Communication with External Systems

Apros® simulation software 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. Apros provides several alternatives for connecting external software to the simulation engine e.g. OPC, Apros Communication library (ACL), External model, SCL User component, and Formatted data and sequence files.

OPC INTERFACES

Windows clients can access Apros model data through OPC interfaces. OPC is an industrial de-facto communication standard that specifies interfaces for the exchange of e.g. data and events. Specifications are widely supported by automation system and visualisation tool providers.

Information about OPC, e.g. list of all released specifications can be obtained from the web pages of the OPC Foundation - http://www.opcfoundation.org

Currently Apros provides OPC UA, DA and XML DA interfaces. Apros utilizes these interfaces and provides a way to browse its address space as well as reading and writing data items to it.

Apros provides also custom interfaces that utilise the same communication protocols as OPC, but provide functionality that is specific for simulators. These interfaces include e.g. routines for starting and stopping the simulation, malfunction handling, recording of the simulation run and adjusting the simulation speed.

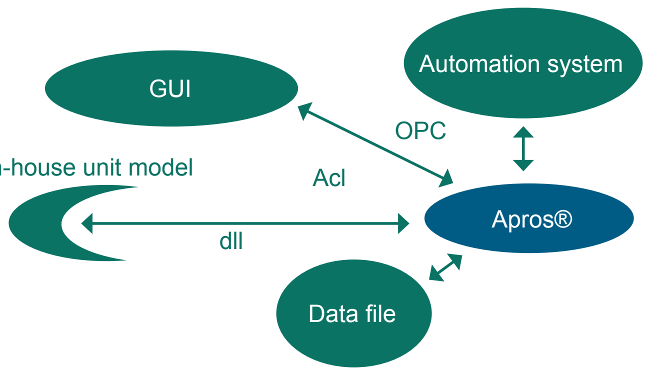
OPC communication is a good choice in case OPC compliant tools are available. The benefit of OPC is that by using compliant tools, one can link Apros data to be used in other software without any programming.

ACL – APROS COMMUNICATION LIBRARY

The Apros Communication Library (ACL) is a low-level library that enables 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 the same hardware and operating system platforms which Apros is available. The headers of the library are available in C, 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.

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.



▲ Figure 1. Apros Communications

Apros®

Communication with External Systems

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 or correlations into the Apros simulation or communicate with other tools. The implementation of the model can be completely hidden from the end-user since no source code needs to be delivered for using the model.

The development of an external model is simple, and Apros® includes example code that user can directly utilise as a part of their own code.

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 that make it easy to develop the models.

Any programming tool can be used that supports implementing stdcall functions and building dynamically linkable (shared) libraries. Apros® itself provides example implementations using standard C and FORTRAN languages.

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 area large number of user-made models to be included in simulation.

SCL USER COMPONENTS

SCL User component is feature where one can define own user component that can utilise generic Apros® components, define own attributes and contain additional functionality that is coded with SCL. SCL is Apros® specific functional scripting language. It provides typical programming language constructs as well as Apros® specific routines for configuring the model and running simulator experiments.

FORMATTED DATA AND SEQUENCE FILES

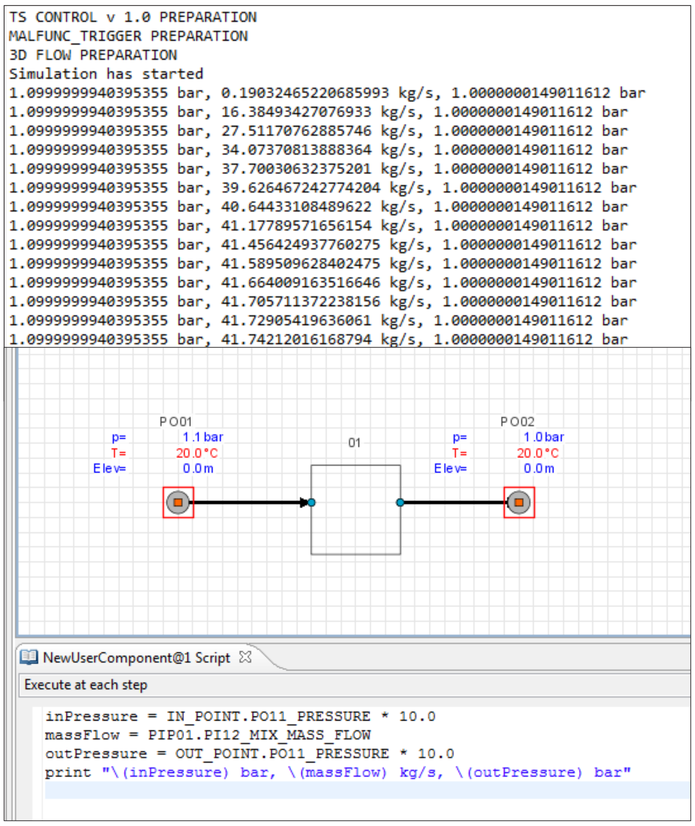
Apros® can also communicate with other programs through formatted data files. These data files could contain e.g. model configuration data or time series data. Files can be generated by Apros® and also read by Apros®.

The simulation model configuration can be written with SCL language. These database definitions can be produced by other software, e.g. a CAD system, and read in to the simulation engine.

Simulation sequences can be written with SCL language or as command queue files. These can define all the events occurring during simulation experiment. The events may be e.g. malfunctions, changes in model parameters or read or write operations. Also more complex functionality could be implemented using SCL e.g. criteria for ending the simulation or optimization cycles

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.

Figure 2. SCL User component example.
User component includes 1 pipe inside of it



```
TS CONTROL v 1.0 PREPARATION
MALFUNC_TRIGGER PREPARATION
3D FLOW PREPARATION
Simulation has started
1.0999999940395355 bar, 0.19032465220685993 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 16.38493427076933 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 27.51170762885746 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 34.07370813888364 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 37.70030632375201 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 39.626467242774204 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 40.64433108489622 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 41.17789571656154 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 41.456424937760275 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 41.589509628402475 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 41.664009163516646 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 41.705711372238156 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 41.72905419636061 kg/s, 1.0000000149011612 bar
1.0999999940395355 bar, 41.74212016168794 kg/s, 1.0000000149011612 bar
```

```
inPressure = IN_POINT.PO11_PRESSURE * 10.0
massFlow = PIP01.PI12_MIX_MASS_FLOW
outPressure = OUT_POINT.PO11_PRESSURE * 10.0
print "\(inPressure) bar, \(massFlow) kg/s, \(outPressure) bar"
```

VTT fortum