The Problem

Standards dealing with the realization of process control and process control engineering applications such as IEC 1131 "Programmable Controllers" are generally accepted because they enable the platform-independent realization of process control and process control engineering applications in a distributed, heterogeneous process control system.

The most important element of the IEC 1131 as well as of the proposed IEC 1499 languages is the function block which represents an autarkic processing unit. A function block communicates with the outside world using input and output ports (analogous to the hardware control devices used in 1965) and may contain internal variables.

Although function blocks can be regarded as objects in the sense of the object oriented paradigm (criteria: identity and combination of state and behavior), these standards do not specify if or even in which way it is possible to handle function blocks as objects at runtime (for example, instantiation of new function blocks). Furthermore, many basic object-oriented concepts such as encapsulation, inheritance, polymorphic operations or typed relationships are missing, which are available in languages such as the graphical Unified Modeling Language (UML).

Modern “open” process control systems, which are usually based on standard operating systems, allow to use general programming languages (ANSI C or C++ for example) as an alternative. However, in heterogeneous systems this means either to forego the system services available or that expensive software for the integration of the various “open components” needs to be developed.

Object Management ACPLT/OV

The object management system ACPLT/OV specifies an object-oriented framework architecture adapted to the needs of process control engineering with the following features:

- toolkit for constructing objects using simple, standardized elements,
- loading and instantiation of classes and associations (relationships) at runtime,
- structured language for modeling of classes and associations,
- online access not only of instance but also of class properties,
- full integration into the process control information environment over ACPLT/KS,
- open specification and freely available implementation (“Open Source”).

Objects are persistently stored in an online database. They become “active” as soon as their methods are invoked. Cyclic processing of objects (for instance using the ACPLT/OV scheduler) as well as event triggered processing of objects (when accessing a variable for example) is possible.

External references to objects are always provided using hierarchical path names in clear text similar as in the World Wide Web. The depth of this hierarchy is not limited. As an example, the path name

/Products/Oven2/Slab34.Temperature

references the temperature of the product “Slab34” which is currently located in the plant section “Oven3”.

Inside the object system objects can be efficiently accessed using local object references (pointers).

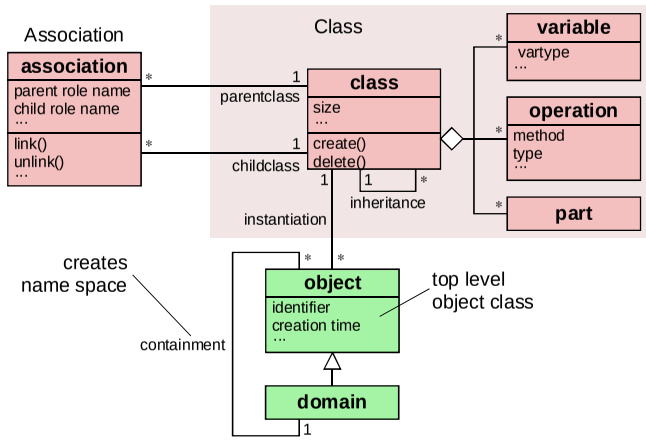
Object-oriented Modeling

An object (an individual “instance”) inside of an ACPLT/OV object system is constructed using the following elements defined in the ACPLT/OV object model:

- **variables**,
- embedded **component objects** (parts),
- relationship ends defined by **associations** (links) and
- **operations** which can be implemented by ANSI C functions.

As a modeling language ACPLT/OV uses its own structured language which can be used for defining object classes (controller, device, ...). The power of this language exceeds the possibilities of IEC 1131 by far:

- **refinement hierarchies:** inheritance,
- **aggregation hierarchies:** basic elements and embedding of objects,
- **polymorphism:** overloading of methods,
- **associations:** specification of relationships which can be used at runtime in order to link objects to other objects.



Meta Model of ACPLT/OV

The object model defined in this way is translated by the ACPLT/OV C code generator into ANSI C code. The implementation of operations of the classes is currently carried out in ANSI C as well, so that the applications are extremely portable.

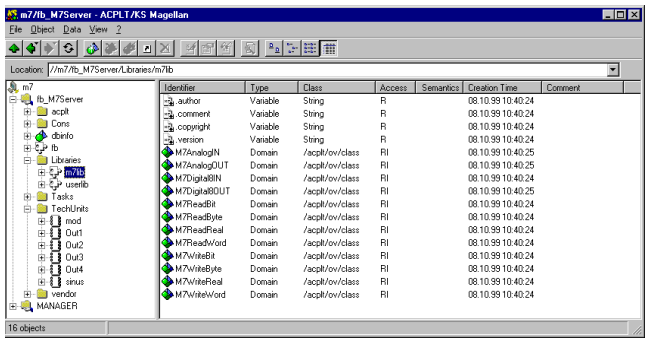
However, the knowledge about the model does not disappear in the depths of the program code as usual. Rather meta objects (instances of the meta model) are created in the database, which describe the object model.

Thus the object system is self-descriptive, i.e. “The Target Tells The Truth”:

- **reflection:** methods have access to meta information (for example the interface of on object) at runtime,
- **introspection:** communication systems such as ACPLT/KS, which use path names for accessing objects, can be integrated without having to configure anything,
- **documenting** of projected instances and links is generically possible.

Standardized System Services

In order to provide online access to objects in the ACPLT/OV database free of configuration needs and in heterogeneous environments, the communication system ACPLT/KS is already integrated into ACPLT/OV (ACPLT/KS server).



Access to an OV System via KS Magellan

Apart from accessing data client applications can directly access the object management services via ACPLT/KS:

- exploration of objects and meta objects (classes) available in the object system,
- creation and deletion of instances,
- creation and deletion of relationships between objects (links).

Of course ACPLT/OV can act as an ACPLT/KS client as well.

Platform-Independence

The object management system ACPLT/OV is currently available for the following platforms:

- Windows NT/9x,
- Linux 2.x and further Unix derivatives,
- OpenVMS,
- 16 bit microcontroller of the C16x family.

The basic libraries of ACPLT/OV are very efficient with respect to execution speed and code size, their code size is only 256k (Windows) including the ACPLT/KS integration. Only code for the transport layer, TCP/IP for instance, is additionally necessary.

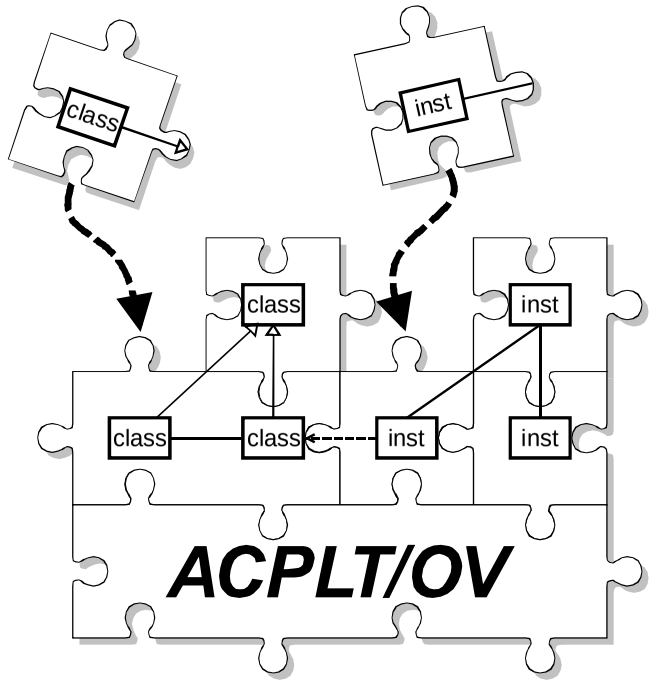
Products

ACPLT/OV is no academic end in itself. Two commercial products based on the object management system already exist:

- the function block system iFBS Pro of LTSoft GmbH and
- the Commutec S Function Module FM of Endress+Hauser.

Document 0W7X05D2-001 • Revision 10.99.1 • Editor: Dirk Meyer

ACPLT
AACHENER
PROZESSLEITTECHNIK



Object Management ACPLT/OV

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