

# CC-Type- und CC-Instance- Teilmodelle

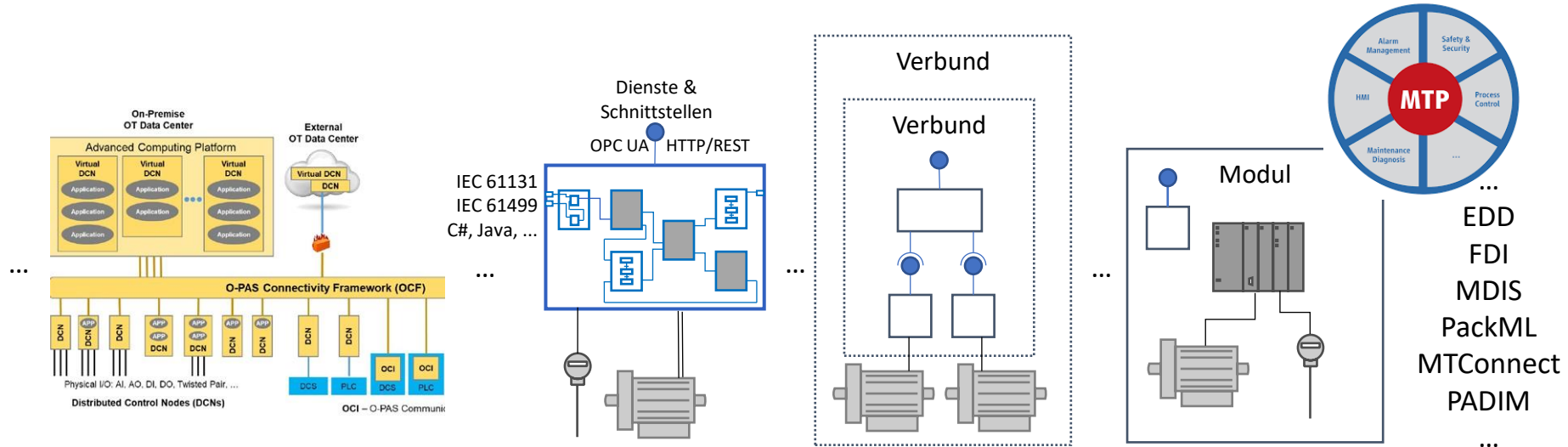
Vorschlag zur Teilmodell-Definition an die Plattform I4.0

GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung

# Begriff Control Component (CC)



Führungsfunktionalität\* wird häufiger in Software-Komponenten modularisiert.

**Aber:** Interoperabilität?

1. Schritt: Einheitliche Bezeichnung zur Begriffsbildung

→ (Prozess-)Führungskomponenten (PFK) / Control Components (CCs)

\*Steuern,  
Regeln,  
diskret &  
kontinuierlich

# Motivation der Control Component

In verschiedenen Produkt- und Anwendungsdomänen, Hierarchieebenen der Prozessführung, sowie Standards finden sich ähnliche Definitionen für Schnittstellen und deren Verhaltensmuster.

Start, Stop, Reset,  
Betriebsartwechsel,  
...

Betriebszustand,  
Betriebsart,  
...

Prozeduren

Regler,  
MPC, APC

Prozess-  
Simulation

Operations

Schnittstellen-  
verhalten  
+  
Fähigkeiten

Status

→ Beschreibung, Abstraktion  
und Vereinheitlichung durch  
CCs für eine hohe  
syntaktische und  
semantische Interoperabilität



Mobile  
Maschinen



Maschinen,  
Anlagenteile,  
Module



Aktorik



Robotik

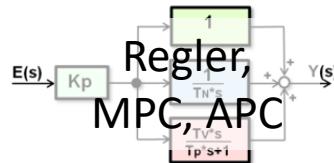
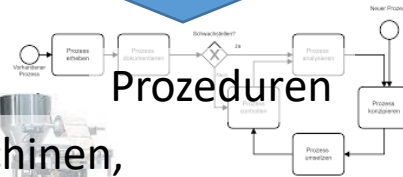


Aktorik

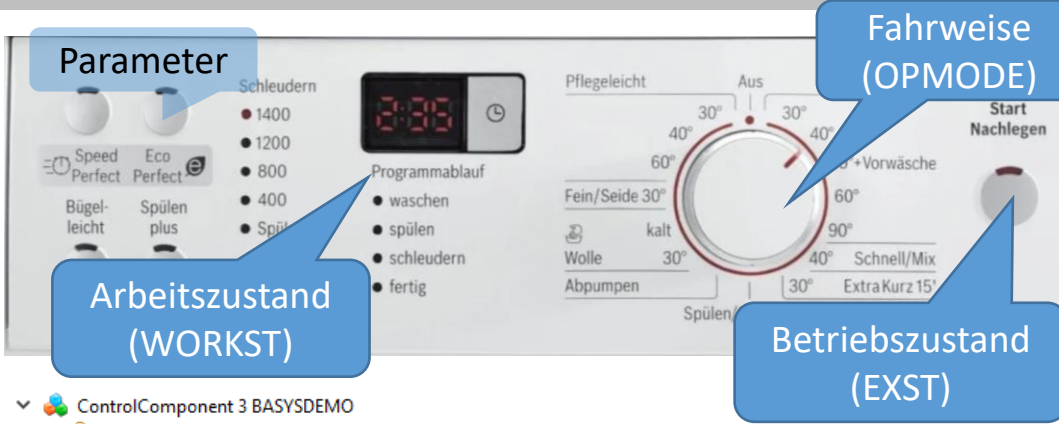


Logistik

...

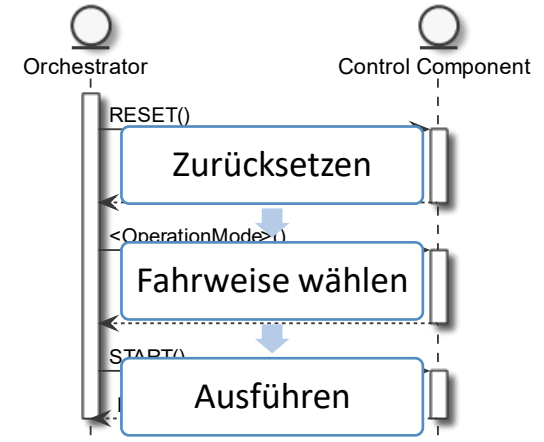
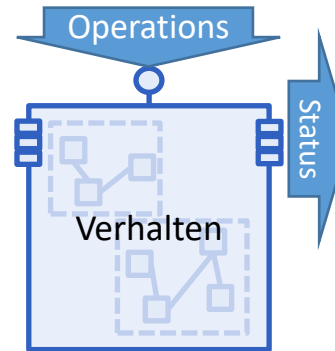
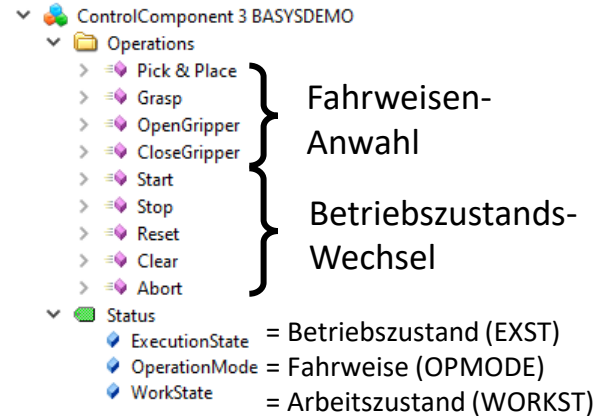


# Motivation der Control Component



Warum kann jeder eine Waschmaschine bedienen?

- Schnittstelle & Verhalten sind klar:
    - Wo (Endpunkt der Schnittstelle)
    - muss ich was (Syntax & Semantik)
    - in welcher Reihenfolge (Sequenz) drücken?
- Wie lässt sich das für verschiedenste industrielle Betriebsmittel vereinheitlichen?



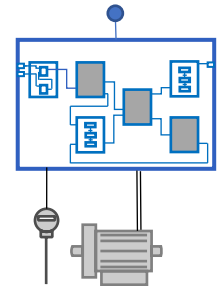
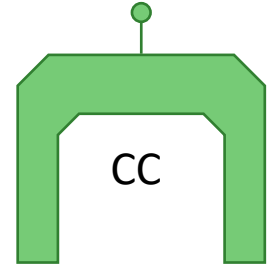
# Was wollen wir „verwalten“?

→ Was kann die CC und wie kann ich das Nutzen?

- Typ-spezifische Informationen, z.B.:
  - Was können alle CCs dieses Typs (Fahrweisen / Skills)?
  - Wie spreche ich mit der CC (Schnittstellen und Protokolle)?
  - Welche Fehlercodes gibt es? Was bedeuten sie?
- Instanz-spezifische Informationen
  - Wie erreiche ich diese CC (Endpunkt)?
  - Was ist anders zum Typ (überschreiben)? Was kann diese eine Instanz besonderes (Fahrweisen / Skills)? Z.B. spezielles Schrauber / Roboter Programm

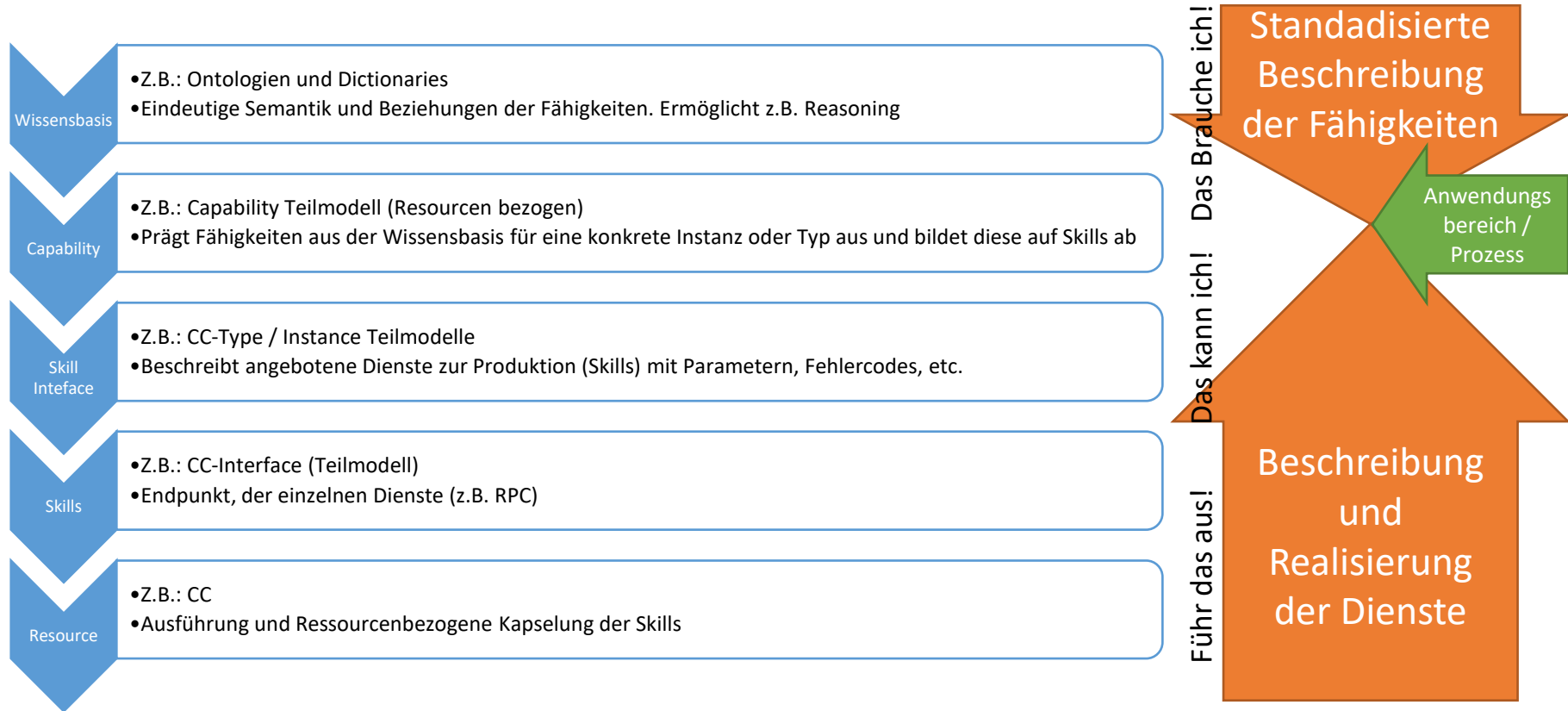
→ Mögliche Assets:

- Control Component (Type) selbst
- Oder auch Ressource (PPR): Stellglied, Modul, SPS, Device, ...





# Einordnung zu Fähigkeiten



# Use Case 1: Informationsmodellierung von CC

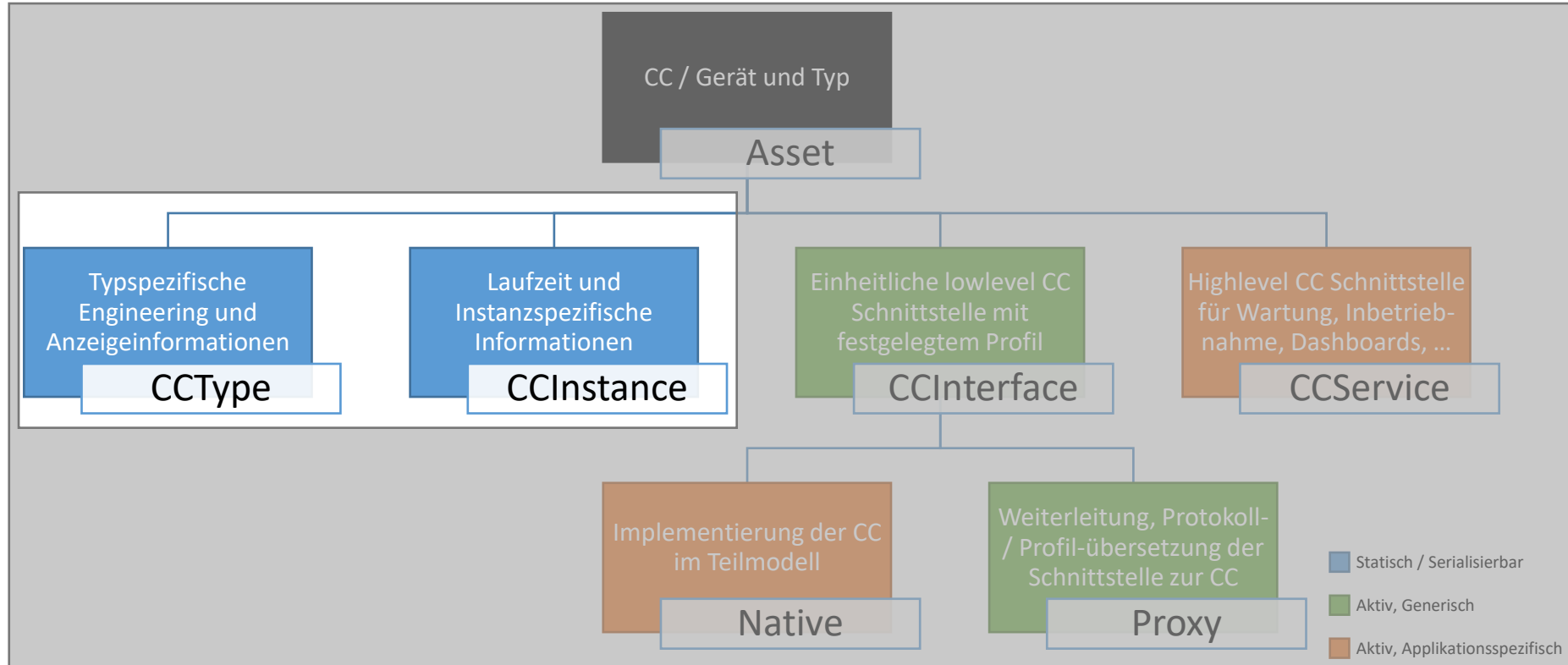
- Manche Informationen sind erst bei der Inbetriebnahme festgelegt oder ändern sich während des Lebenszyklus der CC (z.B. Endpunkt-Adresse)
- Vollständige Beschreibung einer CC enthält mehrere Aspekte:
  - Verwaltung der Information
    - Typ-Informationen
    - Instanz-Informationen
  - Interaktion mit der CC (Orchestrierung)
    - Schnittstelle

# Use Case 1: Informationsmodellierung von CC

- Mehrere CC-Teilmodelle zur Informationsmodellierung
- Statische Teilmodelle (Typ 1)
  - CCType
  - CCInstance
- Aktive Teilmodelle (Typ 2 und 3)
  - CCInterface
  - CCService



# Überblick CC Teilmodelle → Fokus



## Use Case 2: Engineering von Orchestrierung-Prozesse

- Automatischer Abruf von Information der CC bei der Entwicklung von Orchestrierung-Prozesse (z.B. Endpunkt, CC-Profil, verfügbare Fahrweise)
- Information zur verfügbaren/erlaubten Fahrweisen und Fehlerbeschreibungen können für die Programmierung von Ausnahmebehandlung-Prozesse verwendet werden
- Da die Verwaltung dieser Information standardisiert ist, können standardisierte Dashboards/HMI Schnittstellen dagegen programmiert werden

## Use Case 3: Implementierung von instanzspezifischen Fahrweisen

- In einige Anwendungen muss der CC-Owner zusätzliche instanzspezifischen Fahrweisen definieren können
- Grund-Fahrweisen sind vom CC-Hersteller definiert und als Typ-Information gespeichert und verwaltet
- Zusätzliche Fahrweise bzw. vom CC-Owner bearbeitete Fahrweise sollen als Instanz-information behandelt werden

# Requirements

- R1 (from UC1): Type-specific information is managed separately from instance-specific information in the AAS.
- R2 (from UC1): The administration and the orchestration of the CC is handled by different AAS submodels.
- R3 (from UC2): CC instance-specific information required for the engineering of orchestration processes (endpoint address, CC profile, etc.) is stored in a standardized way in an AAS submodel. This includes also error information for the development of exception handling process for the orchestrator.
- R4 (from UC2): Detailed information about errors is also available during runtime.
- R5 (from UC3): It is possible to define Instance-specific services. Component owner can extend the core services offered by a component type for a given application.

# Design Decisions

- DD1: Mapping of CC-instance-specific and CC-type-specific information in the AAS
- DD2: Administration of Operation Mode Error Related Information
- DD3: Administration of Operation Mode Information

## DD1: Mapping of CC-instance-specific and CC-type-specific information in the AAS

1. Use of one AAS submodel for the complete description of CC information. This submodel contains type-specific information if the AAS is of the kind "type" and instance-specific information if the AAS is of the kind "instance".
2. At least two AAS submodels are used for the description of CC information. One submodel is used for the description of type-specific information and other for the description of instance-specific information.

## DD2: Administration of Operation Mode Error Related Information

1. Detailed error code information is stored under each operation mode e.g. as Multi Language Properties (MLP). In each case, only the information about the errors that can occur in said operation mode is stored. This information would be either stored in the CCType AAS submodel or in the CCInstance AAS submodel, depending on whether the operation mode is type-specific or instance-specific.
2. All detailed error code information is stored in the CCType model as a collection of MLP. Each operation mode contains a reference to its related errors



## DD3: Administration of Operation Mode Information

1. The CCTYPE AAS submodel contain information about the original type-specific operation modes of the CC defined by the component vendor. The component owner of the CC copies this information in the CCInstance AAS submodel. If needed, the component owner can change information about the original type-specific operation modes information or define new instance-specific operation modes in the CCInstance AAS submodel.
2. The information about the original type-specific operation modes of the CC defined by the component vendor is stored only in the CCTYPE AAS submodel. If the component owner requires to change some of these operation modes, he can define them as new instance-specific operation modes. Only instance-specific operation modes are stored in the CCInstance AAS submodel.

# CC Type und Instance Teilmodelle

SM	"ControlComponentType" [IRI, <a href="http://basys4.de/submodels/controlcomponent/type/demonstrator/({componentId})">http://basys4.de/submodels/controlcomponent/type/demonstrator/({componentId})</a> ]
SMC	"ControlInterfaces" (1 elements) @({Multiplicity=One})
SMC	"ControlInterface" (3 elements) @({Multiplicity=OneToMany}) @({EditIdShort=False})
Prop	"Protocol" = opc.tcp @({Multiplicity=One})
Prop	"SupportedProfiles" = BASYSDEMO,BASYS,FULL @({Multiplicity=One})
Prop	"SupportedSecurityPolicies" @({Multiplicity=ZeroToOne})
SMC	"OperationModes" (1 elements) @({Multiplicity=One})
SMC	"OperationMode" (7 elements) @({Multiplicity=ZeroToMany}) @({EditIdShort=False})
Prop	"OperationName" @({Multiplicity=One})
SMC	"Parameters" (1 elements) @({Multiplicity=One})
SMC	"Parameter" (4 elements) @({Multiplicity=ZeroToMany}) @({EditIdShort=False})
Prop	"Direction" = In @({Multiplicity=One}) @({FormChoices=In;Out;InOut})
Prop	"ParameterName" @({Multiplicity=One})
Prop	"ParameterType" @({Multiplicity=One})
SMC	"ValidValues" (1 elements) @({Multiplicity=ZeroToOne})
Prop	"Disabled" = True @({Multiplicity=ZeroToOne})
Prop	"AllowedInExecutionModes" = AUTO,SEMI AUTO,MANUAL,SIMULATE @({Multiplicity=ZeroToOne})
MLP	"OperationDisplayName" -> pick and place @({Multiplicity=ZeroToOne})
Prop	"AllowedInOccupationStates" = FREE, OCCUPIED, PRIO @({Multiplicity=ZeroToOne})
SMC	"ErrorCodes" (1 elements) @({Multiplicity=One})
Ref	"ErrorCode" @({Multiplicity=ZeroToMany})
SMC	"ErrorCodes" (1 elements) @({Multiplicity=One})
MLP	"ErrorCode" -> Example description @({Multiplicity=ZeroToMany})

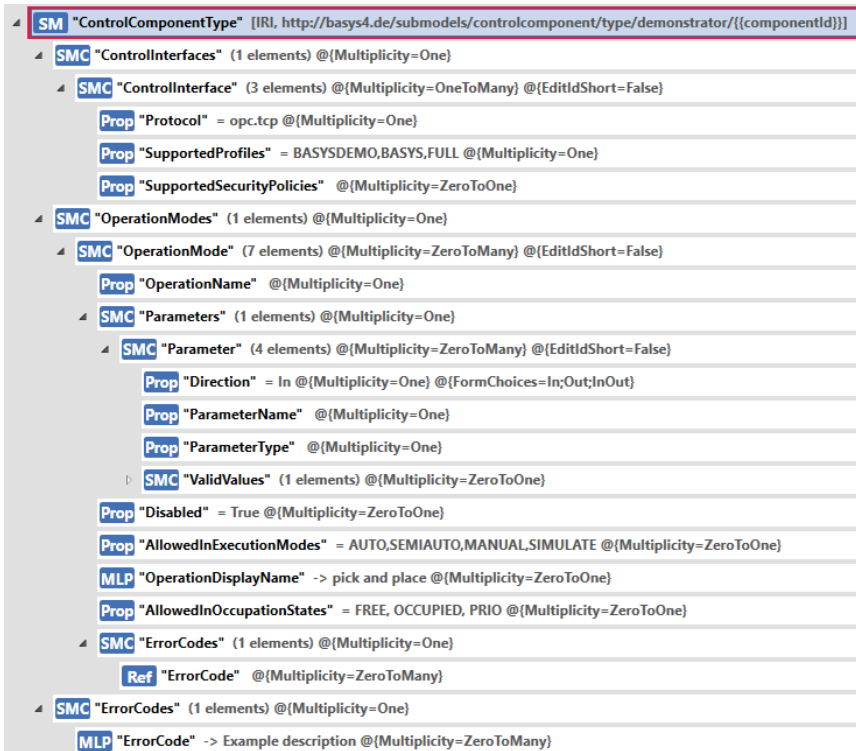
SM	"ControlComponentInstance" [IRI, <a href="http://basys4.de/submodels/controlcomponent/instance/demonstrator/({componentId})">http://basys4.de/submodels/controlcomponent/instance/demonstrator/({componentId})</a> ]
TBD	Submodel to describe a Control Component Instance ready
SMC	"ControlEndpoints" (1 elements) @({Multiplicity=One})
SMC	"ControlEndpoint" (3 elements) @({Multiplicity=OneToMany}) @({EditIdShort=False})
Prop	"Address" = opc.tcp://127.0.0.1:4840/0:Objects @({Multiplicity=One})
Prop	"Profile" = BASYSDEMO @({Multiplicity=One}) @({FormChoices=MINIMAL;BASYS;BASYSDEMO;PACKML;MTP;FULL})
Prop	"SecurityPolicy" @({Multiplicity=ZeroToOne})
SMC	"OperationModes" (1 elements) @({Multiplicity=One})
SMC	"OperationMode" (7 elements) @({Multiplicity=ZeroToMany}) @({EditIdShort=False})
Prop	"OperationName" @({Multiplicity=One})
SMC	"Parameters" (1 elements) @({Multiplicity=One})
SMC	"Parameter" (4 elements) @({Multiplicity=ZeroToMany}) @({EditIdShort=False})
Prop	"Direction" = In @({Multiplicity=One}) @({FormChoices=In;Out;InOut})
Prop	"ParameterName" @({Multiplicity=One})
Prop	"ParameterType" @({Multiplicity=One})
SMC	"ValidValues" (1 elements) @({Multiplicity=ZeroToOne})
Prop	"ValidValue" @({Multiplicity=OneToMany})
Prop	"Disabled" = True @({Multiplicity=ZeroToOne})
Prop	"AllowedInExecutionModes" = AUTO,SEMI AUTO,MANUAL,SIMULATE @({Multiplicity=ZeroToOne})
MLP	"OperationDisplayName" -> pick and place @({Multiplicity=ZeroToOne})
Prop	"AllowedInOccupationStates" = FREE, OCCUPIED, PRIO @({Multiplicity=ZeroToMany})
SMC	"ErrorCodes" (1 elements) @({Multiplicity=One})
Ref	"ErrorCode" @({Multiplicity=ZeroToMany})
Ref	"HasType" -> [Submodel, Local, IRI, <a href="http://basys4.de/submodels/controlcomponent/type/demonstrator/({componentId})">http://basys4.de/submodels/controlcomponent/type/demonstrator/({componentId})</a> ]

# CC Instance Teilmodell

- Wie lauten die Endpunkte für diese Instanz?
- Welche Instanz-spezifischen Fahrweisen existieren?
  - Gleicher Aufbau wie bei Typ-spezifischen Fahrweisen
  - Instanz-spezifisches Überschreiben von Typ-spezifischen Fahrweisen
- Referenz zum Typ-Teilmodell (kann später auch über AAS-Typ Beziehung abgebildet werden)

SM	"ControlComponentInstance" [IRI, http://basys4.de/submodels/controlcomponent/instance/demonstrator/((componentId))]
TBD	Submodel to describe a Control Component Instance ready
SMC	"ControlEndpoints" (1 elements) @({Multiplicity=One})
SMC	"ControlEndpoint" (3 elements) @({Multiplicity=OneToMany}) @({EditIdShort=False})
Prop	"Address" = opc.tcp://127.0.0.1:4840/0:Objects @({Multiplicity=One})
Prop	"Profile" = BASYSDEMO @({Multiplicity=One}) @({FormChoices=MINIMAL;BASYS;BASYSDEMO;PACKML;MTP;FULL})
Prop	"SecurityPolicy" @({Multiplicity=ZeroToOne})
SMC	"OperationModes" (1 elements) @({Multiplicity=One})
SMC	"OperationMode" (7 elements) @({Multiplicity=ZeroToMany}) @({EditIdShort=False})
Prop	"OperationName" @({Multiplicity=One})
SMC	"Parameters" (1 elements) @({Multiplicity=One})
SMC	"Parameter" (4 elements) @({Multiplicity=ZeroToMany}) @({EditIdShort=False})
Prop	"Direction" = In @({Multiplicity=One}) @({FormChoices=In;Out;InOut})
Prop	"ParameterName" @({Multiplicity=One})
Prop	"ParameterType" @({Multiplicity=One})
SMC	"ValidValues" (1 elements) @({Multiplicity=ZeroToOne})
Prop	"ValidValue" @({Multiplicity=OneToMany})
Prop	"Disabled" = True @({Multiplicity=ZeroToOne})
Prop	"AllowedInExecutionModes" = AUTO, SEMIAUTO, MANUAL, SIMULATE @({Multiplicity=ZeroToOne})
MLP	"OperationDisplayName" -> pick and place @({Multiplicity=ZeroToOne})
Prop	"AllowedInOccupationStates" = FREE, OCCUPIED, PRIO @({Multiplicity=ZeroToMany})
SMC	"ErrorCodes" (1 elements) @({Multiplicity=One})
Ref	"ErrorCode" @({Multiplicity=ZeroToMany})
Ref	"HasType" -> [Submodel, Local, IRI, http://basys4.de/submodels/controlcomponent/type/demonstrator/((componentId))]

# CC Type Teilmodell



- Welche Schnittstellen stellen die Instanzen des Typs bereit.
- Welche Fahrweisen mit welchen Parametern stellen die Instanzen des Typs bereit?
  - Wann sind die die Fahrweisen auswählbar?
  - Welche Fehler können in dieser Fahrweise auftreten?
- Welche Fehler können überhaupt auftreten?