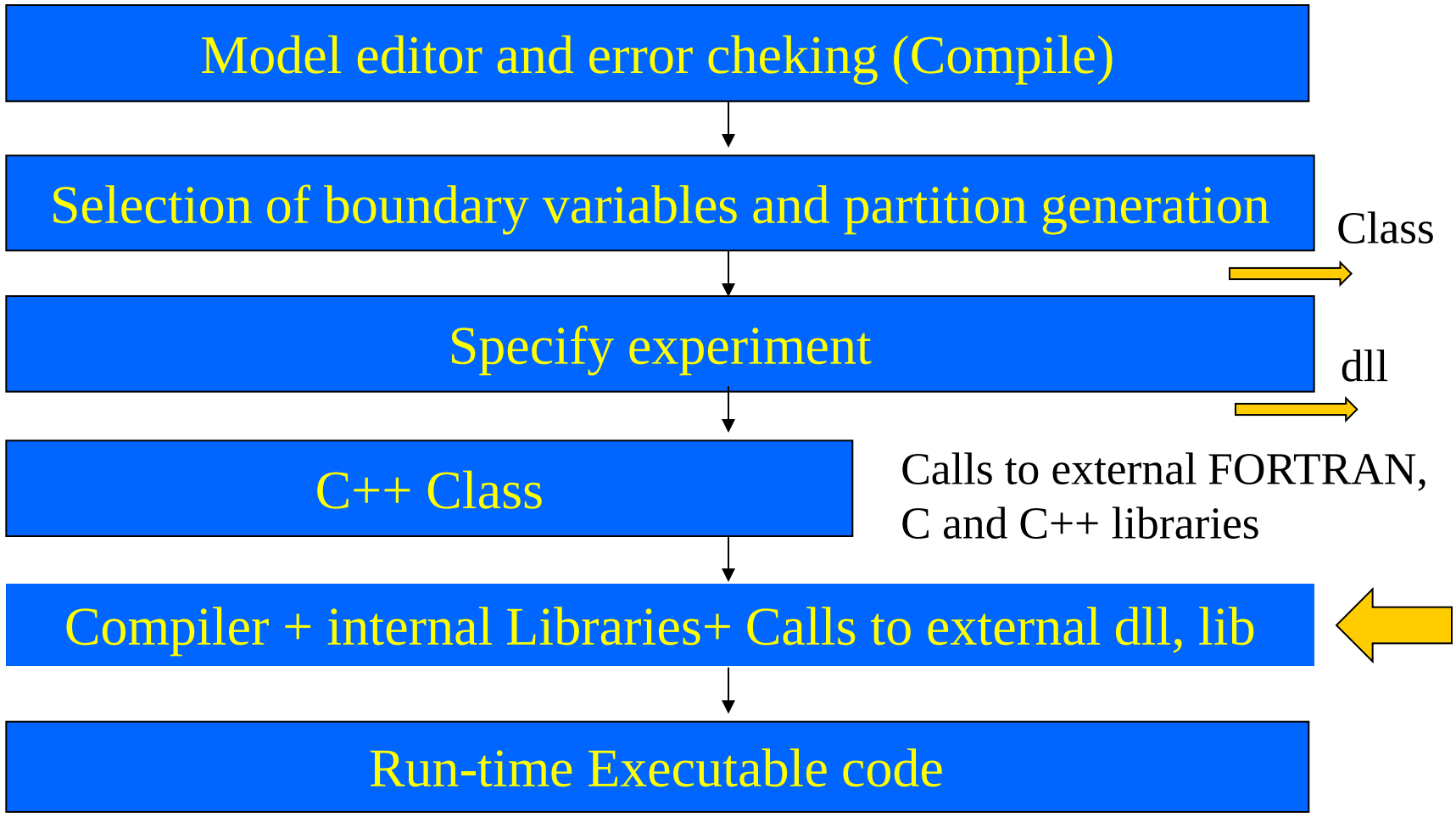
EcosimPro Communications

ISA-UVA

Generation of executable code



C++ Classes

EcosimPro automatically generates two C++ classes in the **autocode** directory of the corresponding library:

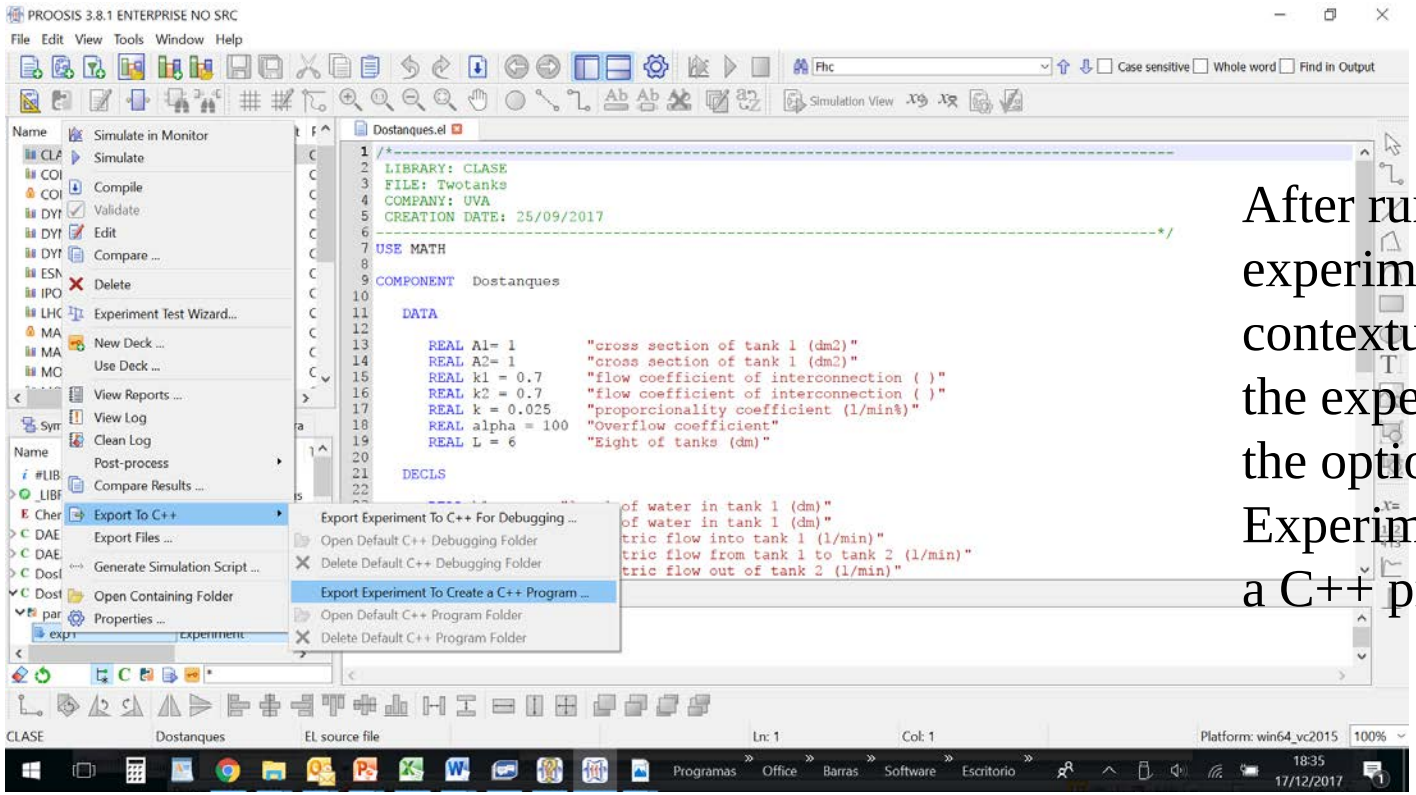
A C++ class that contains the partition
A C++ class that contains the experiment

The second class inherits the first one, adding some extra behaviour such as the initialisation of variables and the experiment code.

Both classes are generated with the standard names:
LIBRARY.COMPONENT.PARTITION(.EXPERIMENT).h
LIBRARY.COMPONENT.PARTITION(.EXPERIMENT).cpp

(+ replaces .) Header files for external use in the \include directory

Export an experiment to a C++ development environment to create a C++ program



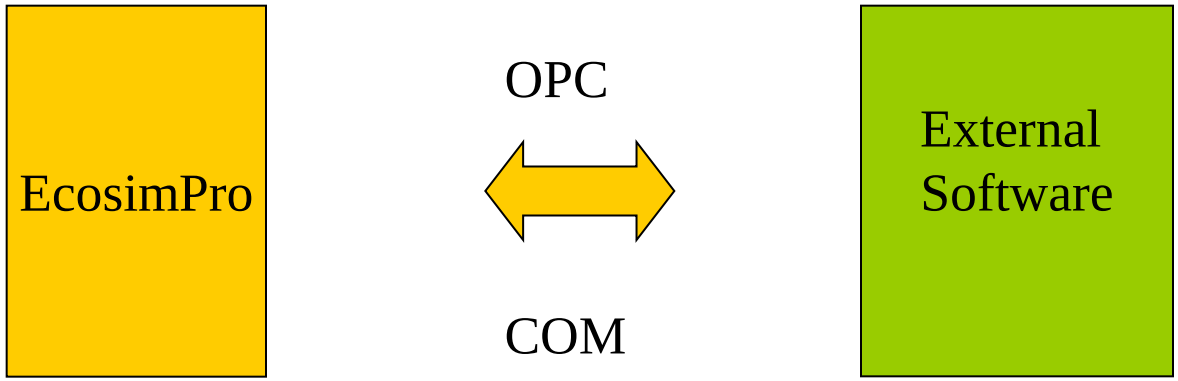
After running the experiment, in the contextual menu of the experiment select the option Export Experiment to create a C++ program.

External communications

- ✓ EcosimPro provides facilities for communication with external environments:
- ✓ Excel
- ✓ Matlab – Simulink
- ✓ Visual Basic, C++
- ✓ OPC
- ✓ ….

A dynamic link library file is generated when an experiment is compiled. It is identified by the name of the experiment plus a .dll extension.

Examples in the Interface Examples subdirectory



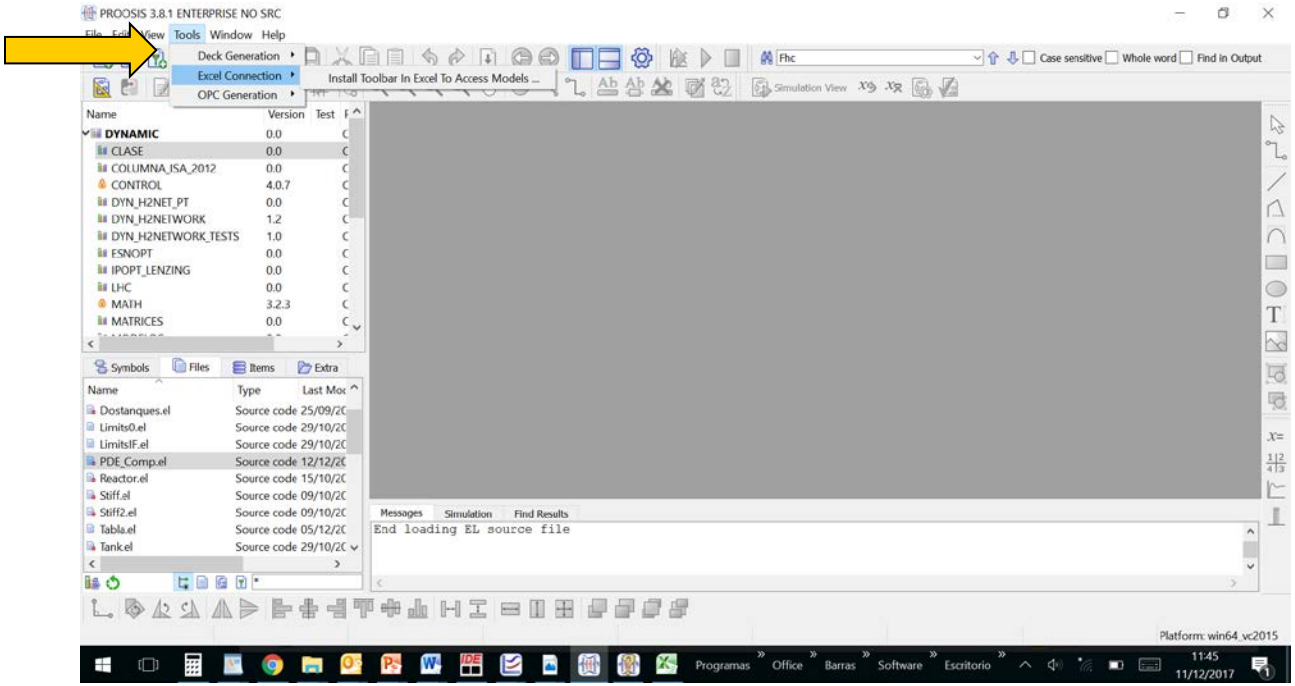
COM

El kernel de EcosimPro is an ActiveX application.
It uses the Microsoft COM protocol to access other applications.

Experiments or Decks generated by PROOSIS can be used easily in Microsoft Excel macros, Visual Basic, Visual Basic .NET, C#, C++ or any COM compatible language. For this purpose, EAI Excel Add-in includes a COM class named EAI_SMView which provides functions, properties and events to handle an experiment DLL file.

Using this COM interface does not require Microsoft Excel to be installed

Excel

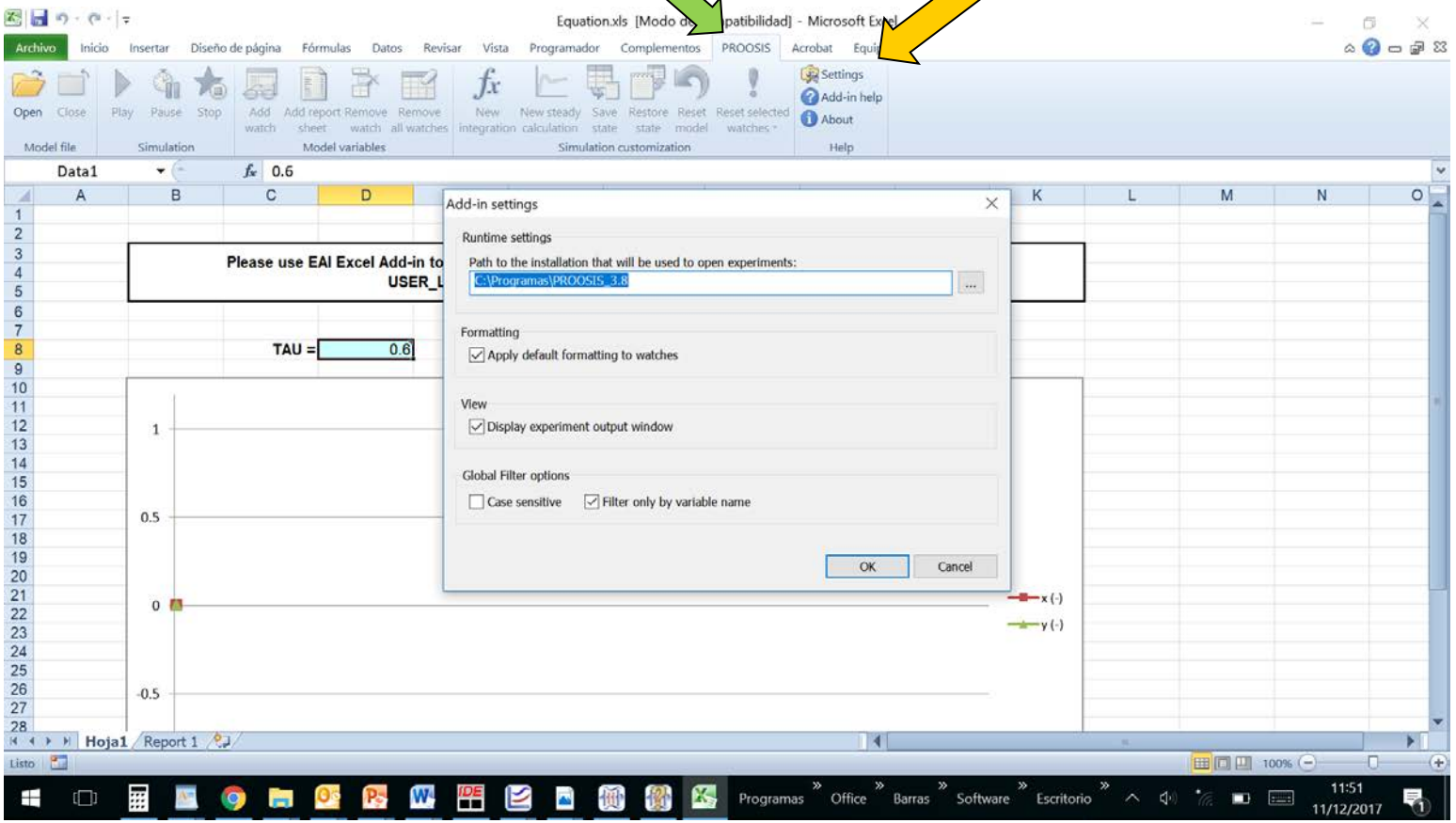


The Excel Add-in can be installed during the EcosimPro installation or later on from the "Tools", "Excel Connection", "Install Toolbar in Excel to Access Models" menu.

Configuring the Excel toolbar

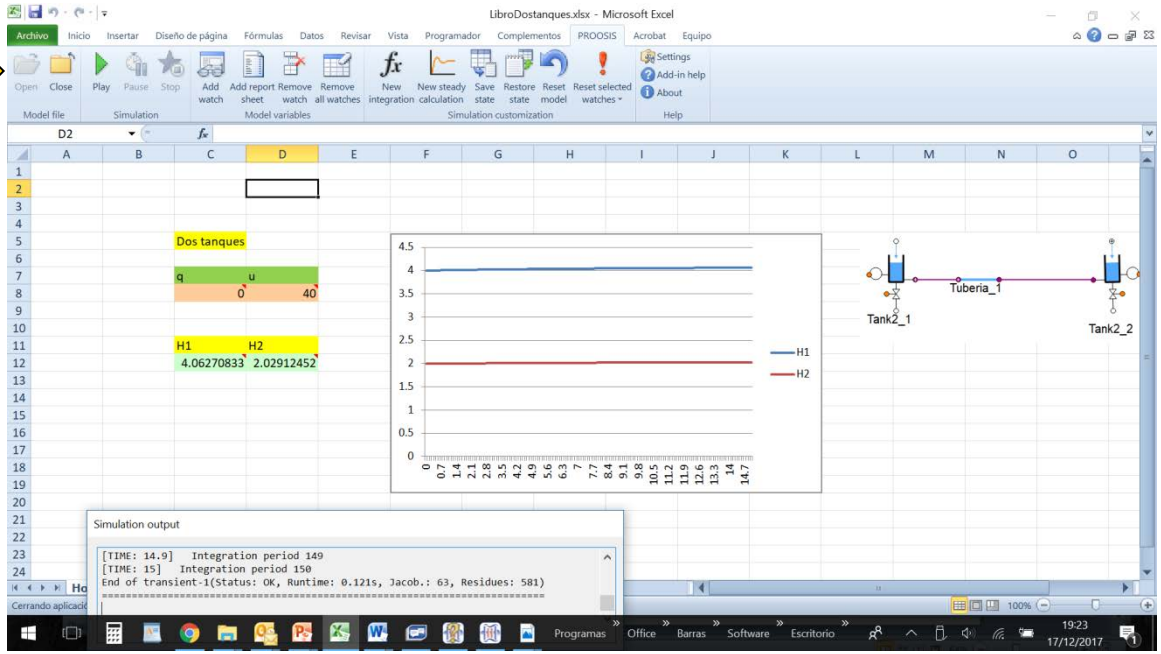
EcosimPro

Settings button



Running EcosimPro from Excel

1



1 Open an experiment

2 Design and configure the interface

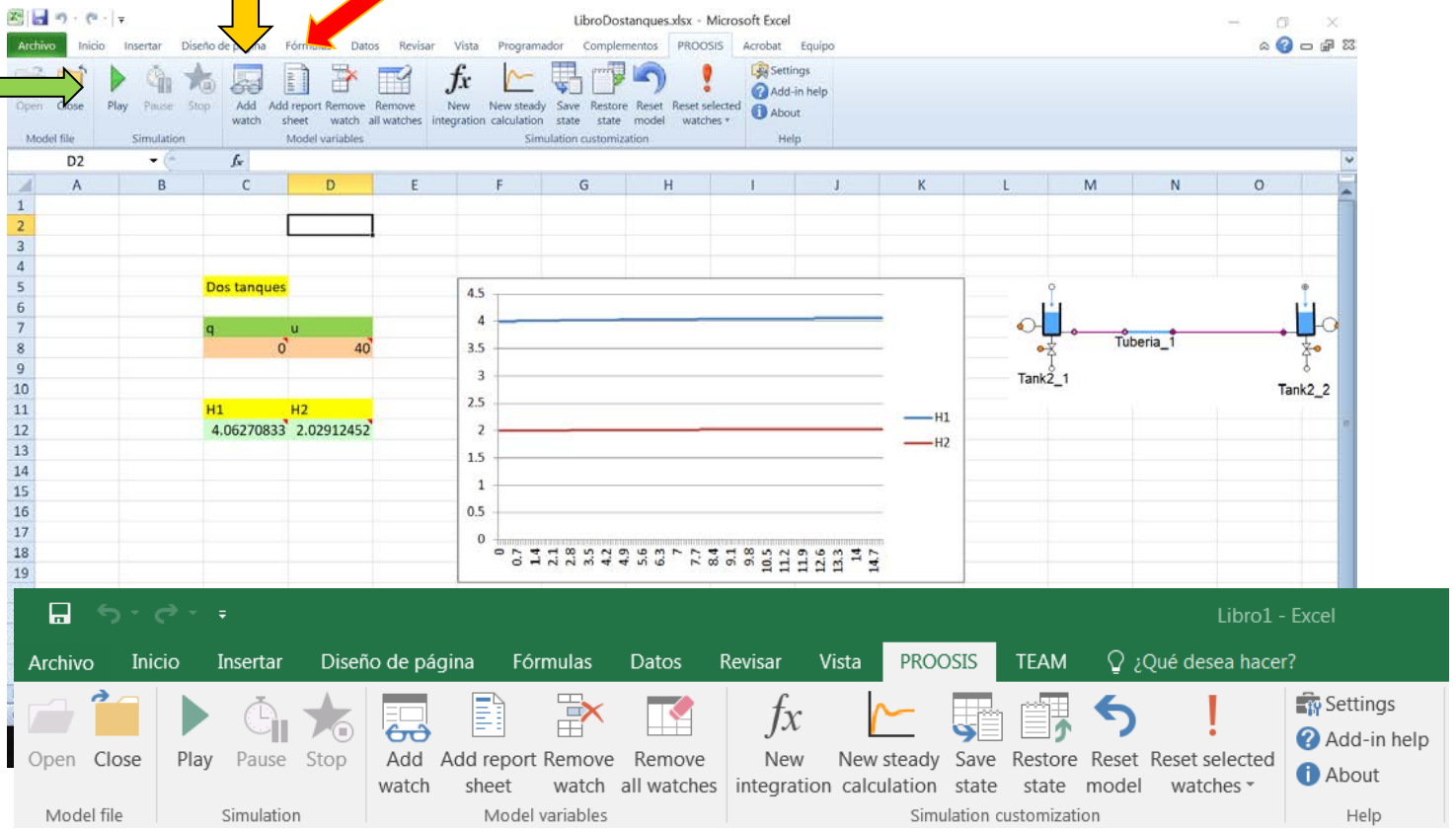
3 Run the simulation changing data and boundary variables and getting results from Ecosimpro

Interface

Assign cells to variables

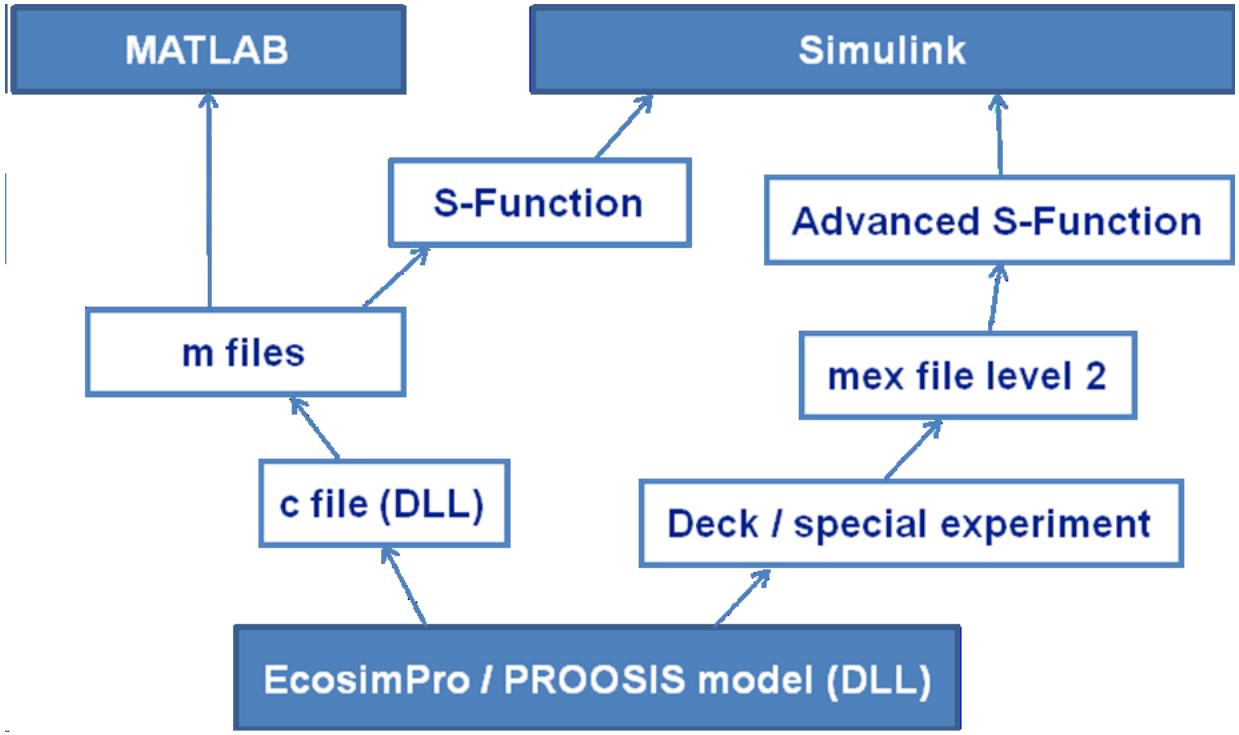
For graphs, Excel adds a new sheet for each variable with the reported values

Run



Matlab-Simulink

A dynamic link library file is generated when an experiment is compiled. It is identified by the name of the experiment plus a **.dll** extension.



Matlab-Simulink

A Toolbox provide finction for communication with Matlab. The following are some of the toolbox functions. This list can be obtained directly in MATLAB with the sentence:
help easim
EAGetValue: Get value of a variable of the model
EAGetVariableList: Get variable list of the model
EAINTEG: Run an integration
EALoadEASIM: Load the interface with models.
EALoadExperiment: Load a model from a DLL experiment file.
EAManual: Open user manual
EARESET: Reset the variables to the initial value
EARunExperiment: Run experiment body
EASetValue: Set value of a variable of the model
……

OPC

OPC is a series of standards specifications. Originally based on Microsoft's OLE COM (component object model) and DCOM (distributed component object model) technologies, the specification defined a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability.

The COM/DCOM technologies provided the framework for software products to be developed. There are now hundreds of OPC Data Access servers and clients.

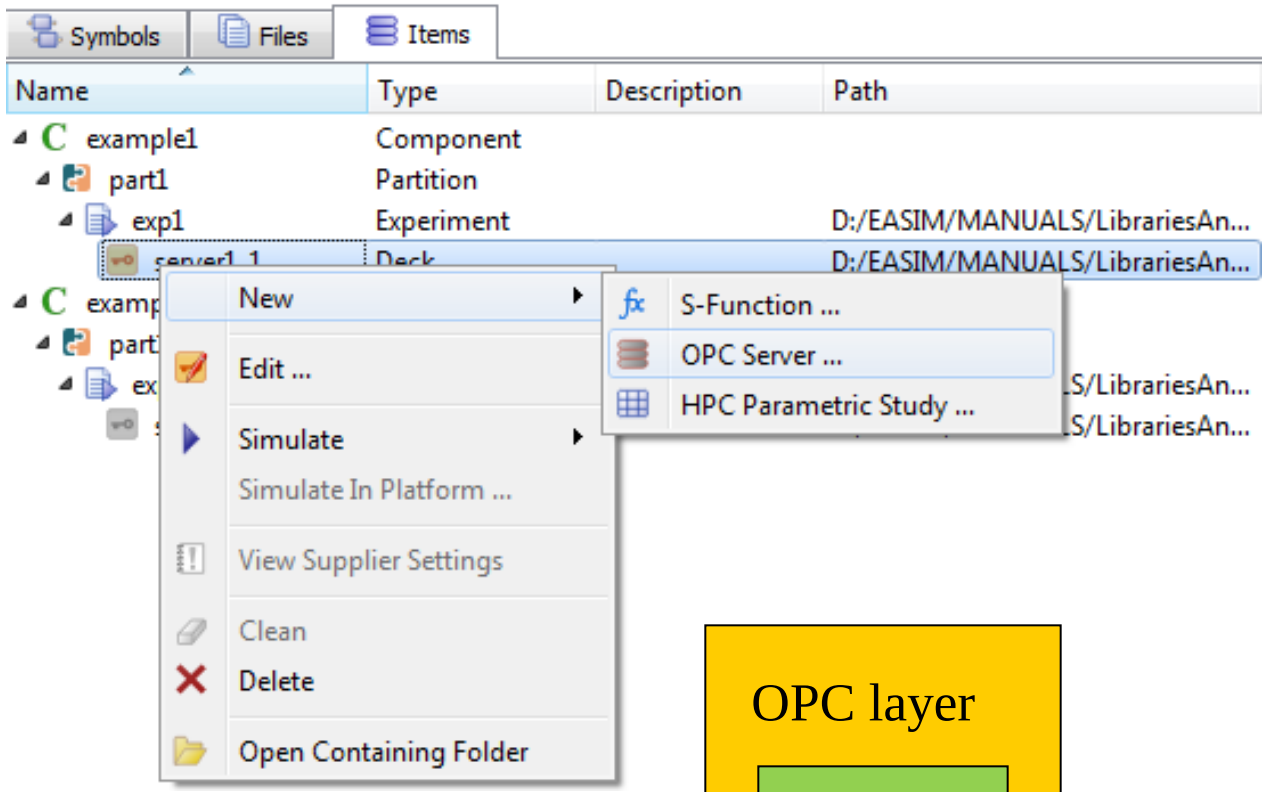
OPC

OPC Data Access (OPC DA) provides access to real time automation data. Using OPC DA, software applications can retrieve real-time data to enable them to monitor a given process. In that scenario, OPC DA is used for data acquisition.

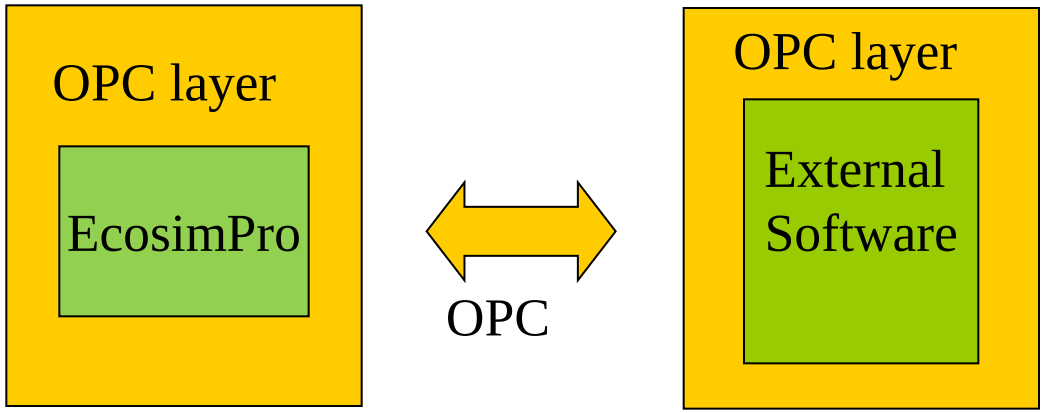
OPC DA can also be used to enable a software application to write data to a control system. This enables OPC DA applications to handle supervisory control. Thus, OPC DA is well suited for Supervisory Control and Data Acquisition (SCADA) scenarios.

OPC DA is typically used by almost all process control applications that are used as an HMI, Historian, Advanced Process Control (APC), Maintenance, ERP (Enterprise Resource Planning), Analysis, Optimization, etc.

OPC Toolbox

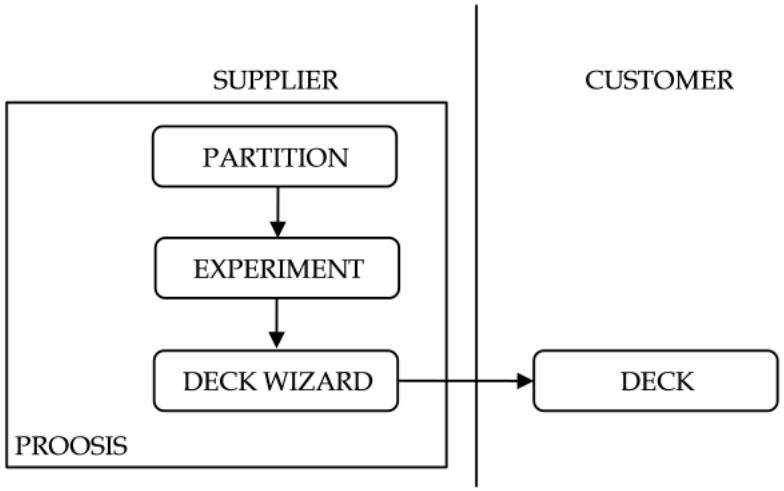


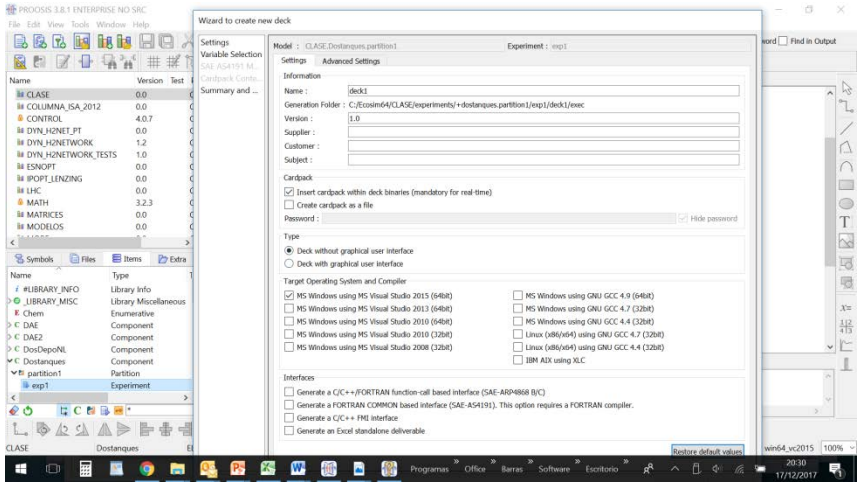
An OPC server can be generated to communicate the simulation with an external OPC client



Decks

A deck is a simulation model designed to run as a standalone black box totally independent from the main program. It encapsulates all its compile-time and run-time dependencies and allows the supplier (creator) to deliver his work to a customer (user) in a known way.





Wizard to create a new deck

FMI

FMI stands for Functional Mock-up interface, which is a standard C interface to instantiate one or more models and simulate them. FMU stands for Functional Mock-up Unit, which is a zip file that contains several files:

- A model description file in XML format which lets the tool that loads the FMU know about several aspects of the model.
- The model as a dynamic library (dll) implementing the FMI 2.0 for co-simulation standard and all its dependencies (support dlls or executables)
- Data files needed by the model.

The FMI standard defines two types of FMUs:

- **FMI for co-simulation**: the model has its own integrator. It is supported by Ecosimpro.
- FMI for model exchange: the model expects the simulator to do numerical integration. *This type of FMU can't be generated by EcosimPro*.

The idea behind FMI is to have a public, free and open interface that can be implemented by different simulation tool vendors so communication between models/tools is standardized hiding tool specific implementation.