

Arbeitspaket 6: Anlagenmodell

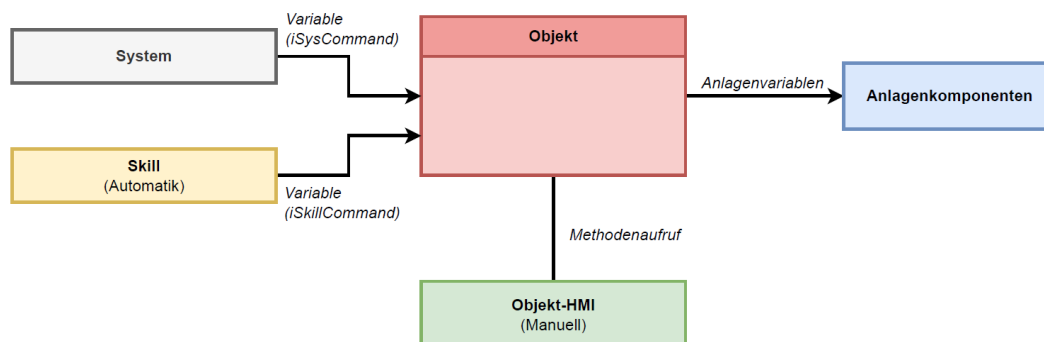
Einführungstext

Definierung der Objekt-Struktur

Wie bereits für die Skills wird auch für die Objekte eine Grundstruktur festgelegt, die als Basis für den Aufbau aller Objekte dient. Diese einheitliche Struktur erleichtert die Standardisierung der Interaktionen innerhalb der Software. Dabei sollen die Objektklassen möglichst objektorientiert gestaltet werden, sodass die Funktionalität der Objekte in klar abgegrenzten Methoden abgebildet wird. Zum Ausführen einer Funktionalität reicht es daher, die entsprechende Methode aufzurufen.

Die Methoden werden in zwei Kategorien unterteilt: nicht-funktionale und funktionale Methoden. Nicht-funktionale Methoden dienen dazu, das Objekt in einen betriebsbereiten Zustand zu versetzen (z. B. Verbinden, Trennen, Ein-, Ausschalten). Funktionale Methoden hingegen steuern den eigentlichen Betrieb des Objekts (z. B. Fahren, Stoppen).

Allerdings kann die objektorientierte Herangehensweise in diesem Projekt nicht durchgängig angewendet werden. Aufgrund der Art und Weise, wie der Skill definiert wurde, interagiert er nicht direkt über Methodenaufrufe mit dem Objekt, sondern über eine Variable. Auch das System interagiert über eine Variable mit dem Objekt.



Das Objekt besitzt grundsätzlich vier Schnittstellen. Wie bereits erwähnt werden die Schnittstellen des Systems und des Skills über Input-Variablen realisiert. Der Skill stellt dabei den Automatik-Modus dar. Jedes Objekt kann für den Manuell-Modus über eine Objekt-HMI bedient werden. Die Interaktion hierbei findet über Methodenaufrufe statt. Die Schnittstelle gegen aussen werden über komponentenspezifische Variablen umgesetzt.

Diese Definition dient als Grundlage für die Bestimmung der Schnittstellenvariablen, welche jedes Objekt besitzen muss. Dabei wird sich wieder an den Standard von PLCopen angelehnt (Verweis).

Input-Variablen:

Variable	Typ	Beschreibung
iMode	INT	Information über aktuellen Betriebsmodi
iSkillCommand	INT	Steuerungsvariabel von Skill
iSysCommand	INT	Steuerungsvariabel von System
iSysState	INT	Information über Stand von System
bReset	BOOL	Trigger für Reset von Objekt

Output-Variablen:

Variable	Typ	Beschreibung
bDone	BOOL	Information ob Objekt erfolgreich ausgeführt wurde
bBusy	BOOL	Information ob Objekt im Moment ausgeführt wird
bError	BOOL	Information ob sich Objekt im Moment im Fehlerzustand befindet
iErrorID	INT	Information um welchen Fehler (bezüglich Prozess) es sich handelt

Eigenschaften

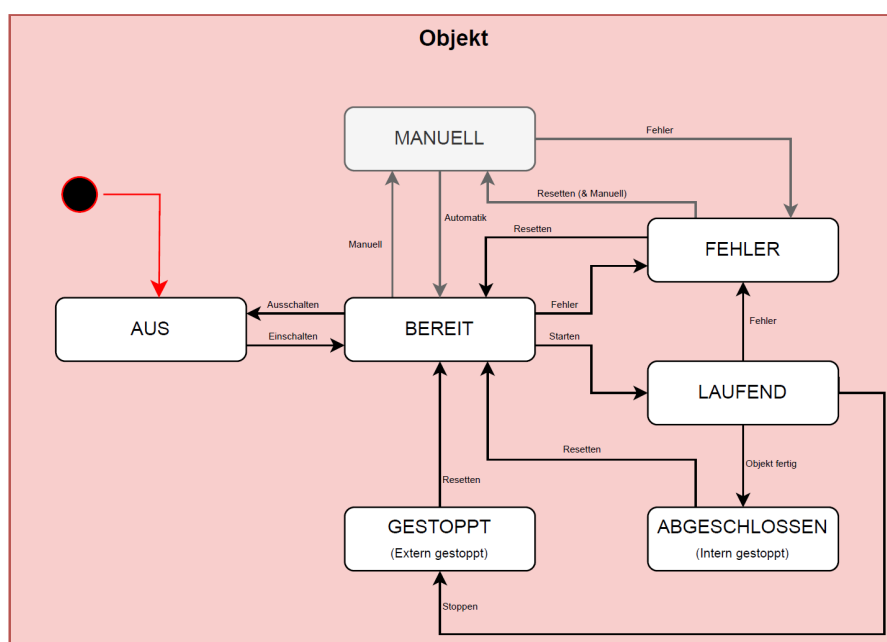
Variable	Typ	Beschreibung
P_State (GET)	INT	Information über Zustand von Objekt

Die Input-Variable «iMode» wird verwendet, um den Betriebsmodi (Manuell und Automatik) des Objektes zu definieren. Die zwei Variablen «iSkillCommand» und «iSysCommand» können für das Bedienen des Objektes verwendet werden. Der Skill gibt dem Objekt dabei z.B. den Befehl einen definierte Funktion auszuführen.

Das unter Kapitel BlaBlabla erstellte Zustandsdiagramm für das Objekt hat sich auf die Interaktion mit einem Skill fokussiert. Betrachtet man nur das Objekt, mit Bezug auf die definierten SchnittstellenvARIABLEN, muss das Zustandsdiagramm erweitert werden. Das Objekt besitzt neu 7 Zustände:

Ein Objekt benötigt mindestens folgende 7 Zustände:

Zustand:		Beschreibung:
0	AUS	Das Objekt ist ausgeschaltet (Startzustand)
1	BEREIT	Das Objekt ist eingeschaltet und bereit
2	MANUELL	Das Objekt befindet sich im Manuell-Modus
3	LAUFEND	Das Objekt ist aktiv
4	ABGESCHLOSSEN	Das Objekt hat den Prozess durchgeführt und hat selbständig gestoppt
5	GESTOPPT	Das Objekt wurde gestoppt durch externe Einwirkung gestoppt
6	FEHLER	Es gibt einen Fehler bezüglich des Objektes



Definieren der Anlagenvariablen