Möglichkeit der Darstellung von Zustandsautomaten in der IEC 61131-3

Abbildung Objektorientierter Strukturen in der IEC 61131-3

Bergische Universität Wuppertal

Lehrstuhl für Automatisierungstechnik/
Prozessautomatisierung

Autoren: Andreas Wannagat

Uwe Katzke

Birgit Vogel-Heuser

- Anforderungen der Automatisierung und der Einsatz modulare Software
- Lösungsansätze zur Integration der Objektorientierung in die IEC 61131-3
  - Statisches Modell: Abbildung von Klassen, Vererbung und Komposition
  - Verhaltensmodell: Zustandsautomaten
- Zusammenfassung und Ausblick

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

- Anforderungen
  - steigende funktionale Anforderungen an die Softwarelösung
    - hohe Komplexität
  - Zuverlässigkeit
  - Echtzeit
  - Reduktion der Kosten

- Einsatz modularer Softwarekonzepte
  - Möglichkeit zur Wiederverwendung
    - steigert Qualität der Module
    - senkt Kosten für Folgeprojekte
  - Komplexitätsreduzierend in der Anwendung
    - Abstraktion durch Kapselung
    - Geordnete Variantenbildung durch Vererbung

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

objektorientierte Programmierung

+ Erfolgreicher jahrelanger Einsatz in der Entwicklung komplexer Software
+ Klassen bündeln verschiedene Verhaltensweisen einzelner Objekte
+ Instanzen komplexer Strukturen möglich (Klassen)
+ Wiederverwendung mit Varianten durch Vererbung
– Keine Abbildung von Echtzeitaspekten

Konzept der IEC 61131-3

+ Etablierter Standard in der Automatisierungstechnik
+ Gute Abbildbarkeit von Anlagenverhalten durch Zustandsorientierung
+ Definiertes Zeitverhalten
– Geringe Möglichkeit zur geordneten Variantenbildung

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

objektorientierte Programmierung

+ Erfolgreicher jahrelanger Einsatz in der Entwicklung komplexer Software
+ Klassen bündeln verschiedene Verhaltensweisen einzelner Objekte
+ Instanzen komplexer Strukturen möglich (Klassen)
+ Wiederverwendung mit Varianten durch Vererbung
– Keine Abbildung von Echtzeitaspekten

Konzept der IEC 61131-3

+ Etablierter Standard in der Automatisierungstechnik
+ Gute Abbildbarkeit von Anlagenverhalten durch Zustandsorientierung
+ Definiertes Zeitverhalten
– Geringe Möglichkeit zur geordneten Variantenbildung

Ansatz: Objektorientierung in der IEC 61131-3

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Wie lassen sich objektorientierte Konzepte in die IEC 61131-3 integrieren?

Eine Möglichkeit: Codegenerator

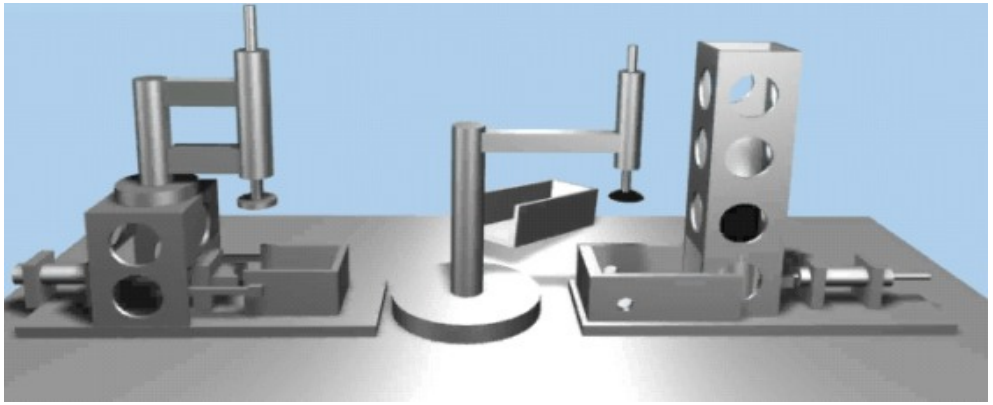
- Einsatz der UML als Modellierungsumgebung
- Laufzeitumgebung bleibt IEC 61131 konform

+ Verständlichkeit / Kommunikationsfähigkeit
+ bessere Wartbarkeit
+ Geringere Komplexität durch Erhöhung des Abstraktionsniveaus

Realisierbarkeit wurde am Lehrstuhl für Automatisierungstechnik in Wuppertal gezeigt

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat



Modellierung

Software-
Projektierung

Hardware-
Projektierung

UML - Modell

Systemteil

Software -Teil

OLE

XML-Export

Codegenerator

AWL–Text Import

XML-Import
PLC-Open

IEC-61131-3

CoDeSys 3.0

Step 7

softing

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Statisches Modell

Abbildung von Klassen, Vererbung und Komposition in der IEC-61131-3

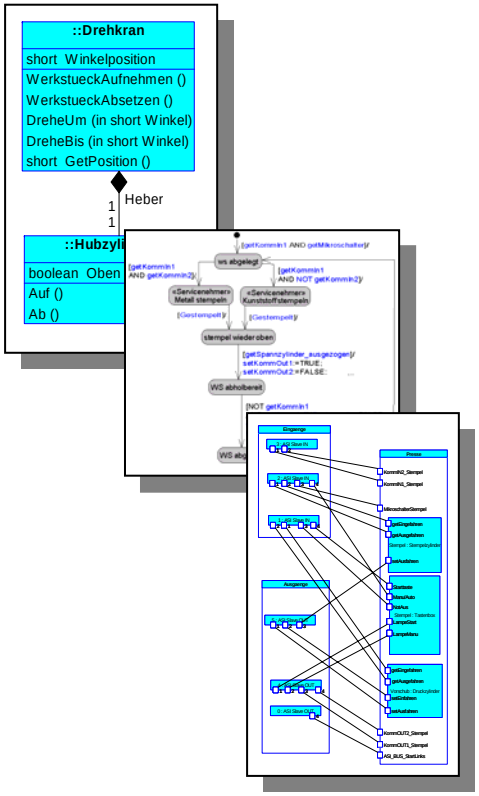
Verhaltensmodell

Zustandsautomaten → Ablaufsprache

Implementierungsmodell

Mapping von Softwareobjekten auf CPUs

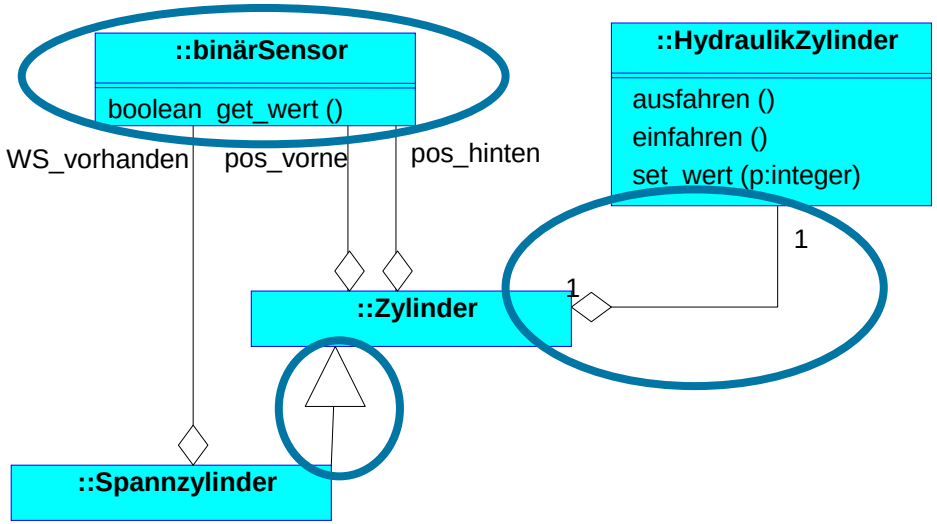
Mapping von Hardwareobjekten (Sensoren und Aktoren)



Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Klassendiagramm in der UML



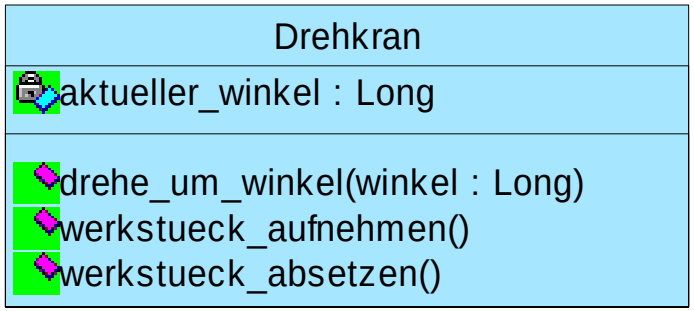
Klassen als Kombination aus STRUCT und Funktionsbaustein

Vererbung

Aggregation / Komposition

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat



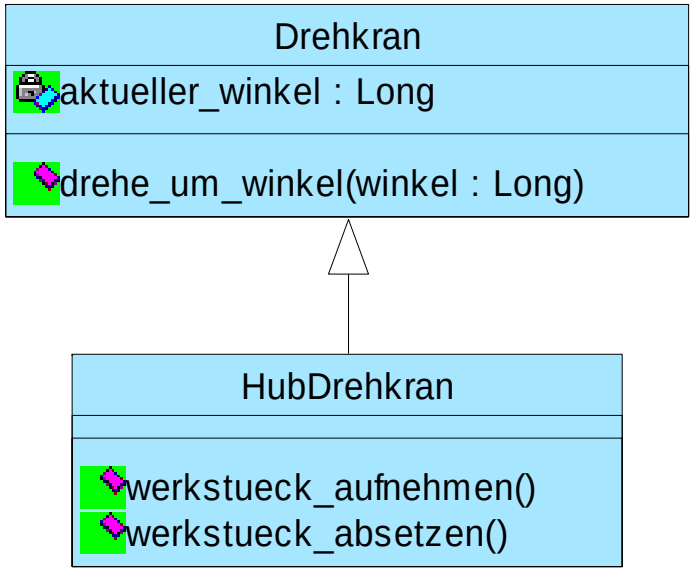
- Innerhalb von **Funktionsbausteinen** lassen sich **Variablenstrukturen** aufbauen und ein **Ablauf** definieren.
- **Verschiedene Abläufe**, wie sie durch Operationen von Klassen gekennzeichnet sind, sind allerdings in Funktionsbausteinen ***nicht weiter zu kapseln***.
- Variablenstrukturen (STRUCT): können Variablen und Funktionsbausteine aufnehmen und instanziieren

```
FUNCTION_BLOCK _Drehkran__drehe_um_winkel
VAR_INPUT
   winkel : DINT;
END_VAR
VAR_IN_OUT
   aktueller_winkel : DINT;
END_VAR;

END_FUNCTION_BLOCK;
FUNKTION_BLOCK _Drehkran__werkstueck_aufnehmen
END_VAR
VAR_IN_OUT
   aktueller_winkel : DINT;
END_VAR;
END_FUNCTION_BLOCK;

FUNCTION_BLOCK _Drehkran__werkstueck_absetzen
END_VAR
VAR_IN_OUT
   aktueller_winkel : DINT;
END_VAR;
END_FUNCTION_BLOCK;

STRUCT Drehkran
   aktueller_winkel : DINT;
   drehe_um_winkel: _Drehkran__drehe_um_winkel;
   werkstueck_aufnehmen: _Drehkran__werkstueck_aufnehmen;
   werkstueck_absetzen: _Drehkran__werkstueck_absetzen;
END_STRUCT;
```


Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Statisches Modell - Vererbung



- Vererbung durch Kopie und Erweiterung der Struktur
- Vererbung durch Übernahme vorhandener Funktionen und Strukturen

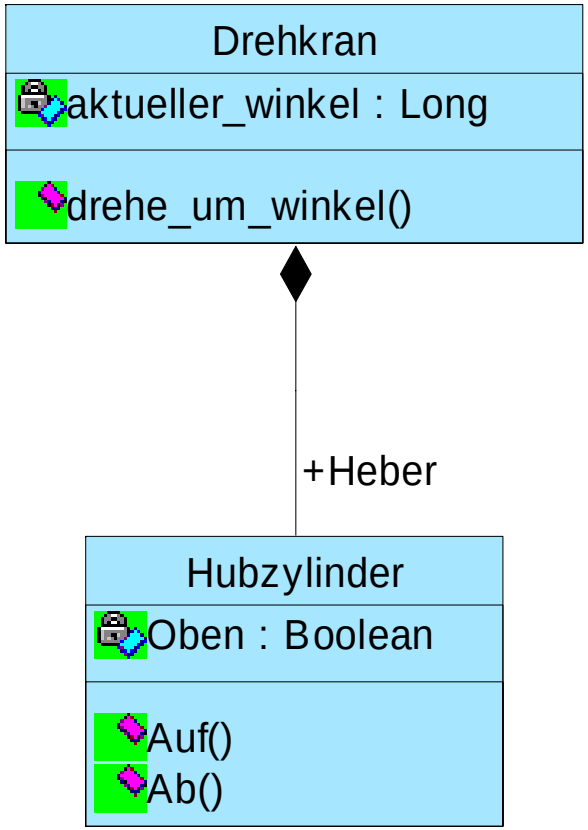
```
STRUCT Drehkran
   aktueller_winkel: DINT;
   drehe_um_winkel:
_Drehkran__drehe_um_winkel;
END_STRUCT;
```

```
FUNCTION_BLOCK _Drehkran__drehe_um_winkel
VAR_INPUT
   winkel : DINT;
END_VAR
VAR_IN_OUT
   aktueller_winkel : DINT;
END_VAR;
END_FUNCTION_BLOCK;

FUNCTION_BLOCK _HubDrehkran__werkstueck_aufnehmen
END_VAR
VAR_IN_OUT
   aktueller_winkel : DINT;
   drehe_um_winkel : _Drehkran__drehe_um_winkel
END_VAR;
END_FUNCTION_BLOCK;

FUNCTION_BLOCK _HubDrehkran__werkstueck_absetzen
END_VAR
VAR_IN_OUT
   aktueller_winkel : DINT;
   drehe_um_winkel : _Drehkran__drehe_um_winkel;
   werkstueck_aufnehmen: _HubDrehkran__werkstueck_aufnehmen
END_VAR;
END_FUNCTION_BLOCK;

STRUCT HubDrehkran
   aktueller_winkel : DINT;
   drehe_um_winkel: _Drehkran__drehe_um_winkel;
   werkstueck_aufnehmen: _HubDrehkran__werkstueck_aufnehmen;
   werkstueck_absetzen: _HubDrehkran__werkstueck_absetzen;
END_STRUCT;
```


Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

```
STRUCT Hubzylinder
   Oben: BOOL;
   Auf: _FB_Hubzylinder_Auf;
   Ab:  _FB_Hubzylinder_Ab;
END_STRUCT;


STRUCT Drehkran
   winkel: DINT;
   drehe_um_winkel: _Drehkran__drehe_um_winkel;
   Heber: Hubzylinder;
END_STRUCT;
```

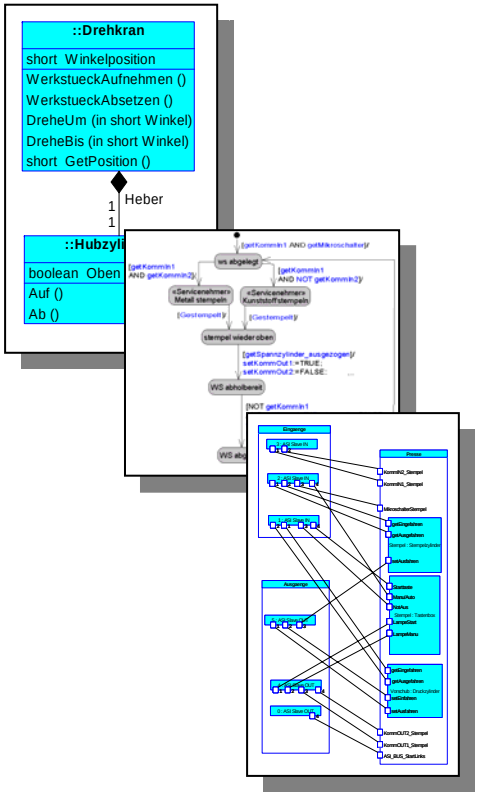


Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Statisches Modell

Abbildung von Klassen, Vererbung und Komposition in der IEC-61131-3



Verhaltensmodell

Zustandsautomaten → Ablaufsprache

Implementierungsmodell

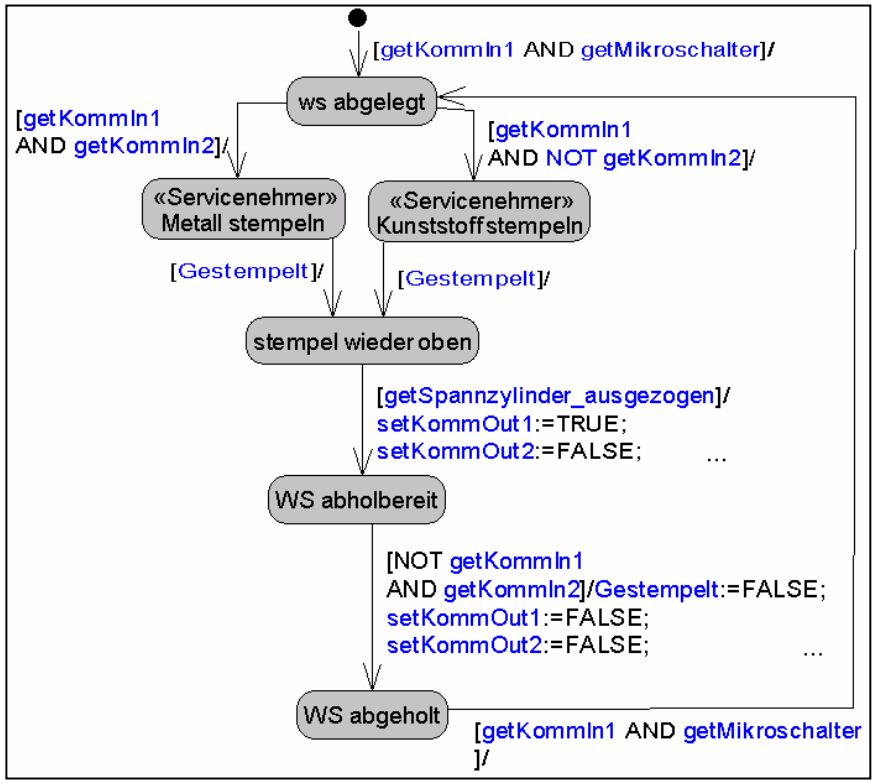
Mapping von Softwareobjekten auf CPUs

Mapping von Hardwareobjekten (Sensoren und Aktoren)

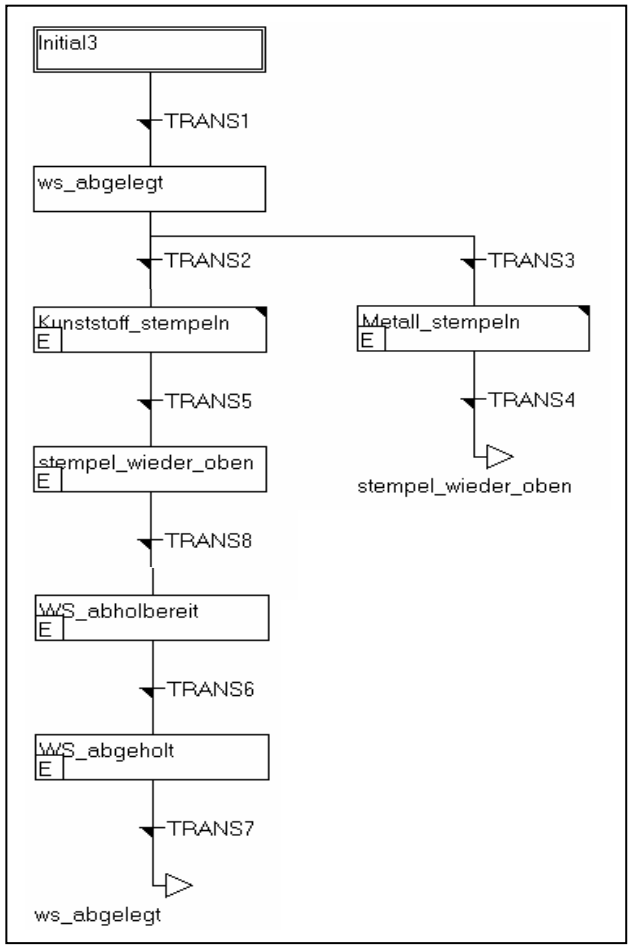
Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

UML - Statecharts



Ablaufsprache



- ähnlicher Aufbau
- AS: keine Aktionen an den Transitionen
- UML: Unterscheidung zwischen Aktionen und Aktivitäten

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

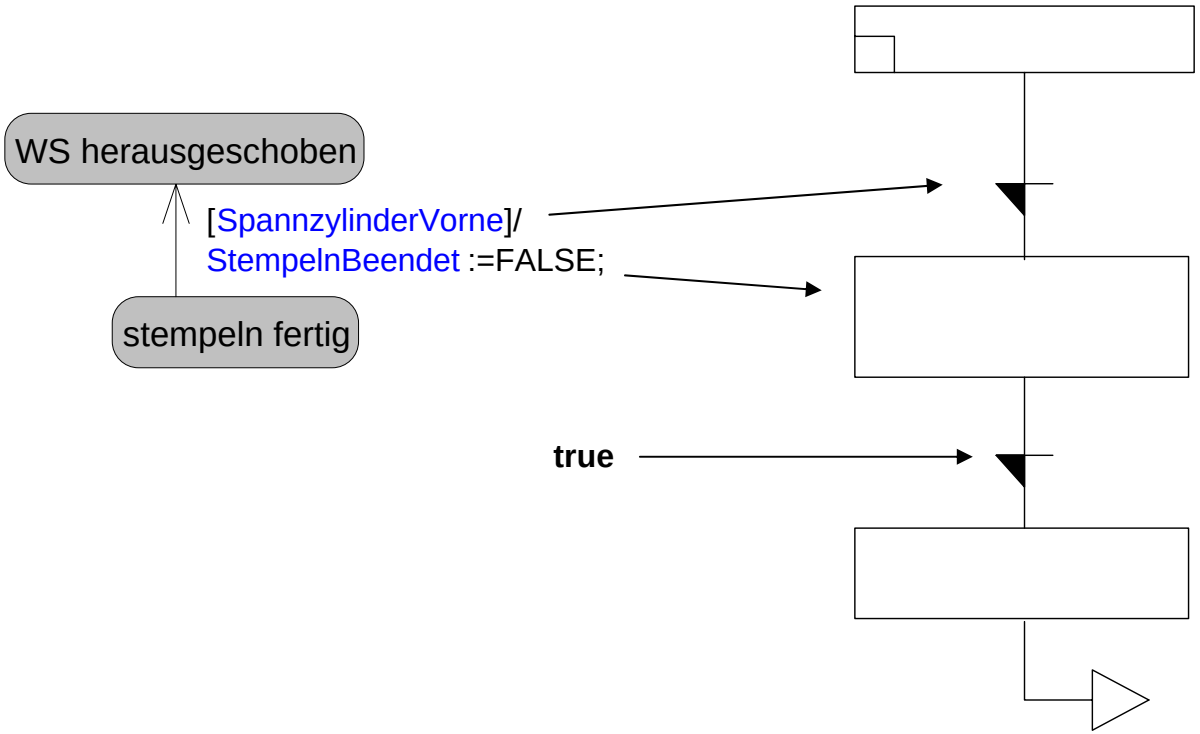


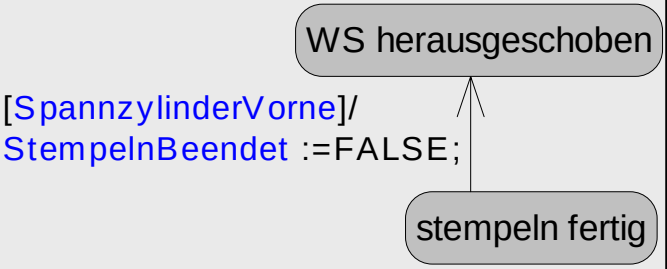
Abbildung von Aktionen mit Hilfe eines Zwischenzustandes

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Verhaltensmodell – Zwischenzustände in textueller Form

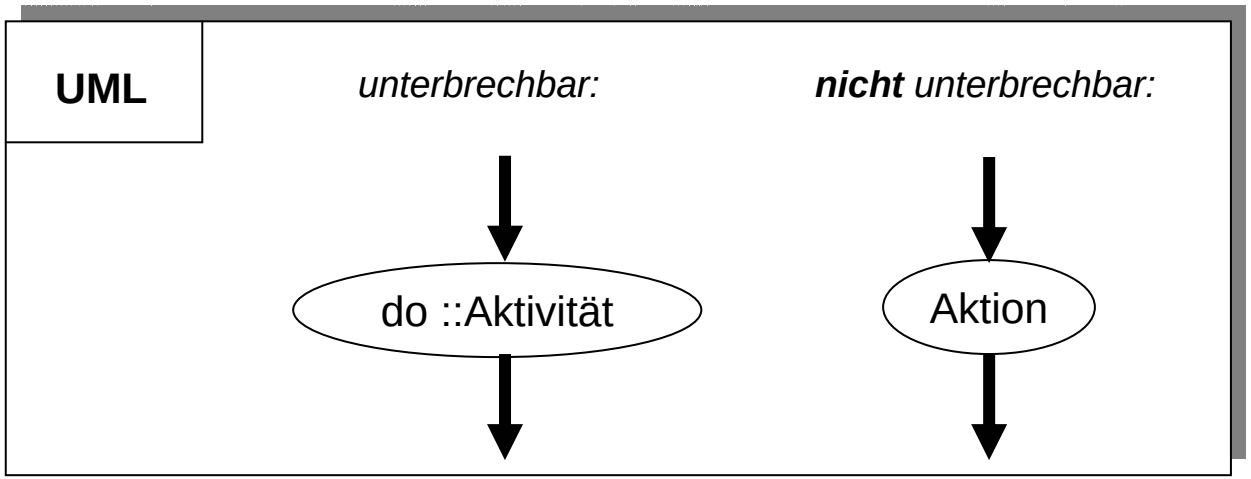
```
STEP stempeln_fertig:
END_STEP

TRANSITION TRANS1 FROM stempeln_fertig TO
   BETWEEN_stempeln_fertig_WS_herausgeschoben
   :=SpannzylinderVorne
END_TRANSITION

STEP BETWEEN_stempeln_fertig_WS_herausgeschoben:
   StempelnBeendet :=FALSE;
END_STEP

TRANSITION TRANS2 FROM
   BETWEEN_stempeln_fertig_WS_herausgeschoben  TO WS_herausgeschoben := TRUE
END_TRANSITION

STEP WS_herausgeschoben:
END_STEP
```



Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Die UML unterscheidet zwei Fälle

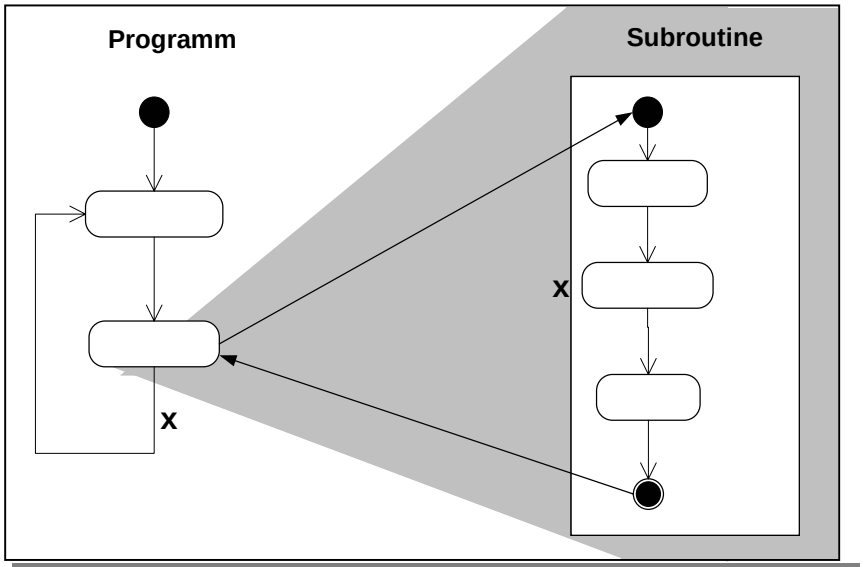


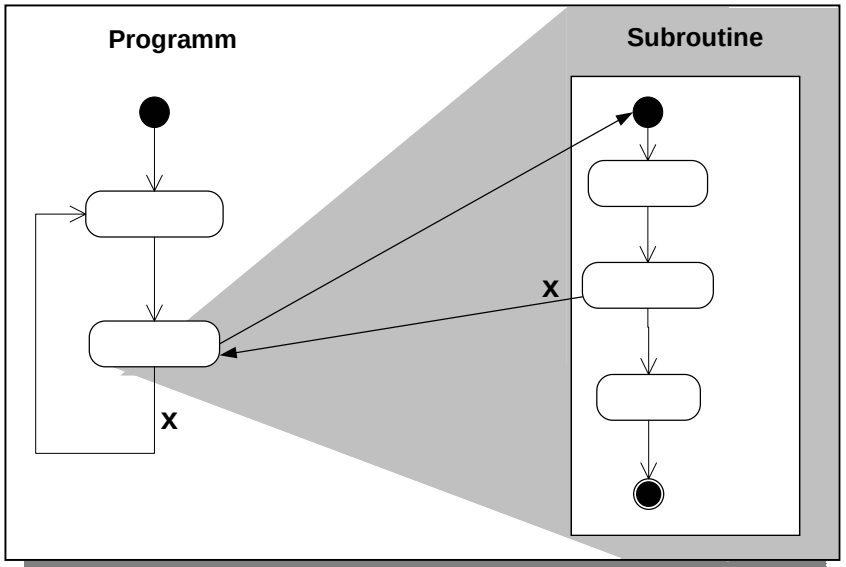
Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de
Dipl.-Ing. Andreas Wannagat

- **Hierarchische Zustandstransitionssysteme:**
  - Zustände können wiederum weitere Zustandstransitionssysteme enthalten.
  - Wird ein Zustand erreicht, so wird eine Methode mit einem eigenen Verhaltensmodell gestartet.
- **Problematik:**
  - Wie mit diesem unterlagerten Systemen zu verfahren ist, wenn eine Weiterschaltbedingung für das übergeordnete System erfüllt ist?

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat





Konventionelle Programmiersprachen

- Abläufe durch Anweisungen, Verzweigungen und Subroutinen
- Subroutinen unterbrechen den sie aufrufenden Ablaufsstrang und setzen ihn erst nach Abschluss fort.
- Unterbrechungen einer Subroutine müssen explizit zugelassen werden.
  →Exception Handling

IEC 61131-3 (Ablaufsprache)

- Zustandsorientierung durch eigene Darstellungsform
- Möglichkeit präzise das Verhalten von hierarchischen Diagrammen zu beschreiben
- Subroutine wird durch Transitionsbereitschaft des Hauptprogramms unterbrochen

{Aufruf}

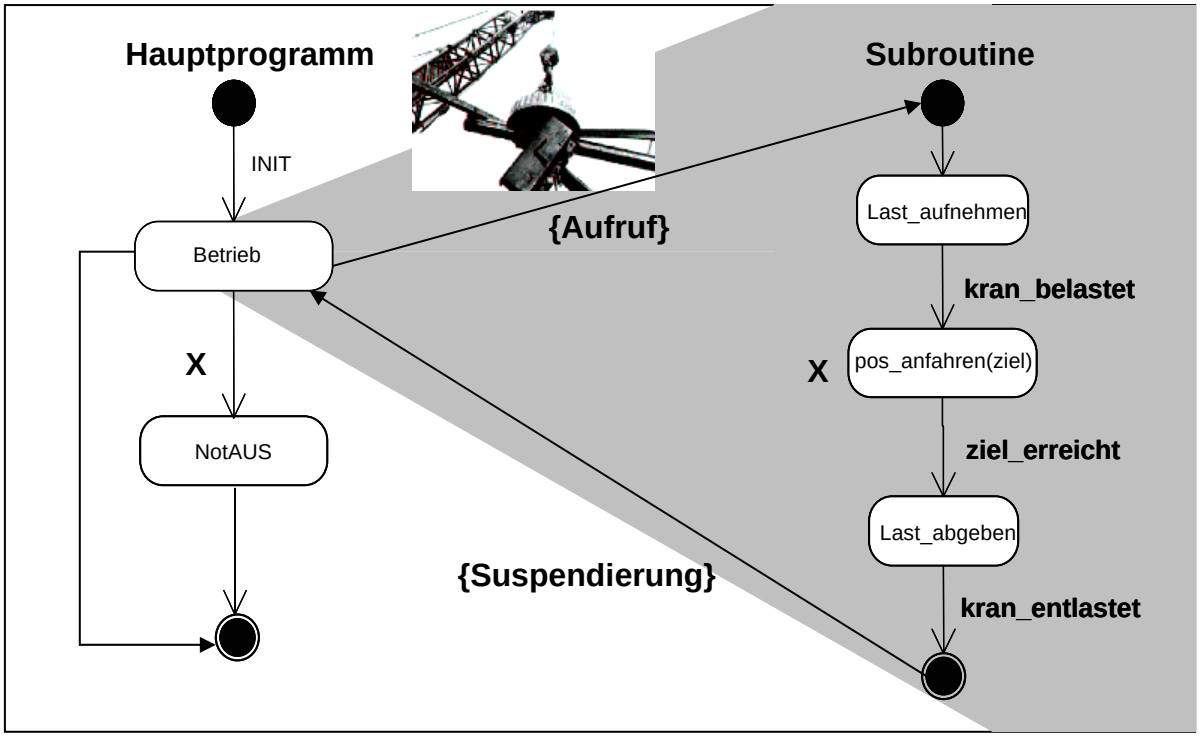
ziel_erreicht

Auftrag_ausführen

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

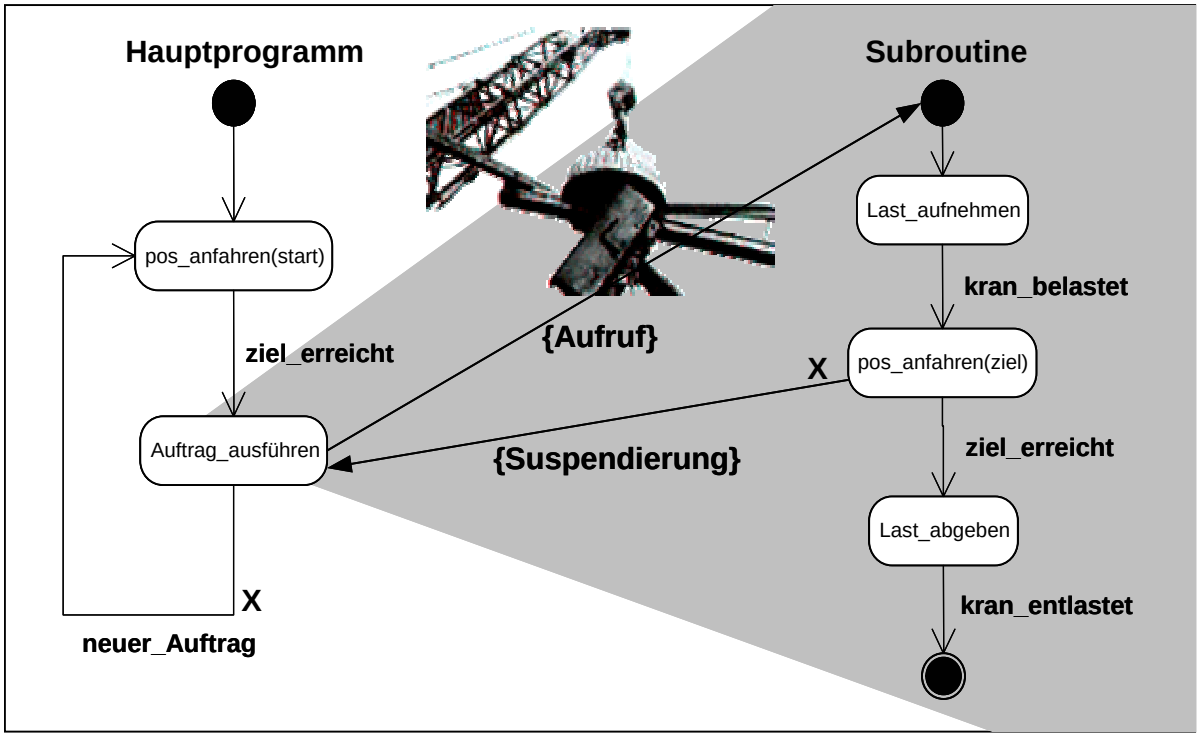
Dipl.-Ing. Andreas Wannagat

Möglichkeit 1: Subroutine gibt die Kontrolle erst zurück, wenn sie abgearbeitet ist.



+ eindeutig bestimmtes Verhalten
- keine Möglichkeit im Hauptprogramm den Zustand zu wechseln

Möglichkeit 2: Subroutine wird ausgeführt, solange der Zustand im Hauptprogramm aktiv ist



+ zeitliche Anforderungen
– kann zu einem unvorhersehbaren Zustand führen (Wiederaufruf)

Verhalten in den Implementierungen von CoDeSys und Step7

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

- besondere Eigenschaften von FB und FC:
  - Funktionsbausteine werden instanziiert und sind unterbrechbar.
  - Unterbrechbarkeit ist hinderlich, sobald diese Operationen Aktionen und nicht Aktivitäten darstellen sollen.
  - Funktionen werden nicht unterbrochen und eignen sich daher zur Beschreibung von Aktionen. → Instanzenbildung fehlt

  - Um die gewünschten Eigenschaften von Funktionsbaustein und Funktion zu vereinen, ist es möglich eine Funktion innerhalb eines Funktionsbausteins aufzurufen.

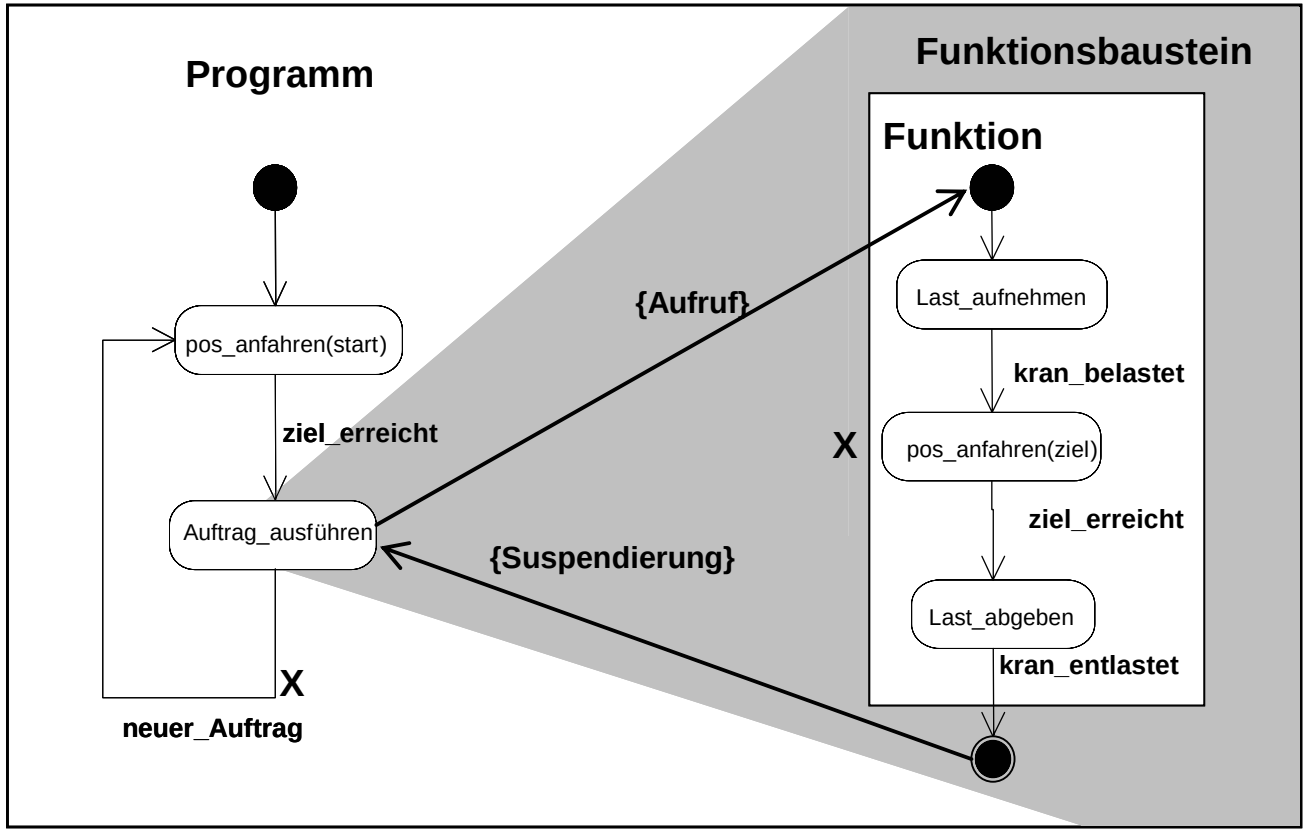
    ✓ Abgeschlossenheit einer Klasse und ihrer Datenstrukturen

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Kapselung innerhalb des Funktionsbaustein durch eine Funktion

- Realisierung nicht unterbrechbarer Subroutinen
- Kann in einer Klasse (als Einheit von Daten und Code) implementiert werden



Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat

Es konnte gezeigt werden,

daß es möglich ist verschiedene Konstrukte aus der Objektorientierung in der IEC 61131-3 abzubilden

- Klassen
- Vererbung
- Komposition / Aggregation

Gleiches gilt für das Verhalten mit

- Zuständen
- Transitionen
- Aktionen (an Transitionen und Zuständen)
- Aktivitäten

Weitere und zum Teil schon begonnene Aufgaben sind:

- Erweiterung des Codegenerators um die vorgestellten Mechanismen
- Breitere Unterstützung verschiedener Zielumgebungen durch PLC-open konforme XML
- Rückfluß von Debugging-Möglichkeiten in das Modell

Automatisierungstechnik / Prozessinformatik
Professor Dr.-Ing. Birgit Vogel-Heuser
Bergische Universität Wuppertal
www.lfa.uni-wuppertal.de

Dipl.-Ing. Andreas Wannagat