

A Graphical Tool for Design Portable Automation Software

G. De Tommasi, P. Di Sanzo, A. Pironti

Università di Napoli "Federico II"
Dipartimento di Informatica e Sistemistica

21-23 June 2006, Madrid



Outline

- 1 Introduction
- 2 UniSim
- 3 Example
- 4 Conclusions



PLCs and industrial automation

- Control of industrial processes is today dominated by computerized systems
- Programmable Logic Controller (PLC) are widely used for industrial automation
- PLCs have been developed in the late 60's, and since then have evolved into powerful devices
- Teaching PLCs programming in industrial automation classes
- Students should learn how to:
 - design automation systems (methodology)
 - develop automation software (practice)



Automation projects and IEC 61131-3 standard

Automation project

An automation project includes all the information about the control system configuration, the data and the code

- Many manufacturers - many ways to define automation projects - many different programming languages
- The *IEC 61131* standard has been introduced by the *International Electrotechnical Commission*
- IEC 61131 - Part 3 specifies:
 - how an automation project should be structured
 - the programming languages that can be used by the developers



Teach IEC 61131-3 standard

- Although IEC 61131-3 has been introduced in 1993, only few manufacturer produce devices which fully complies with the standard
- Students who have learned IEC 61131-3, should be introduced to the platform available in the laboratory



UniSim

UniSim is an educational tool developed at University of Naples

- UniSim can be used to design automation software which complies with IEC 61131-3
- UniSim allows to avoid the choice of a specific commercial platform when teaching PLCs programming.
- Thanks to its *simulation engine*, UniSim allows to validate *off – line* the developed software
- UniSim can be used to *fast prototype* the automation systems using a desktop equipped with low-cost I/O boards



... moreover ...

- UniSim can be used
 - by the teacher - in the classroom to work out examples
 - by students - when solving their homework
- UniSim interfaces with off-the-shelf I/O boards, thus it can be used during lab activities
- Labs do not need to be equipped with a large number of expensive commercial PLC platforms
- UniSim makes use of the *XML Formats for IEC 61131-3* to import/export the projects. This feature give the possibility to reuse the developed software on a commercial platform
→ portability



UniSim - Snapshots

The screenshot displays the UniSim software interface for a 'Pump Control' simulation. The main window shows a control logic diagram with several pumps (Pump1, Pump2, Pump3, Pump4) and level indicators (L1, L2, L3, L4, L6). The diagram includes logic blocks for pumps and level indicators, connected by lines representing control signals. Key signals include 'S3 - Start', 'S5 - LevelDown(S1 b=10s)', 'S7 - true', 'S4 - Level1', 'S6 - Levels', 'S9 - Level2', and 'S8 - Level3'. The simulation is currently in the 'RUN' state, as indicated by the 'Simulation State' bar at the bottom, which also shows a speed of 0.004 Cycles/s.

Two data monitoring windows are open:

- PumpControlOutputs:** This window displays the actual values for four pumps. The data is as follows:

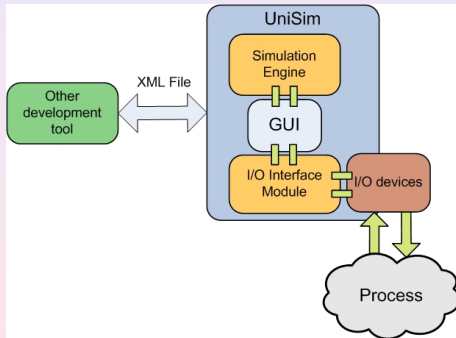
Type	Name	Actual value	Init
BOOL	Pump1	True	
BOOL	Pump2	False	
BOOL	Pump3	True	
BOOL	Pump4	True	
- MainInputs:** This window displays the actual values for various input signals. The data is as follows:

Type	Name	Actual value	Initial value
BOOL	Start	True	False
BOOL	Stop	False	False
BOOL	LevelUp	True	False
BOOL	LevelDown	False	False
BOOL	Liquid Temperature	False	False
BOOL	Ambient Temperature	True	False

The bottom status bar shows the simulation is running, with a speed of 0.004 Cycles/s. The status bar also includes buttons for Start, Stop, Step, and Reset, along with a Priority slider.



UniSim - Software architecture

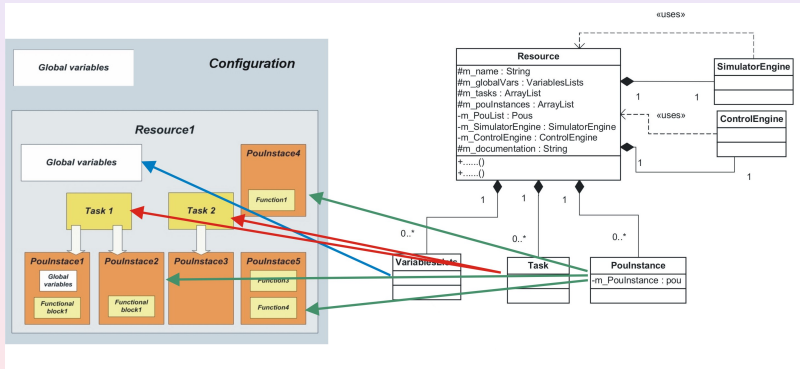


UniSim

- It has been designed by using an object-oriented approach
- It has been developed on the **.Net** platform

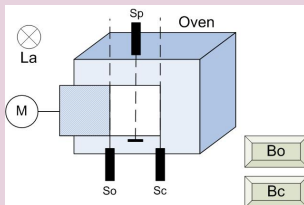


Objects hierarchy



Toy example

Oven

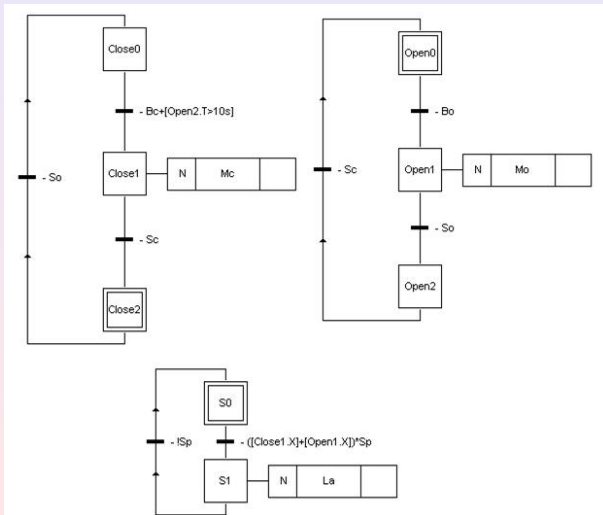


Requirements

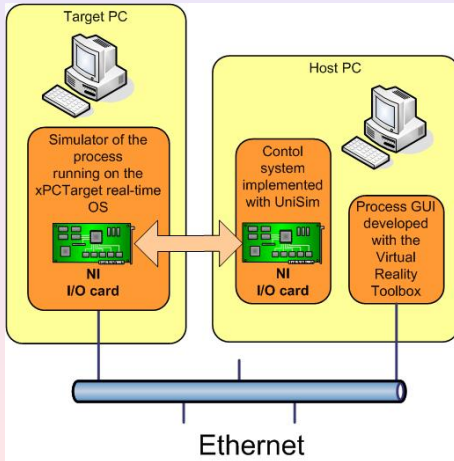
- The door must be opened when the button B_o is pushed
- The door must be closed when the button B_c is pushed, or if it stays open for more than 10s
- If an object is detected by S_p while the door is moving, lamp L_a must be turned on



Control algorithm



Lab setup



- The **Host** runs the control algorithm with UniSim and a graphical user interface for the process
- The **Target** runs the *xPC Target real-time OS*, which executes the process simulator



Conclusions

UniSim

- It allows to teach IEC 61131-3 standard without tying to any commercial platform
- It can be used to easily set up a lab for an industrial automation class using only *off-the-shelf* devices
- It uses *XML Formats for IEC 61131-3* guaranteeing software portability



Conclusions

But

- It is a *work in progress* release
Not all the features provided by the standard have been implemented
 - Only one *resource* can be specified in each *project*
 - Only the *sequential functional chart* and the *ladder diagram* languages have been implemented yet
 - ...
- The development of UniSim it is itself a way to teach IEC 61131-3 standard

By the way

UniSim is distributed with a *GPL* license

<http://wpage.unina.it/detommas/unisim>