
Modellgetriebene Softwareentwicklung reaktiver Systeme

Agenda

- Einleitung
- Motivation
- Begriffsklärung
 - Reaktive Systeme
 - MDA
- Umsetzung
- Ausblicke

Vorträge aus der TI

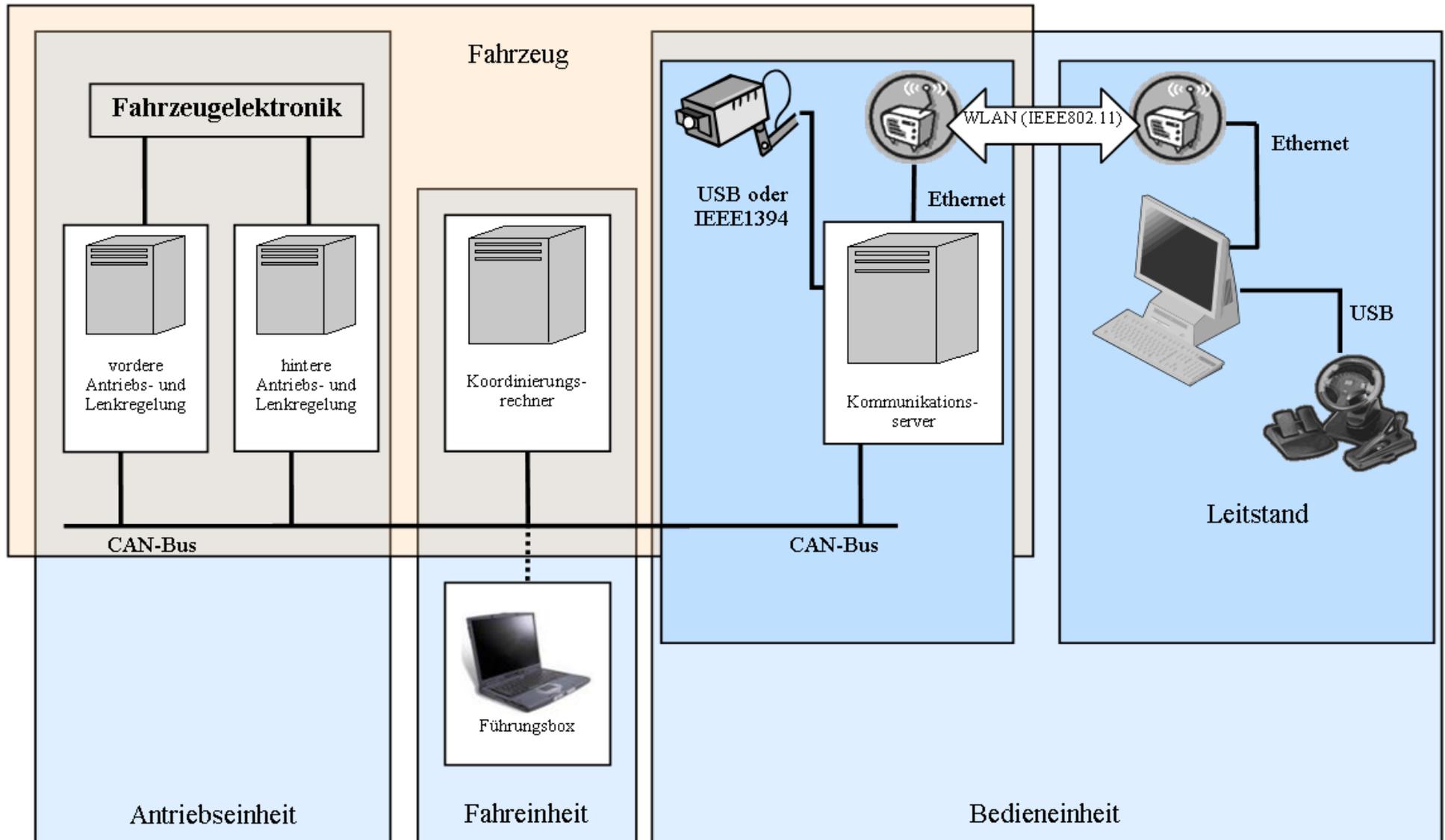
- J. Sellentin „Zeitgesteuerte Kommunikationssysteme für verteilte Hard-Real-Time Anwendungen“
- S. Cordes „Situationserkennung auf Basis eines Fahrzeugumgebungsmodells in Fahrerassistenzsystemen“
- A. Pröhl „Entscheidungsstrategien und Steuerungskonzepte eines Kollisionsvermeidungssystem“
- O. Tetzlaff „Sensordatenfusion zur Umgebungserfassung in Fahrerassistenzsystemen“

FAUST

Fahrerloses Autonomes Transportsystem



FAUST Architekturüberblick



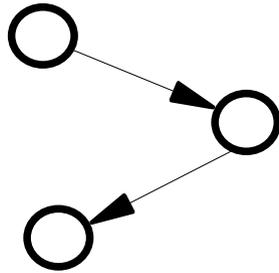
Motivation

- Den Entwicklungsprozess vereinfachen
- Entwicklungszeit für Anwendungen verkürzen
- Ein Modell muss grundsätzlich gezeichnet werden -> gute Basis für Weiterverarbeitung
- Kommerzielle Tools sind verfügbar, aber teuer und unflexibel

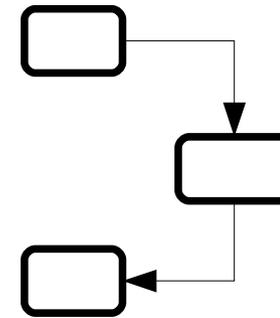
Begriffklärung

- Reaktive Systeme
 - Automaten
 - Reaktive Systeme
 - AIRA
 - Visualisieren reaktiver Systeme
- MDA (Model driven Architecture)

Automaten



- Finden Verwendung zur Beschreibung formaler Sprache.
- Finden Verwendung im Compilerbau.
- Mealy & Moore



- Zeitlich nicht definierte eingaben führen zu Zustandsübergängen
- Eingaben sind zu keiner Zeit vorhersehbar.

Reaktive Systeme

- Harel Automaten('87)
 - Verbinden die Ansätze von Mealy und Moore
 - Erweitern das Konzept um eine Hierarchie
 - Parallele Spezifikation
 - Schachtelung(Subautomaten)
 - Erweitert um Historie
- Anpassungen durch Prof. Kaltenhäuser ('01)
 - Keine Historie
 - n Startzustände im Subautomaten
 - Anforderungen an Semantik restriktiver (Abarbeitung eines Automaten)

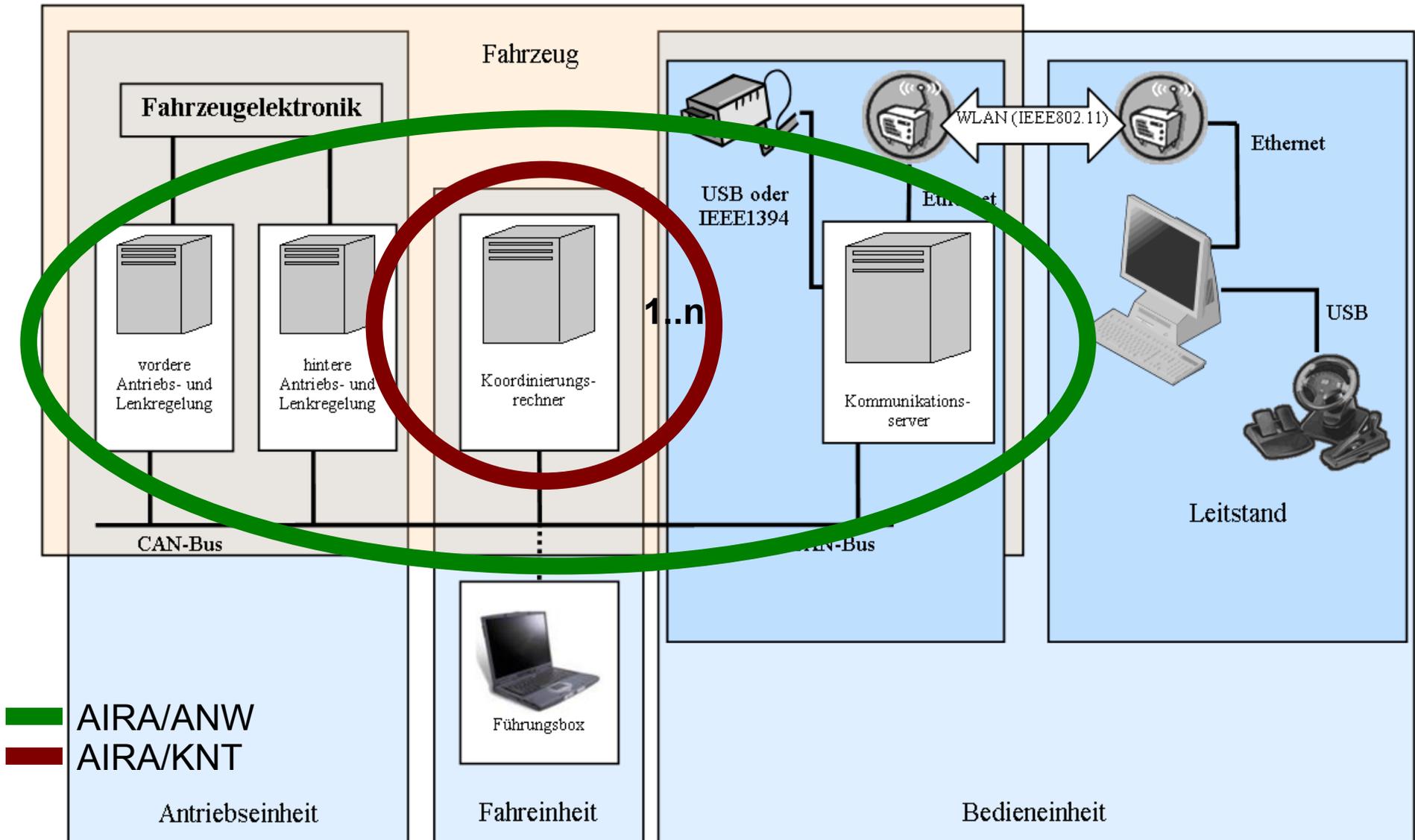
AIRA Komponenten

- **Automaten In Reaktiven Anwendungen**
- Die AIRA-Reports
- Die AIRA-Compiler
 - AIRA/KNT
 - AIRA/ANW
- Die AIRA-Laufzeitumgebung für QNX
 - Entwickelt im letzten Semester in Anw.1

AIRA Reports

- AIRA KNT / Sprache zum Beschreiben eines Kontrollers (Automaten, die einzelne Task)
- AIRA DAT / Sprache zum Beschreiben von Datentypen(-strukturen)[soll in AIRA/KNT einfließen]
- AIRA ANW / Sprache zum Beschreiben einer verteilten Anwendung(Welche Tasks laufen auf welchem Rechner, Kommunikation der Tasks)

AIRA



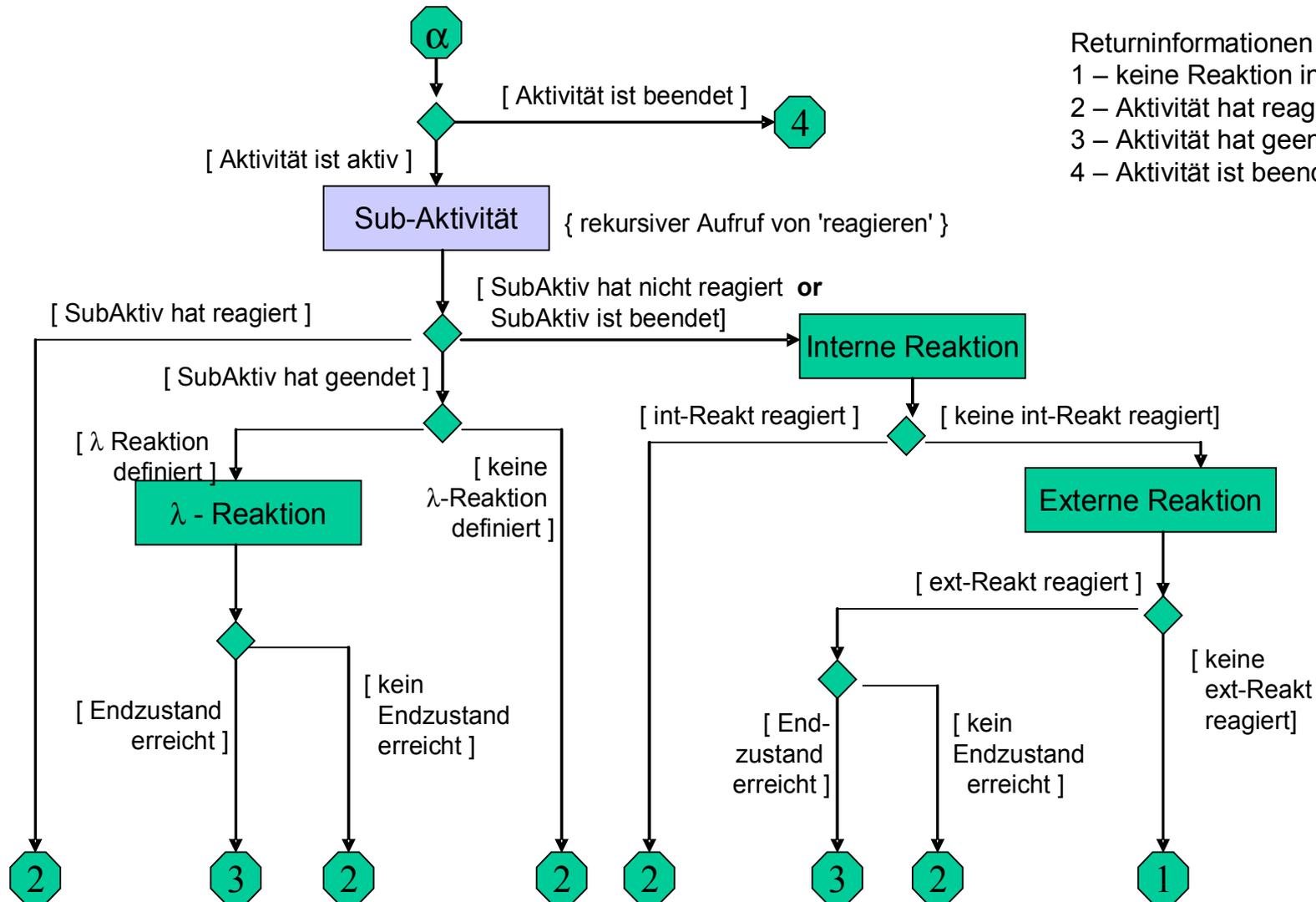
AIRA Compiler

- Implementiert in Eiffel
- Syntaxprüfung
- Erzeugen eines Parsebaumes (Eiffelobjekte)
- Semantikprüfung
- Ausgabe von Anwendungscode (C, Java, ...)

AIRA Laufzeit

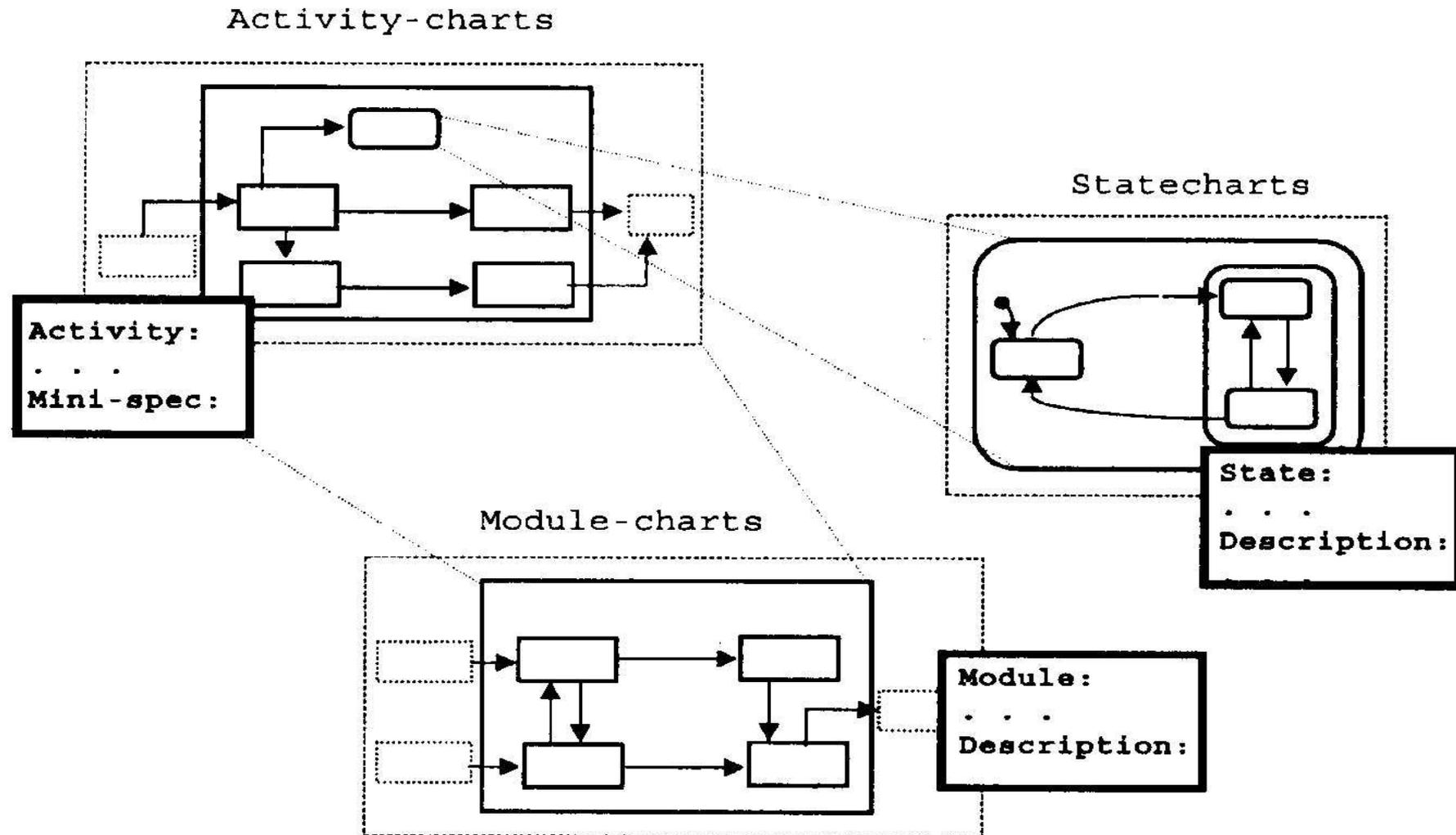
- Die Schaltlogik eines Automaten ist immer gleich...
- ...lässt sich in einer Laufzeitumgebung umsetzen.
- Das Zustandsmodell ist für jede Anwendung anders.
- Minimiert den Codeumfang bei der generierung von ausführbarem Code.

AIRA-Logik



- Returninformationen
- 1 – keine Reaktion in der Aktivität
 - 2 – Aktivität hat reagiert
 - 3 – Aktivität hat geendet
 - 4 – Aktivität ist beendet

Visualisieren reaktiver Systeme



Begriffsklärung

- Reaktive Systeme
- MDA (Model Driven Architecture)
 - Allgemein
 - MOF
 - XMI

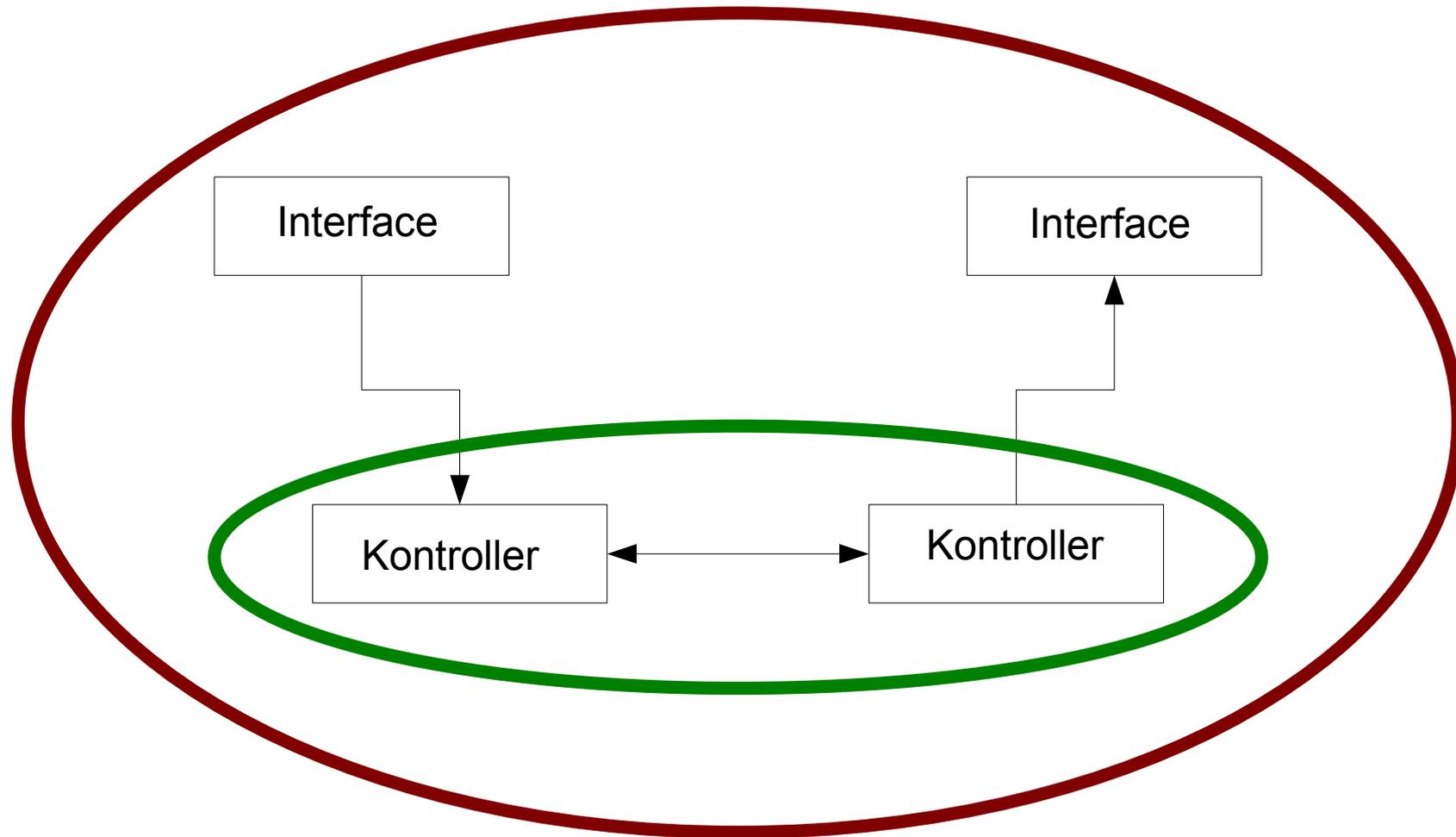
Modellgetriebene Softwareentwicklung

- Woher kennen wir das?
 - J2EE Umfeld
 - Klassendiagramme in Code transformieren.
 - ER-Diagramme direkt in DB übertragen.
- Ist das das Kernthema dieses Vortrages?
 - NEIN....
- Es geht darum, andere Diagrammtypen zur Erzeugung von Code heranzuziehen.

Modellgetriebene Softwareentwicklung

- Vom Bild zur Anwendung
- Steigerung der Entwicklungsgeschwindigkeit
- Besser handhabbar
 - Durch höhere Abstraktion
 - Bei komplexen (verteilten) Systemen
- Je nach Anforderung 20%-80% Generierung

Vollständigkeit der Generierung

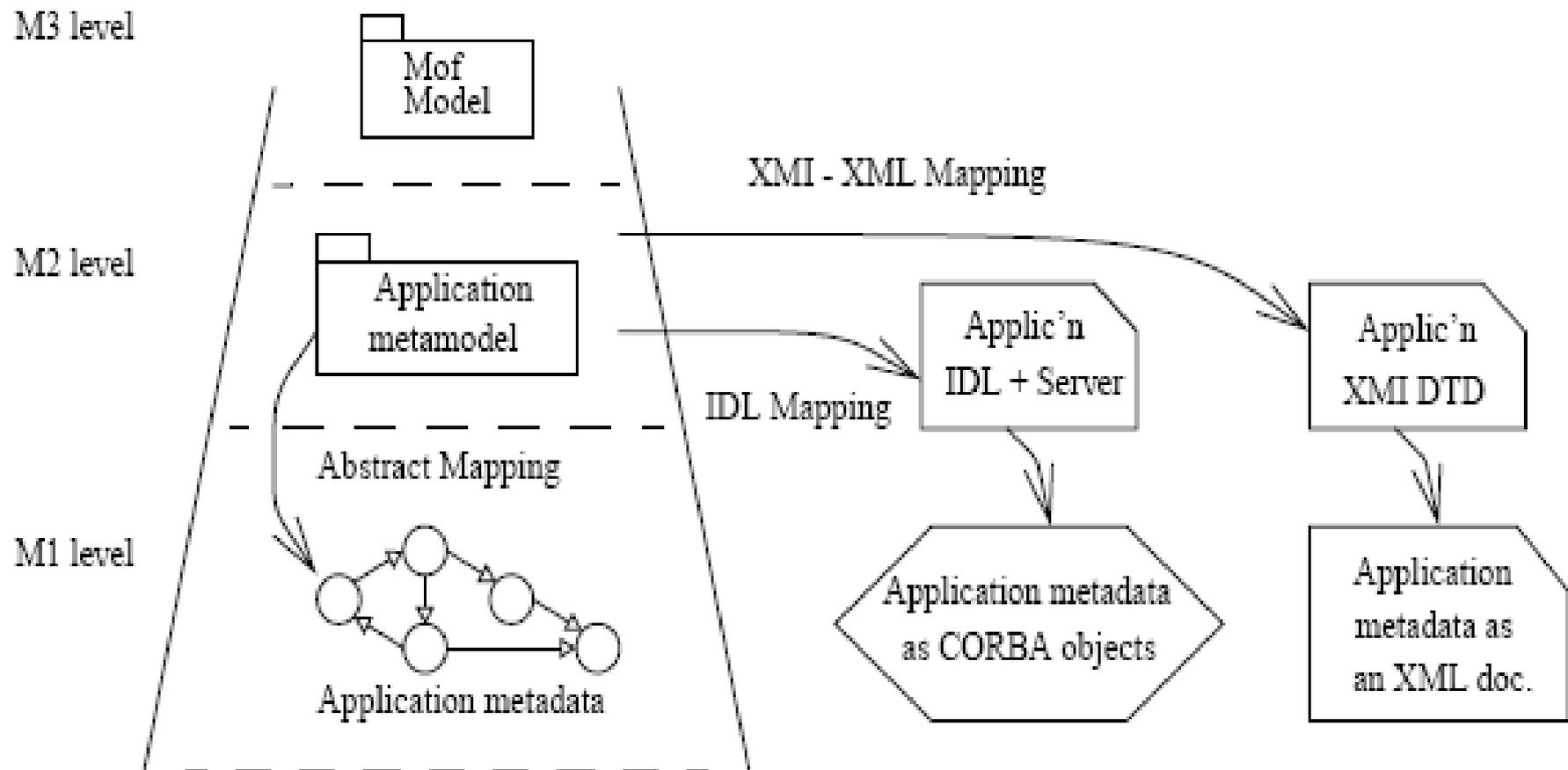


- █ 100 %?
- █ deutlich weniger

MOF - Meta-Object Facility

- Die Grundbegriffe der MOF
 - MOF 0-3(MoF-0,MoF-1,UML,Code)
 - CIM (Computation Independent Modell) für die umgangssprachliche Beschreibung
 - PIM (plattformunabhängiges Modell) für Geschäftsprozesse
 - PSM (plattformabhängiges Modell) für Architektur, Services
 - ein Codemodell, die Zielplattform
- Die Transformationssprache XMI

MOF-Level

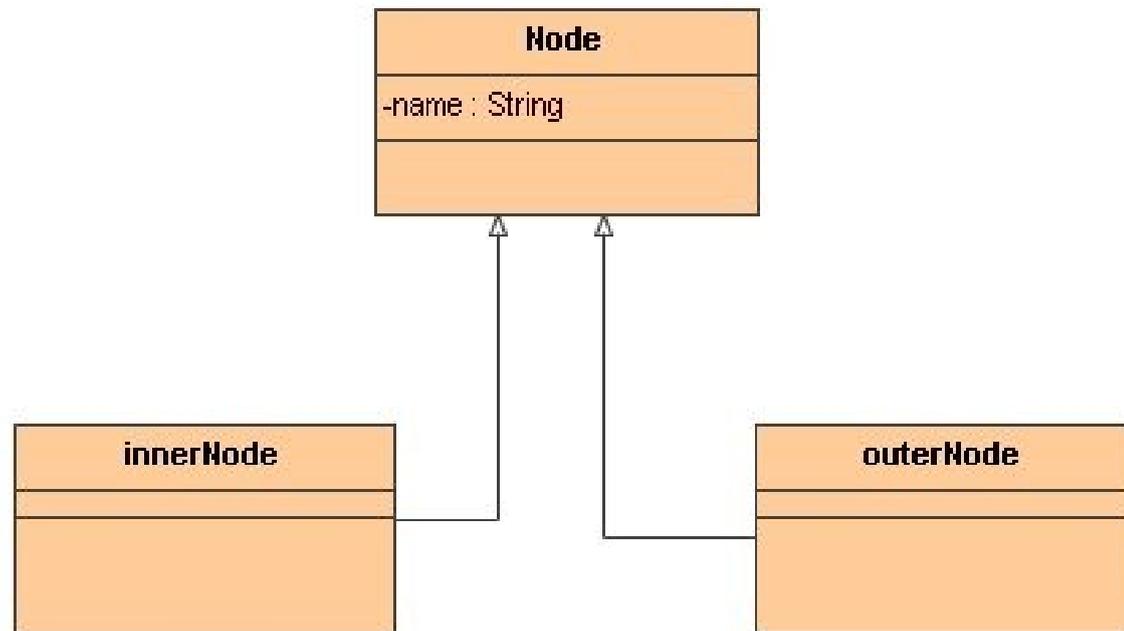


XML Metadata Interchange

- Entwickelt von der OMG
- Textuelle Beschreibung von UML-Diagrammen
- Formalismus zur Generierung von XML-Vokabularen
- XMI als zentrales Austauschformat zwischen CIM-PIM- und den MOF Levels.
- ☹ Sehr umfangreich und nicht Human-readable
- ☹ Keine Objektinformationen über das Aussehen in Tools

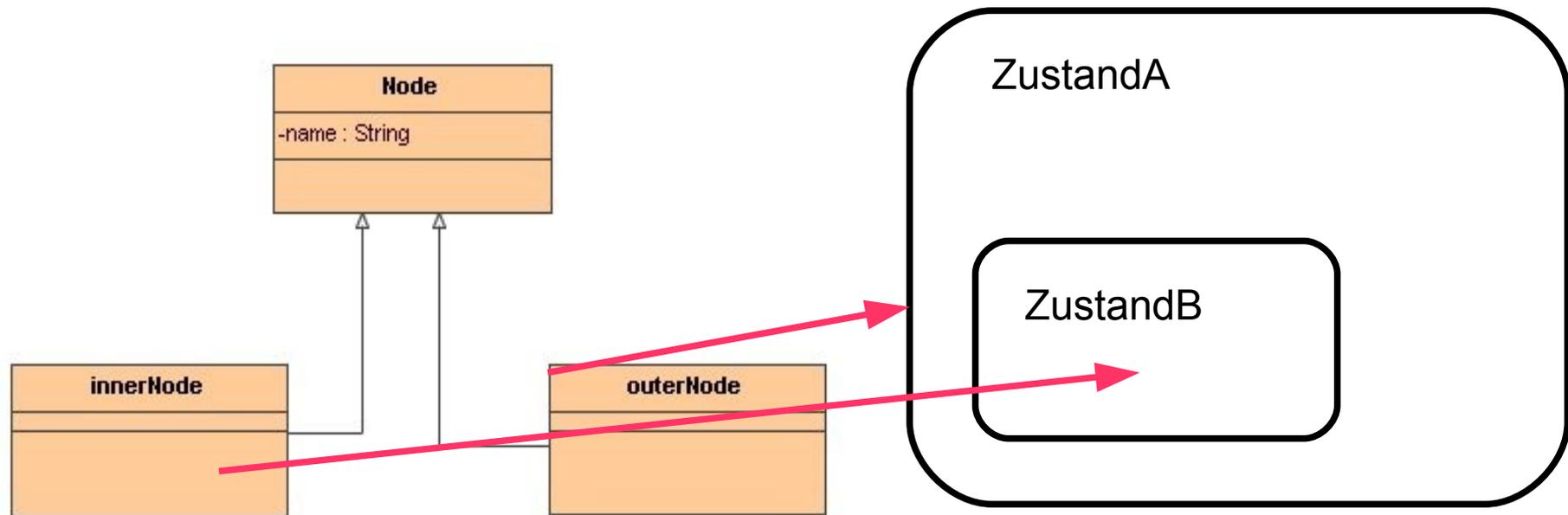
XMI Beispiel

- Erzeugen eines Metamodells
 - Definieren von Klasse



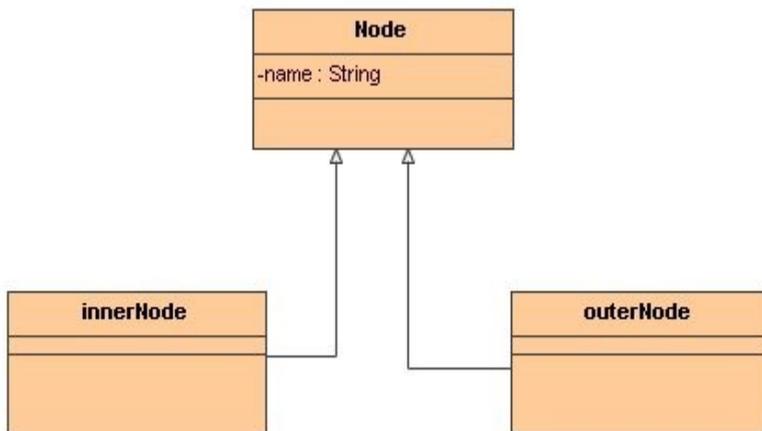
XMI Beispiel

- Instanzen der Klassen bilden



XMI Beispiel

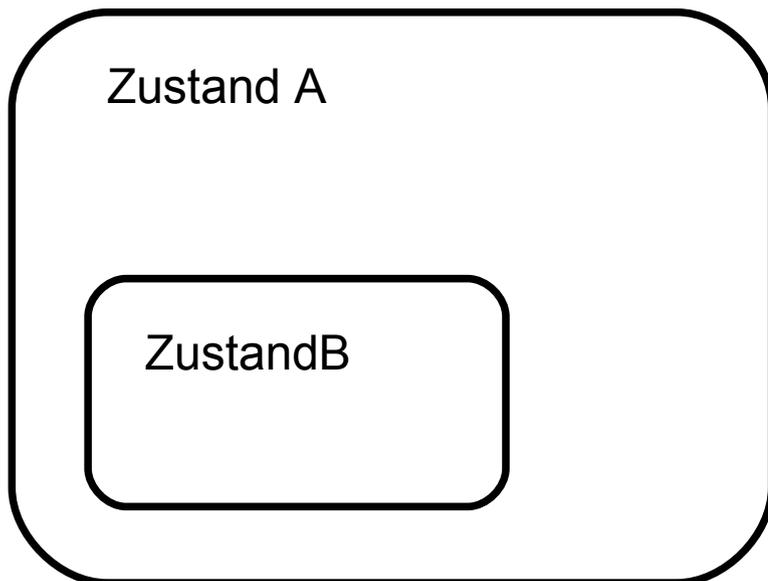
- Generieren einer DTD aus dem Metamodell



```
<?xml version="1.0" encoding="UTF-8" ?>
.
.
.
<!ELEMENT Node.name (#PCDATA | XMI.reference)* >
<!ELEMENT Node (Node.name?, XMI.extension*)? >
<!ATTLIST Node
    %XMI.element.att;
    %XMI.link.att;
>
<!ELEMENT innerNode.node (Node | innerNode | leafNode)* >
<!ELEMENT innerNode (Node.name?, XMI.extension*,
innerNode.node*)? >
<!ATTLIST innerNode
    %XMI.element.att;
    %XMI.link.att;
>
<!ELEMENT outerNode.information (#PCDATA | XMI.reference)* >
<!ELEMENT outerNode (Node.name?, leafNode.information?,
    XMI.extension*)? >
<!ATTLIST leafNode
    %XMI.element.att;
    %XMI.link.att;
>
```

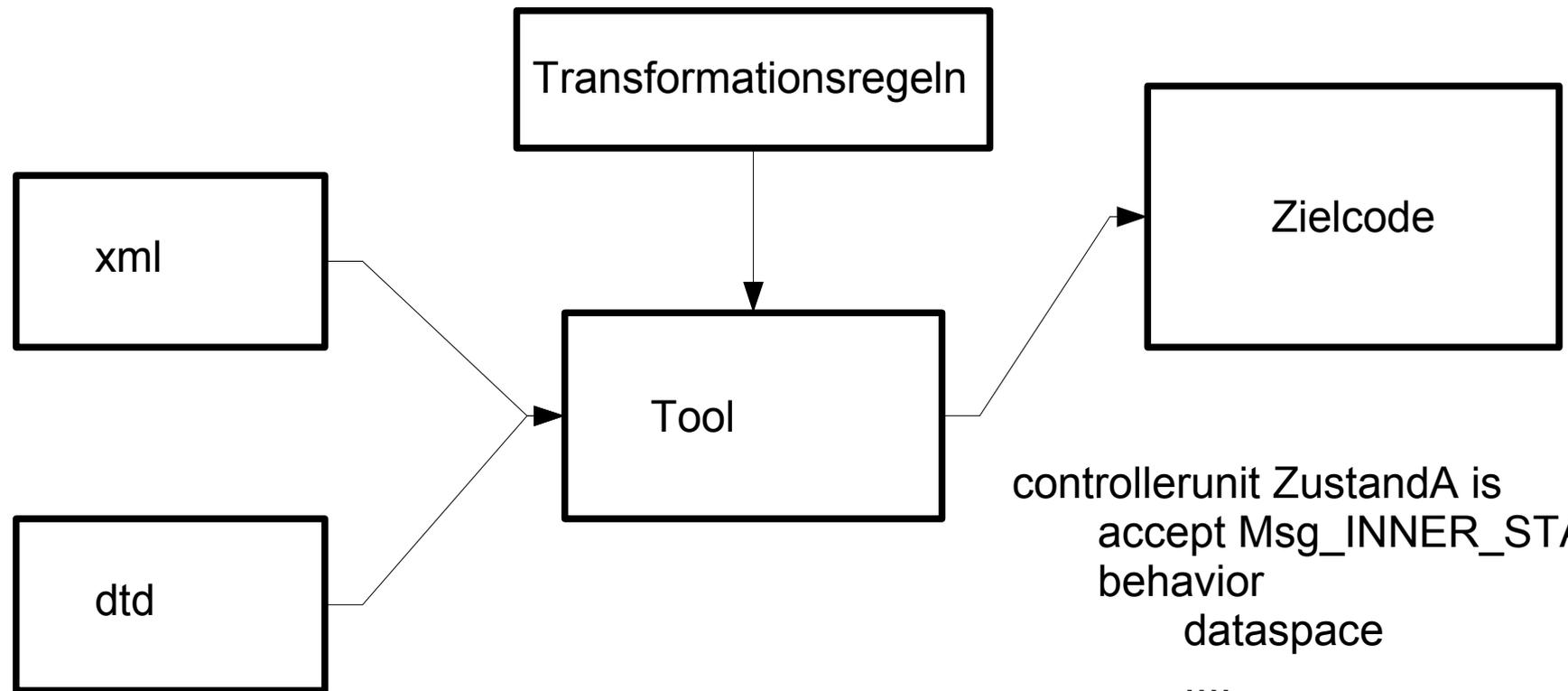
XMI Beispiel

- Erstellen eines XML-Files das zum Automaten passt



```
<?xml version="1.0"?>
<!DOCTYPE XMI SYSTEM "node.dtd">
<XMI>
  <XMI.header>
</XMI.header>
  <XMI.content>
    <outerNode>
      <Node.name>ZustandA</Node.name>
      <innerNode>
        <Node.name>ZustandB</Node.name>
      </innerNode>
    </outerNode>
  </XMI.content>
</XMI>
```

XMI Beispiel



```
controllerunit ZustandA is
  accept Msg_INNER_STATE
  behavior
    dataspace
    ....
  end dataspace
  action switchState is
    ....
  end action switchState
end behavior
end controllerunit
```

Generieren von Code

- Probleme
 - AIRA stellt besondere Anforderung an die Modellierung
 - Semantische Prüfung des Modells zur Entwicklungszeit.
- Lösungen
 - UML Profile
 - MOF (Anpassung von Werkzeuge nicht möglich)

Eclipse Modeling Framework

- EMF core Framework beinhaltet das Ecore Metamodell
 - Ermöglicht das Beschreiben von eigenen Modellen
 - Runtime Support für Modelle (inkl. Nachrichtensystem für Änderungen am Modell)
 - XMI-Serialisierung
 - Einfacher Editor zum erstellen von Instanzen des Modells
 -

EMF/GEF Lösung

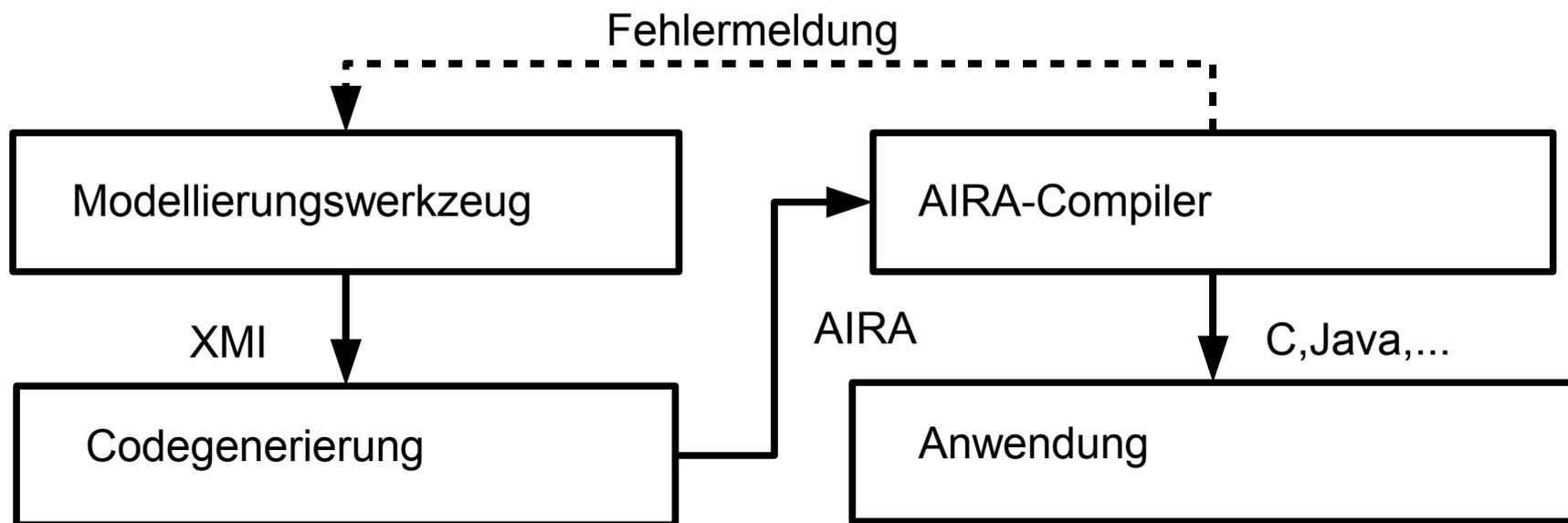
- EMF ermöglicht präzises def. eines Metamodells für AIRA
- Instanzieren von Modellen mit Hilfe des Graphical Editing Framework (GEF)
 - Grafiken und EMF Modells können miteinander verbunden werden (Statecharts!!).
- Speichern von XMI-Dateien

Tools zum Erzeugen von Code

- Tools, die XMI oder EMF verwenden um Code zu erzeugen
 - AndroMDA
 - Cartridges für AndroMDA um neue Zielsprachen zu implementieren
 - MOFScript
 - Eclipse plugin, basierend auf EMF.
 - 'Modell zu Text' Transformationswerkzeug

Modellierung von Automaten

- Erstellen von Anwendungsmodellen
- Transformieren mit XMI
- Code für Zielsystem erstellen



Ausblicke

- Evaluieren von MOF und Ecore
 - Eigenentwicklung oder
- Transformationsregelwerk erstellen
 - AndroMDA, EMF
- Prüfbarkeit von Modellen untersuchen (semantische Prüfung während des Modellierens)
- Sicherheitsprüfung von Modellen ermöglichen (Live-,Deadlock, etc.)

Litaratur

- [1] Modeling Reactive Systems with Statecharts - David Harel, Michal Politi
- [2] Model Driven Architecture – David S. Frankel
- [3] MOF 2.0/XMI Mapping Specification, v2.1 – OMG
- [4] AIRA-Reports – Prof. Dr. H Kaltenhäuser

Ende

