

Problemstellung

Standards zur Realisierung leittechnischer Applikationen, wie beispielsweise der IEC-Standard 1131-3 „Speicherprogrammierbare Steuerungen“, stoßen beim Anwender auf große Akzeptanz, denn sie ermöglichen die plattform-unabhängige Realisierung leittechnischer Applikationen in verteilten, heterogenen Automatisierungssystemen.

Wichtigstes Sprachelement der IEC 1131 wie auch der geplanten IEC 1499 ist der Funktionsbaustein, der in Analogie zur Einzelgerätetechnik (um 1965) eine autarke Verarbeitungseinheit darstellt, die mit der Außenwelt über Eingangs- und Ausgangsports kommuniziert sowie interne Variablen besitzen kann.

Obwohl sich Funktionsbausteine durchaus als Objekte im Sinne der Objektorientierung auffassen lassen können (Kriterien: Identität, Zustand und Verhalten), machen die Standards keine Aussage darüber, ob Funktionsbausteine sich auch zur Laufzeit wie Objekte hantieren lassen (zum Beispiel Erzeugung von neuen Funktionsbausteinen). Darüber hinaus vermißt man grundlegende Konzepte der Objektorientierung wie Kapselung, Vererbung, polymorphe Operationen oder typisierte Beziehungen, wie sie zum Beispiel mit Hilfe der graphischen Unified Modeling Language (UML) verwendet werden können.

Moderne „offene“ Automatisierungssysteme, die in der Regel auf Standardbetriebssystemen beruhen, ermöglichen als Alternative die freie Programmierung. In heterogenen Systemen bedeutet dies jedoch im allgemeinen entweder auf die Systemdienste der vorhandenen Komponenten verzichten oder aufwendige Anbindungen an diese „offenen“ Komponenten entwickeln zu müssen.

Objektverwaltung ACPLT/OV

Das Objektverwaltungssystem ACPLT/OV spezifiziert eine an die Bedürfnisse der Prozeßleittechnik angepaßte objektorientierte Rahmenarchitektur mit den folgenden Eigenschaften:

- Werkzeugkasten zum Aufbau von Objekten aus einfachen, standardisierten Elementen,
- Laden und Instantieren von Klassen und Assoziationen (Beziehungen) zur Laufzeit,
- strukturierte Sprache zur Modellierung von Klassen und Assoziationen,
- Online-Verfügbarkeit nicht nur von Instanz-, sondern auch von Klasseneigenschaften,
- volle Integration in den leittechnischen Informationsverbund über ACPLT/KS,
- offene Spezifikation und frei verfügbare Implementation als „Open Source“.

Objekte werden persistent in einer Online-Datenbasis abgelegt. Sie werden „aktiv“ sobald ihre Methoden ausgeführt werden. Sowohl zyklische Objektbearbeitung (zum Beispiel über den OV-Scheduler) als auch ereignisgesteuerte Objektbearbeitung (beispielsweise beim Zugriff auf Variablen) sind möglich.

Externe Referenzen auf Objekte erfolgen grundsätzlich über hierarchische Klartextnamen (Pfadnamen) ähnlich wie im World Wide Web. Die Hierarchietiefe ist dabei unbegrenzt. So bezeichnet

/Produkte/Ofen2/Bramme34.Temperatur

beispielsweise die Temperatur des Produktes „Bramme34“, das sich zur Zeit in der Teilanlage „Ofen2“ befindet.

Innerhalb des Objektsystems kann der Zugriff auf Objekte effizient über lokale Objektreferenzen (Zeiger) erfolgen.

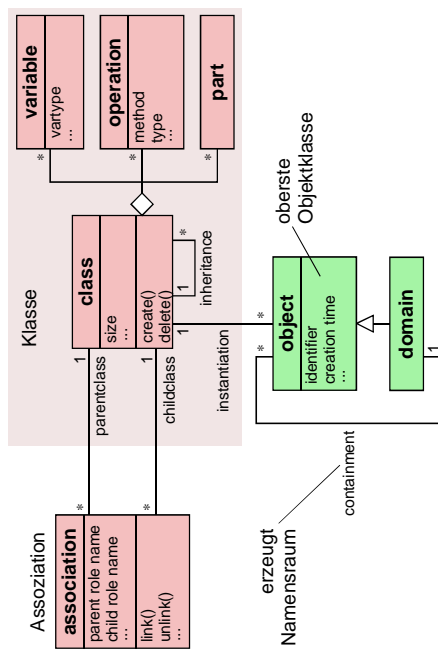
Objektorientiertes Modellieren

Ein Objekt („Exemplar“ oder „Instanz“) innerhalb eines ACPLT/OV-Objektsystems besteht aus den folgenden im ACPLT/OV-Objektmodell definierten Elementen:

- **Variablen** (variables),
- eingebetteten **Komponentenobjekte** (parts),
- durch **Assoziationen** definierte Verknüpfungen (links) und
- **Operationen** (operations), die durch ANSI C-Funktionen implementiert werden können.

Als Modellierungssprache verwendet ACPLT/OV eine eigene strukturierte Sprache, mit der sich die Objektklassen (Regler, Gerät, ...) definieren lassen. Die Mächtigkeit dieser Sprache geht weit über die Möglichkeiten der IEC 1131 hinaus:

- **Verfeinerungshierarchien:** Vererbung,
- **komplexbildende Hierarchien:** Basiselemente und Einbettung von Objekten,
- **Polymorphie:** Überladen von Methoden,
- **Assoziationen:** Spezifikation von Querverweisen, mit denen sich Objekte zur Laufzeit miteinander verknüpfen lassen.



Metamodell von ACPLT/OV

