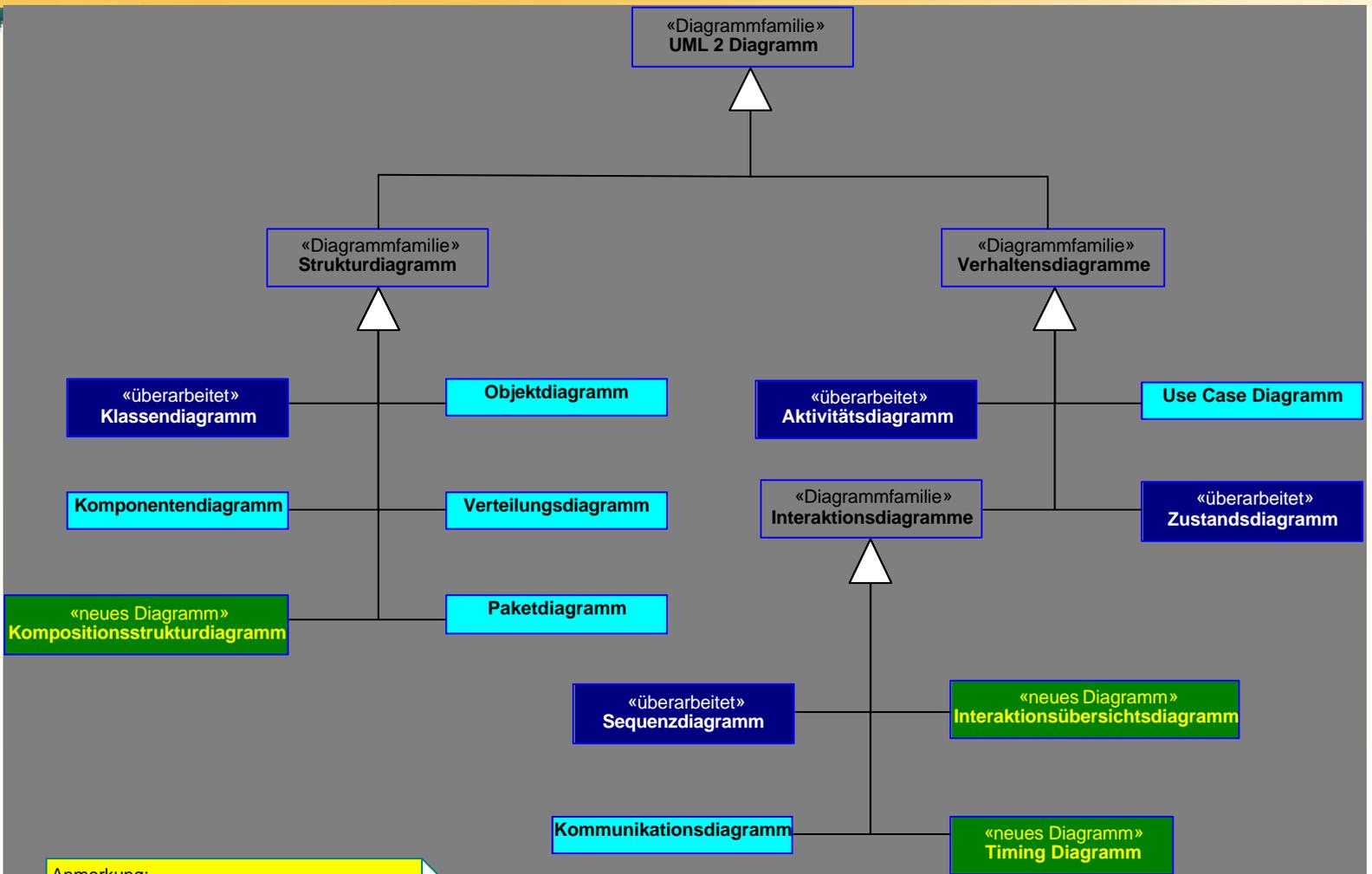
# Warum UML?



- Vorteile
  - Die Beschränkung auf nur die Festlegung einer Notation macht die UML in allen Bereichen anwendbar.
  - Use Case-zentrierte Analyse und Design ermöglicht den Fokus auf Kunden- bzw. Nutzersicht.
  - Objektorientierte Strukturen gewährleisten den Re-Use-Aspekt.
  - Die Netzwerkstruktur eines UML-Modells führt zu einem „Design-to-Change“.
- Nachteile
  - Softwarelastig, wichtige Systementwicklungskonzepte sind nicht berücksichtigt
  - Herkunft aus der Zusammenfassung mehrerer Modellierungsstrategien z.T. sichtbar (z.B. Aktivitätsdiagramme, OCDs, OSDs)

- **Hauptfokus:**
  - **Bessere Modellierbarkeit von Architekturkonzepten**
  - **Verbesserte Skalierbarkeit => Große Systeme sind besser darstellbar**
  - **Strukturierbarkeit von Systemen in Teilsysteme**
  - **Strukturierbarkeit von Verhalten**
- **Insgesamt 13 Diagrammarten erweitern die Modellierungsmöglichkeiten**
  - **Diese Freiheit sollte aber im jeweiligen Entwicklungsprozess an die Notwendigkeiten für das zu entwickelnde System angepasst werden.**

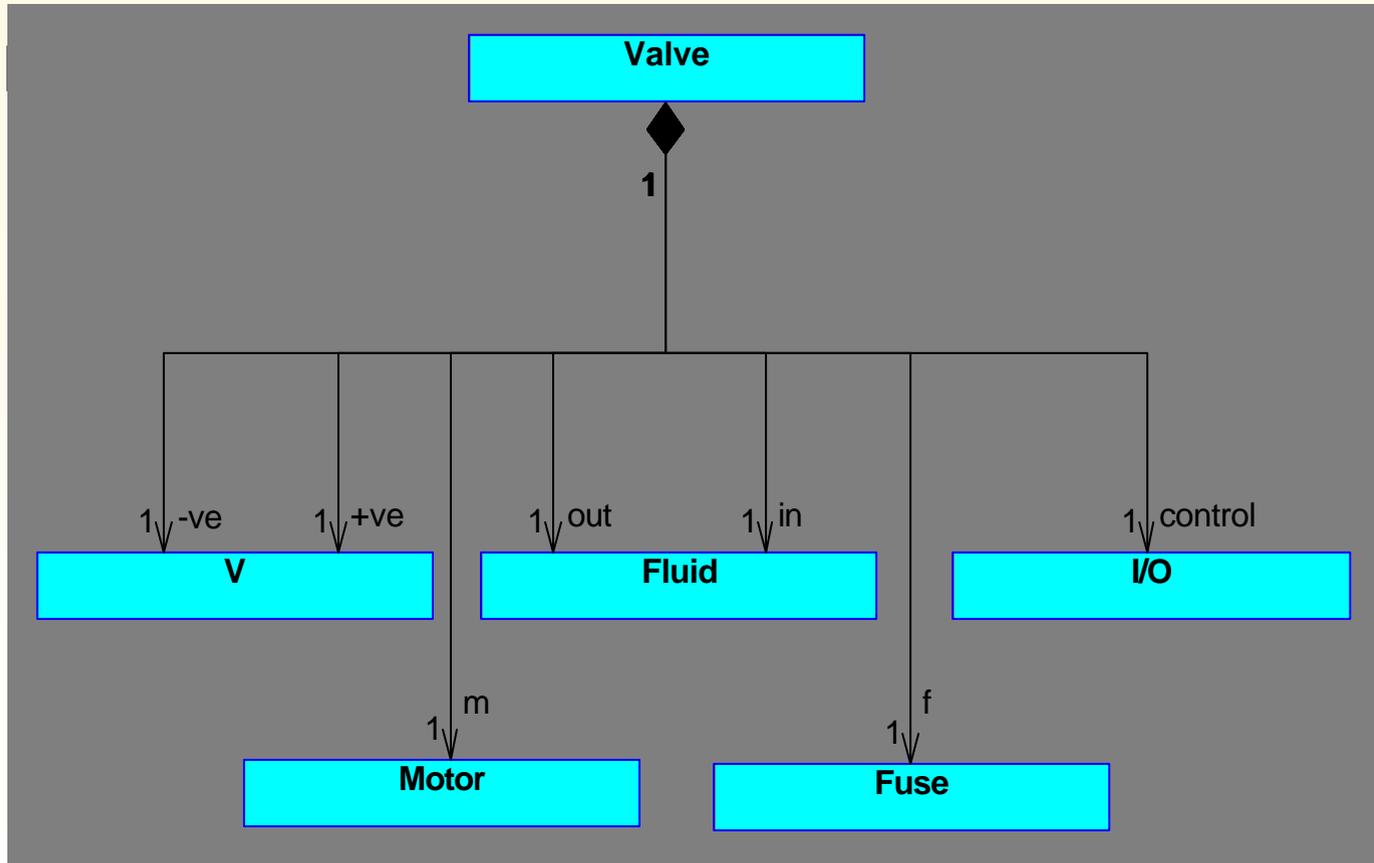
# Diagrammarten der UML 2



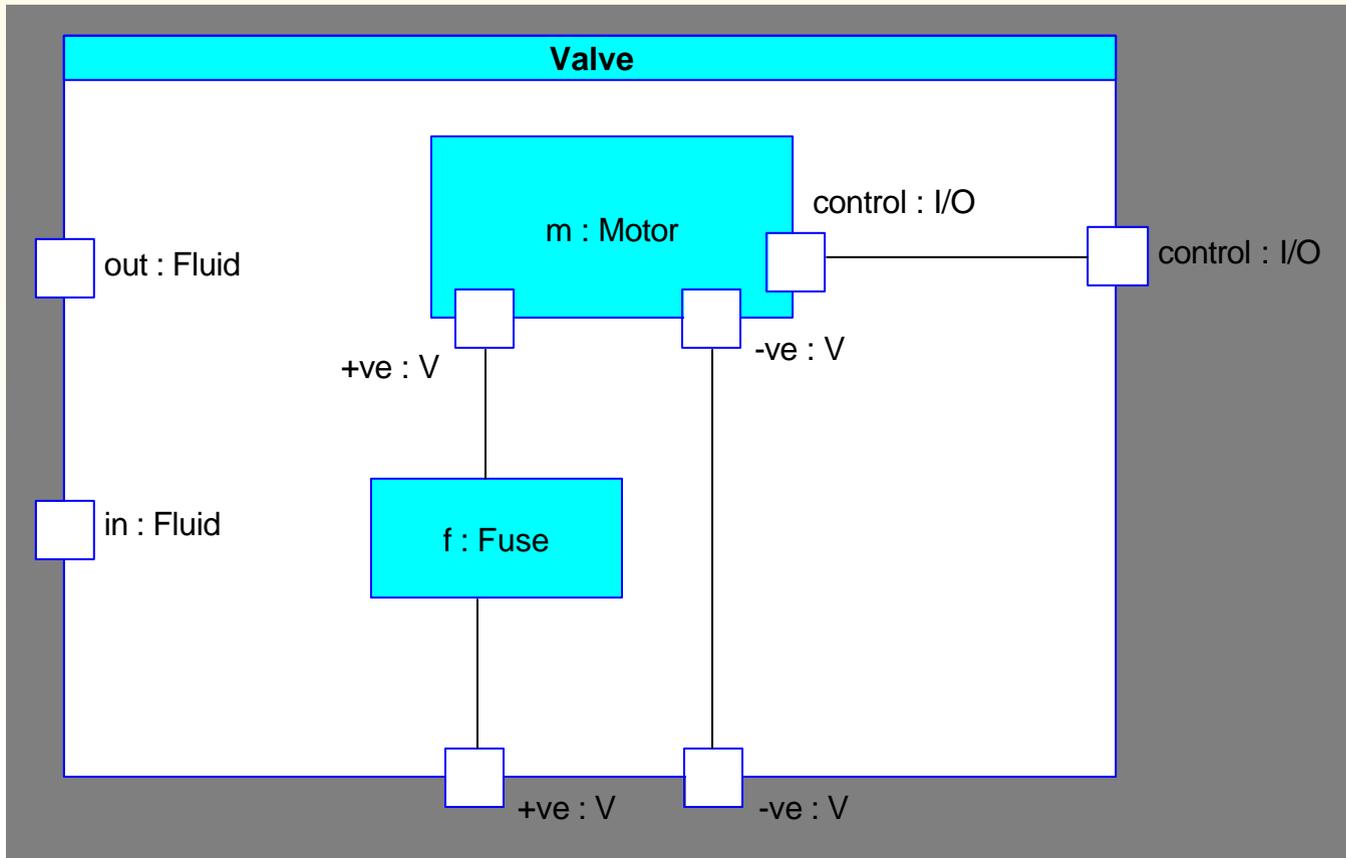
Anmerkung:  
 Jede Diagrammart wurde überarbeitet.  
 Das explizite <<überarbeitet>> weist auf eine  
 signifikante Änderung hin.

- Strukturinformation
- Kontextdefinition
  - Systemgrenzen mit Interfacedefinition
- Darstellungsmöglichkeit komplexer Abläufe
  - Nachrichtenfluss
  - Aktivitäten

- In UM



- In UML 2 überwiegt die Systemsicht



# Neue Elemente im Kompositionsstrukturdiagramm

- **Part:**
  - Classifier innerhalb des „ganzen“ Classifiers
  - Rolle oder Attribut innerhalb der Teil-Ganzes-Hierarchie
  - Zeigt die interne Strukturierung des „Ganzen“
- **Port:**
  - Teil des „Ganzen“, das zur Interaktion mit anderen, äußeren Elementen dient
  - Kann durch benötigte und/oder bereitgestellte Schnittstellen ergänzt werden
- **Konnektor:**
  - Zeigt die interne „Verkabelung“ von Ports und Parts
  - Die enthaltende Instanz leitet die auftreffenden Informationen entlang der Konnektoren weiter

## Referenz zu anderen Sequenzen

```

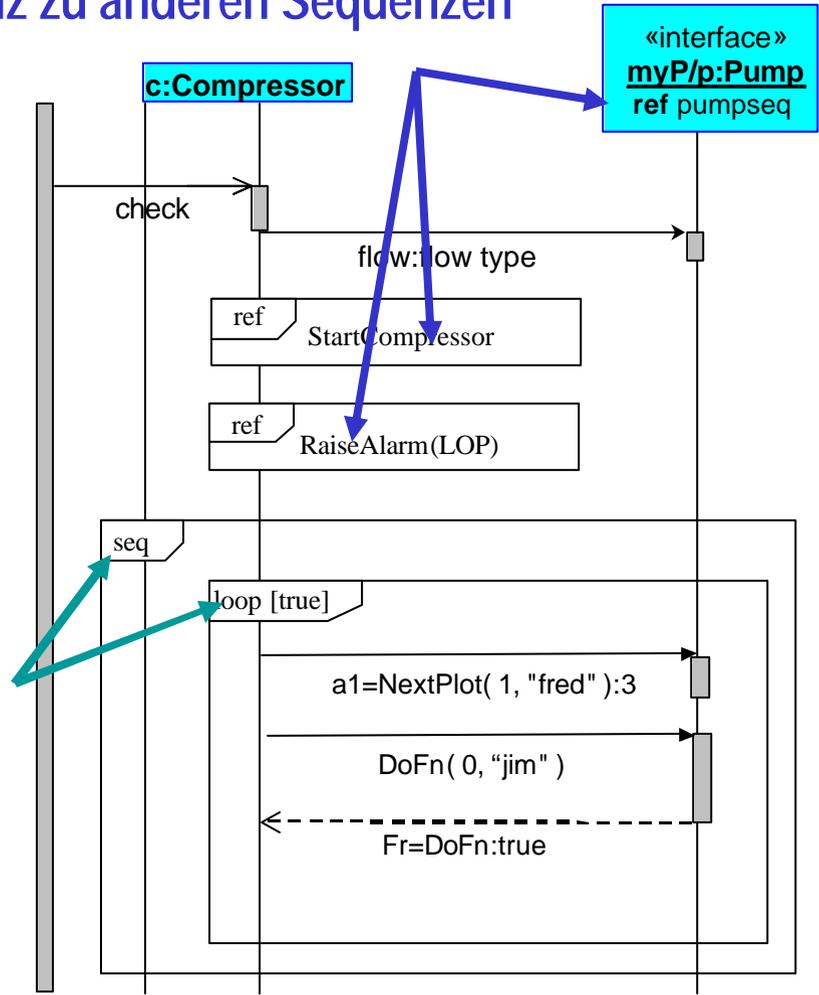
seq StartUp
  Standard event message
  ... and an I/O flow
  alt compressorOK
    diagram ref
  else
    and one with parameters
  end alt

  seq
    Loop forever
    Operation message (full)
    Operation message

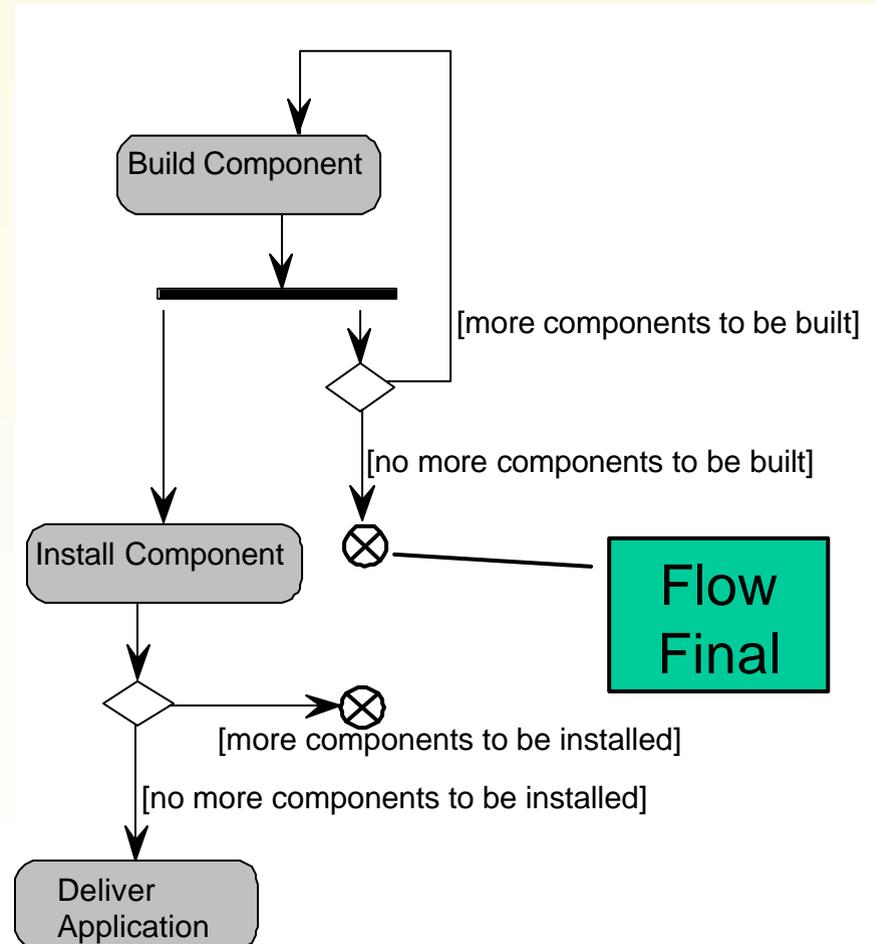
    Return message

    break [fr]
  end loop
  
```

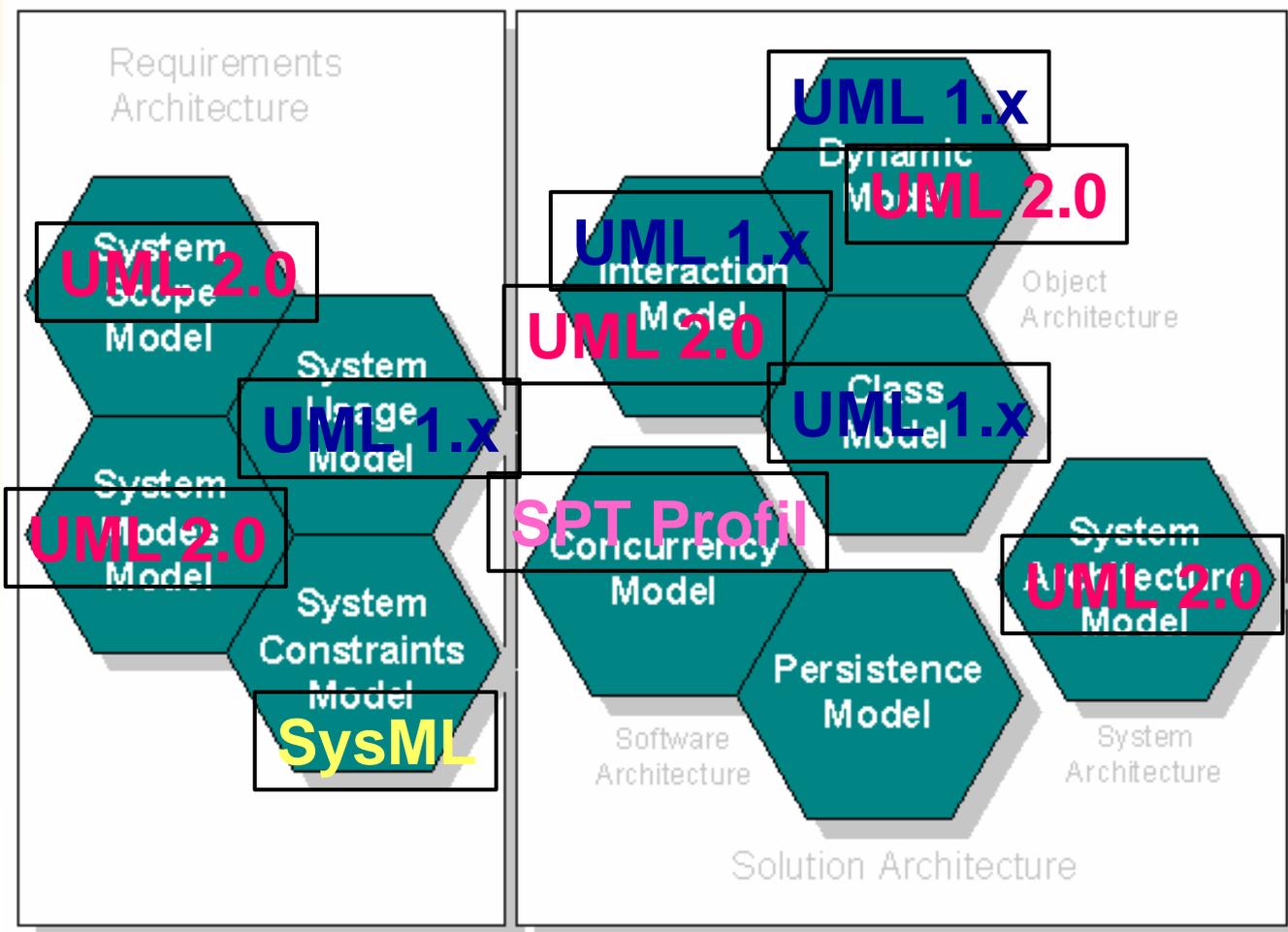
Beispiele zur Strukturierung



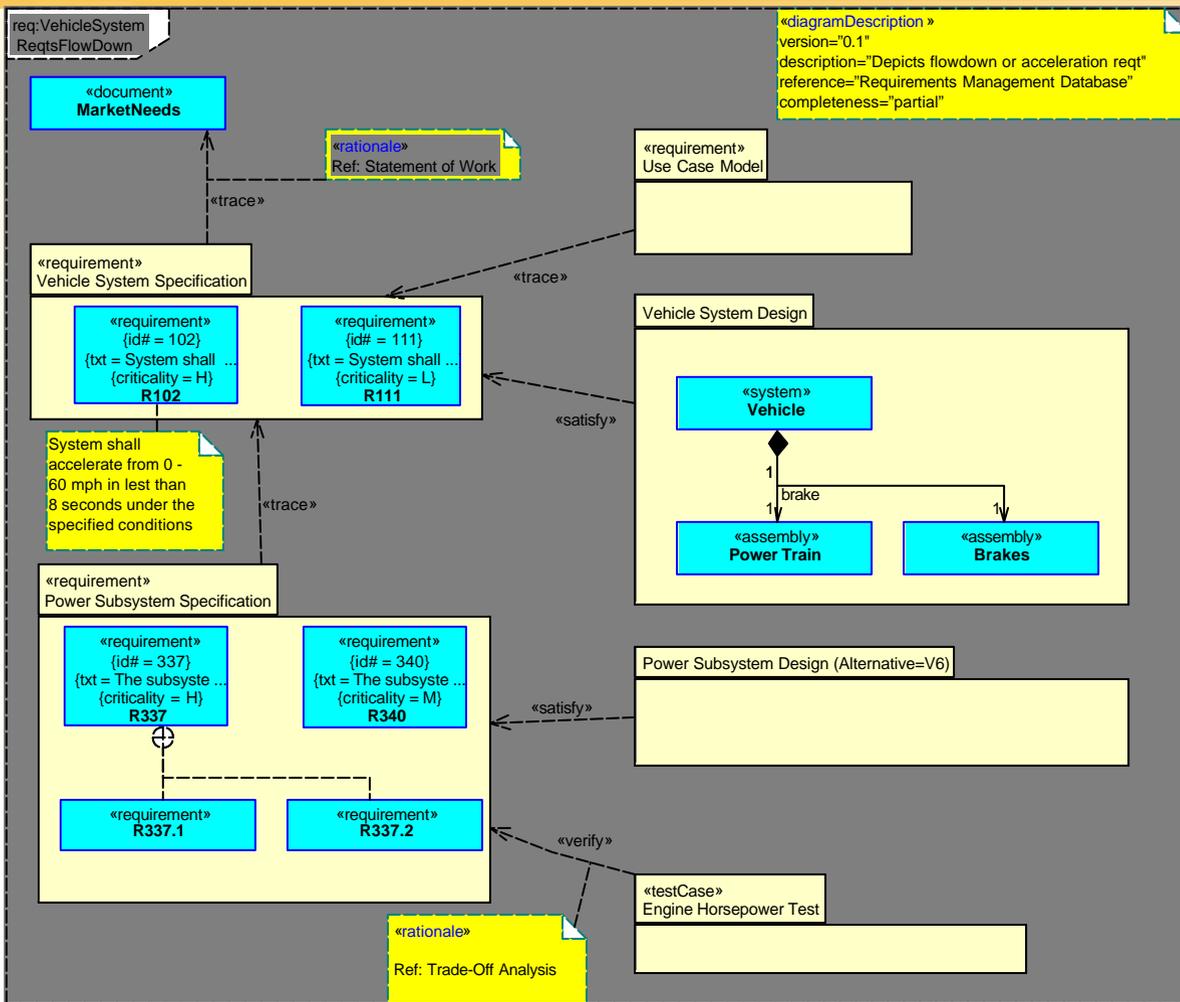
- Eigene Semantik (Token-basiert)
- Neu:
  - nested swimlanes
  - strukturierte Knoten
  - Final Flow Symbole (die nur einen Ablauf, aber nicht die ganze Aktivität beenden)
  - Pins
  - und vieles mehr



# Prozess-Sicht (Beispiel RtP)



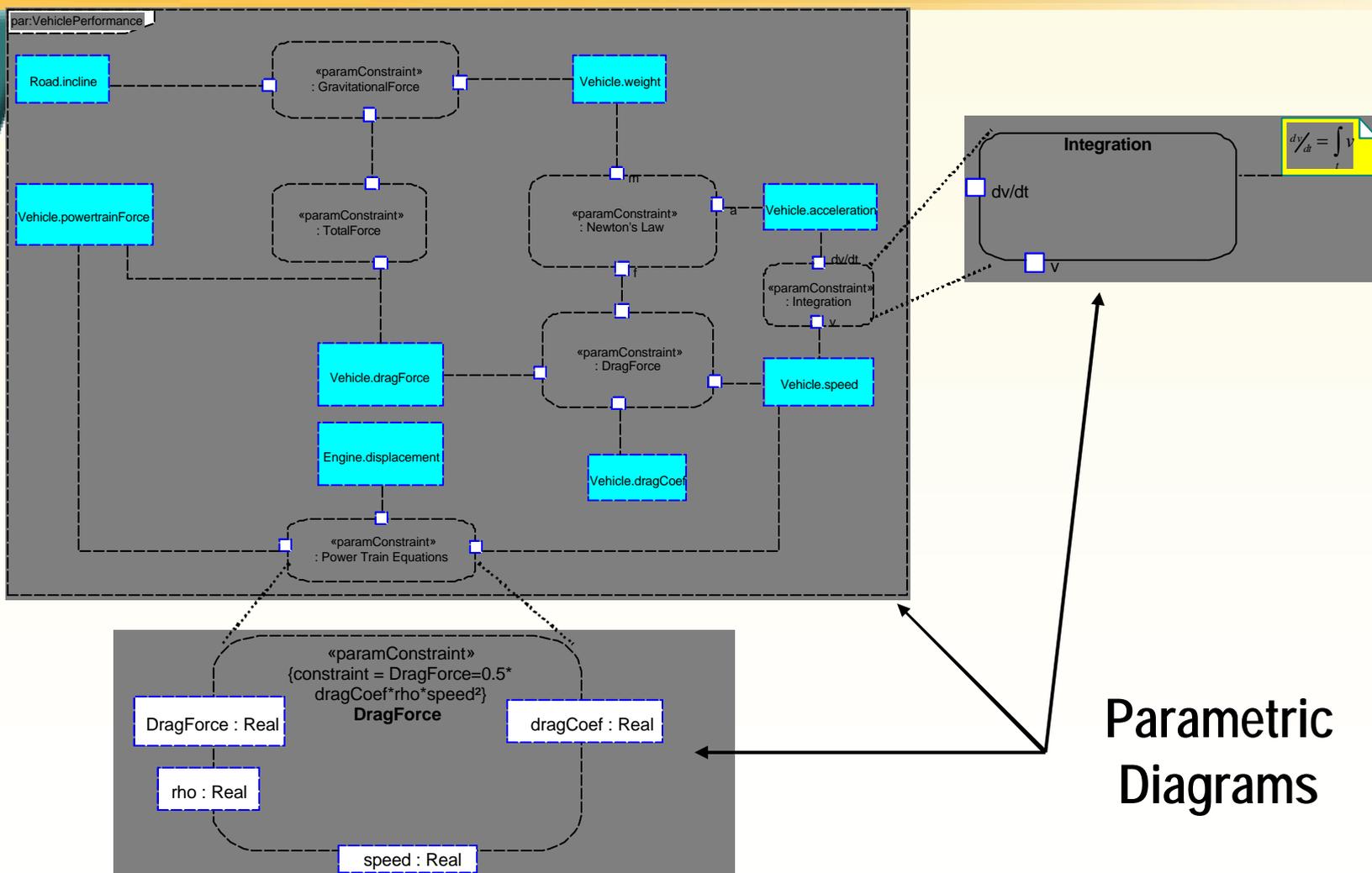
# Ausblick: SysML



«diagramDescription»  
 version="0.1"  
 description="Depicts flowdown or acceleration reqt"  
 reference="Requirements Management Database"  
 completeness="partial"

## Requirements Diagram

# Ausblick: SysML



Parametric Diagrams

- ARTiSAN im Internet:

[www.artisansw.com](http://www.artisansw.com)