Communication with External Systems

The APROS environment is used for a wide variety of purposes e.g. to support process and automation design, testing, operator training, operation, re-design and maintenance. For every phase in the system lifecycle, there are a number of other applications that are used to perform calculations, visualise results, connect to plant measurements etc. Easily configurable, efficient and reliable connections between APROS and these applications are vitally important. The APROS environment provides several alternatives for connecting external software to the simulation engine.

OPC and Custom DCOM interfaces

Windows clients can access APROS model data through an OPC (OLE for Process Control) interface, which is based on the DCOM (Distributed Component Object Model) technology. OPC is an industrial de-facto communication standard that specifies a number of DCOM interfaces for the exchange of e.g. data and events. OPC specifications are widely supported by automation system and visualisation tool providers.

Information about OPC, e.g. all released specifications can be obtained from the web pages of the OPC Foundation.

http://www.opcfoundation.org

Currently APROS supports the custom interface part of the OPC Data Access specification version 2.04.

APROS provides also custom DCOM interfaces, which provide functionality not directly defined in the OPC standard. These functions include e.g. start/stop of simulation, malfunction handling, recording of the simulation run and adjusting of the desired simulation speed.

A list of OPC compliant software products is maintained on the OPC Foundation web pages. They include e.g.

- automation system components
- virtual control systems
- visualisation packages
- control room emulation packages
- links to office applications

OPC communication is a good choice in case OPC compliant tools are available. The benefit of OPC is that by using compliant tools, one can link APROS data to be used in other software without any programming.

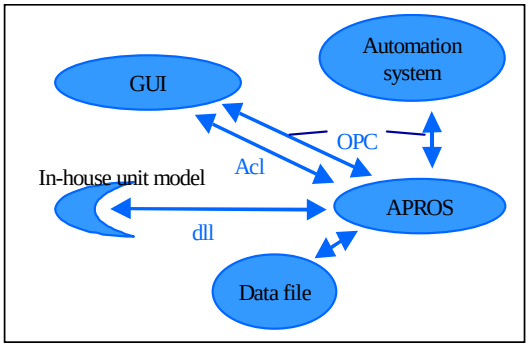

Figure 1 APROS communications

ACL – APROS Communication Library

The APROS Communication Library ACL is a low-level library that enables any client software to give APROS commands and exchange data. The implementation of the library is built on top of the TCP/IP protocol, and it is available on every hardware platform on which APROS is available. The headers of the library are available in C and FORTRAN.

There are two types of channels in ACL. The command channel accepts formatted ASCII data containing commands written in APROS command language. The data channels of different types transfer data to and from APROS at very high rates.

In case the simulation is distributed in several APROS simulation engine instances, ACL can link data and synchronise operations between the simulation engines.

ACL is a good choice in case the custom-made code needs to be implemented as a separate executable program, and for some reason, the OPC interface cannot be used. Possible reasons are high performance requirements, a need of portability to other platforms than Windows, or a need to give APROS commands.

External models

Custom-made computation models can be included in the simulation cycle directly linked in the APROS simulation engine. The feature may be used e.g. for inserting unit models, reactions, automation components or correlations into the APROS simulation. The implementation of the model remains completely hidden from the end-user since no source code needs to be delivered for using the models.

The development of an external model is simple:

1. Implement the code as a stdcall function
2. Build the function into a dynamically linkable library
3. In the APROS model, enter the library and function names, calling frequency and input and output variables. Also the call order of different functions can be defined.

When the simulation is started, the library will be loaded and the routine will be called at the end of each time step. Similar functions can be implemented for external model initialisation at simulation start, and for clean up at the end of the simulation run. The libraries of external models are released between simulation runs, which makes it easy to develop the models.

Any programming tool can be used that supports implementing stdcall functions and building dynamically linkable (shared) libraries. The mechanism has been proven on every supported APROS platform using standard C and FORTRAN compilers.

External model is a good choice when extreme communication performance or tight execution synchronisation between the APROS model and the custom code is required. Another benefit is that the resulting software architecture is relatively simple, as no inter-process communication is needed. This is a major advantage especially if there are a large number of user-made models to be included in simulation.

Formatted data files

In addition to graphical tools, three types of text files can be used in the APROS environment: database definition files, command queue files and time series data files.

The APROS simulation model is defined by entering the model configurations, parameters of each component, and the initial state, into the simulation engine. This data forms the simulation database. During simulation, the state is updated by the solvers integrated in the simulation engine, and the external software connected to the simulation engine.

The entire contents of the simulation database can be written into a text-format definition file for re-use in another APROS version, on another hardware platform or another operating system. Similarly, database definitions can be produced by other software, e.g. a CAD system, and read in the simulation engine.

By the aid of command queue files, one can define events that occur during simulation on given conditions. The events may be e.g. malfunctions, changes in model parameters or read or write operations.

The APROS simulation engine can read and write time series data files. Output time series files can be used in any suitable data series post processing software (e.g. Excel) for visualisation or further analysis. Input time series can be configured to form time-dependent boundary conditions in the simulation model.