#### 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 plattformunabhängige 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 Eingangsund 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) verdendet 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
  - 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 OVScheduler) 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.

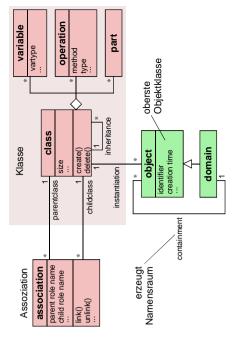
# **Objektorientiertes Modellieren**

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

- Variablen (variables),
- eingebetteten Komponentenobjekte (parts),
- durch Assoziationen definierte Verknüpfungsenden (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

Das so definierte Objektmodell wird mit dem ACPLT/OV C-Codegenerator in ANSI-C-Code übersetzt. Die Implementation von Operationen der Klassen erfolgt zur Zeit ebenfalls in ANSI C, wodurch Applikationen hochportabel sind.

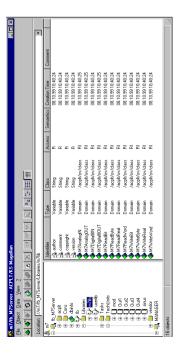
Das Wissen über das Modell verschwindet durch die Compilation jedoch nicht wie üblich in den Tiefen des Programmcodes. Vielmehr werden Metaobjekte (Instanzen des Metamodells) erzeugt, die das Objektmodell beschreiben.

Das Objektsystem ist somit selbst-deskriptiv, das heißt "die Wahrheit liegt im Zielsystem":

- Reflexion: Methoden können zur Laufzeit auf Metainformationen zugreifen (zum Beispiel die Schnittstelle eines Objekts),
- Introspektion: Kommunikationssysteme wie ACPLT/KS mit Zugriff auf Objekte über Klartext-Pfadnamen können projektierungsfrei eingebunden werden,
- Rückdokumentieren projektierter Instanzen und Verknüpfungen ist generisch möglich.

# Standardisierte Systemdienste

Um in heterogenen Umgebungen netzwerkweit projektierungsfreien Online-Zugriff auf Objekte der ACPLT/OV-Datenbasis zu erhalten, ist das Kommunikationssystem ACPLT/KS bereits in ACPLT/OV integriert (ACPLT/KS-Server).



Zugriff auf ein OV-System via KS Magellan

Neben dem Zugriff auf Daten können Klientenanwendungen via ACPLT/KS direkt auf die Dienste der Objektverwaltung zugreifen:

- Erkundung der im Objektsystem verfügbaren Objekte und Metaobjekte (Klassen),
- Erzeugen und Löschen von Instanzen,
- Erzeugen und Löschen von Querverweisen (Links).

Selbstverständlich kann ACPLT/OV auch als ACPLT/KS-Klient auftreten.

#### **Plattformübergreifend**

Die Objektverwaltung ACPLT/OV ist zur Zeit für folgende Plattformen verfügbar:

- Windows NT/9x,
- Linux 2.x und weitere Unix-Derivate,
- OpenVMS,
- 16bit-Microcontroller aus der C16x-Familie.

Die Basisbibliotheken sind laufzeiteffizient, schlank und hochportabel in ANSI C realisiert und benötigen lediglich 256k Programmcode (Windows) inkl. ACPLT/KS-Anbindung. Hinzu kommt lediglich noch die jeweilige Transportschicht, zum Beispiel TCP/IP.

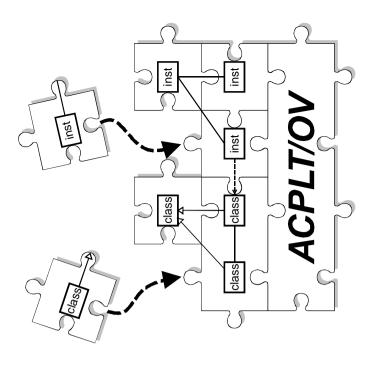
#### **Produkte**

ACPLT/OV ist kein akademischer Selbstzweck. Es existieren bereits zwei kommerzielle Produkte, die auf der Objektverwaltung basieren:

- das Funktionsbausteinsystem iFBS Pro der LTSoft GmbH und
- das Commutec S Funktionsmodul FM von Endress+Hauser.

Dokument 0W7X03S6-001 • Stand 10.99.1 • Editor: Dirk Meyer © 1999 Lehrstuhl für Prozeßleittechnik, RWTH Aachen ACPLT ist ein eingetragenes Markenzeichen der Rheinisch-Westfälischen Technischen Hochschule Aachen





#### Objektverwaltung ACPLT/OV

Lehrstuhl für Prozeßleittechnik (PLT)
RWTH Aachen
D-52056 Aachen
mailto:ov@plt.rwth-aachen.de/ov