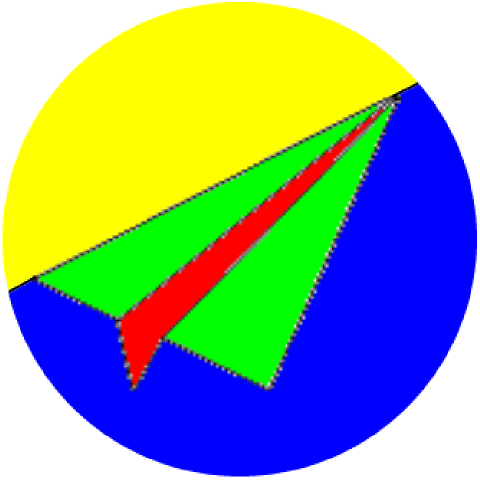
PROSIMUL



Graphic process and flow simulator

PRODUCT NAME : PROSIMUL	**EDITOR : CIMI** (www.cimi.fr)
ADRESS : 8, rue de l'Azin - 41018 BLOIS CEDEX	
TEL : +33 254 74 65 15	**CONTACTS :** Sylvain LISJAK
https://www.cimi.fr/contact.html	Ludovic BUTIN

Supported systems : Windows **XP/7/10/11-32/64bits**

SOFTWARE CARACTERISTICS

PLC connexion	Parallel : Binary or analog I/O. Serial : **Modbus** (RTU – **TCP/IP** – **C/S**), Unitelway, P3964, DF1 DH+ (Rslinx), **EthernetIP (C/S)**, Applicom (FIP, Profibus,…), MPI, **OPC client**, **S7-ISO on TCP**
Quantity of I/O	4096 Binary I/O, 2048 Analog I/O.
Compatible PLC	Parallel connexion : all Serial connexion : depending from available protocol.
Components modelling	Configurable process objects library (valves, motors, tanks, conveyors,...), and medias (pictures, sounds, GIF, videos, 3D objets). User library.
HMI modelling	Configurable HMI objects library (lamps, displays, switchs,...), pictures, User library.
Modelling languages	Ladder language: logical equations, timers, counts, calculation blocks Structured Text Language (For, While, If,…) Chart Language (2048 steps on 8 pages). Functions Blocks for continues flow and process. User library.
Process visualization	All process objects are animated with predefined rules. 16 graphics pages.
Fault generation	Variable modification, functions access (F01 to F12 / 48 combinations), Direct screen forcing (blocking, pulse, …). Random generation available.
Test sequence	Recordable manual interactions sequences. Programmable sequences.
Events storing	Timestamps trigged by state evolution. (absolute or relative time - precision 1 ms) - Export CSV and **Emails** transmission.
Documentation	All elements of project are printable. Trace of recorded events. I/O configuration.
Others	Standalone simulation available. Flow contents modelling available (Ladder / functions blocks). ***Real or virtual Time*** *simulation available (configurable time squale).* Language translation tool with integrated dictionary. Application protection by password.

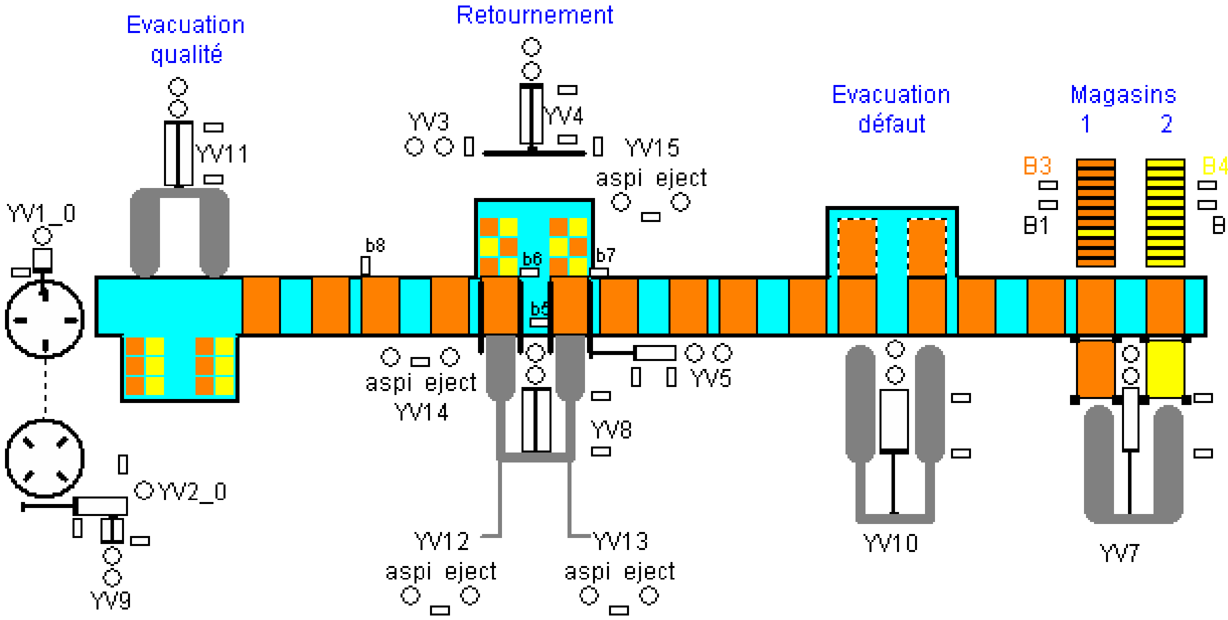
INTRODUCTION

PROSIMUL is a process and flow simulator developed by CIMI – CIFOP Val de Loire. This software is issued from the difficulty to work (training) efficiently on running or in development applications controlled by PLC. Actually, more than 300 process simulations have been build using **PROSIMUL**.

Developed with C++ language, for Windows (XP/7/10/11 – 32/64bits) it produces a very short time cycle on every basic computer and is compatible with a large range of PLC. The modelling method is as easy as possible and depends upon the technologic simulation level chosen by designer..
It can be used during all the production system cycle :

- Study draft, technological choices.
- Development, tests, adjustments.
- Maintenance, operating modes, training courses development.
- Process users and operators training.

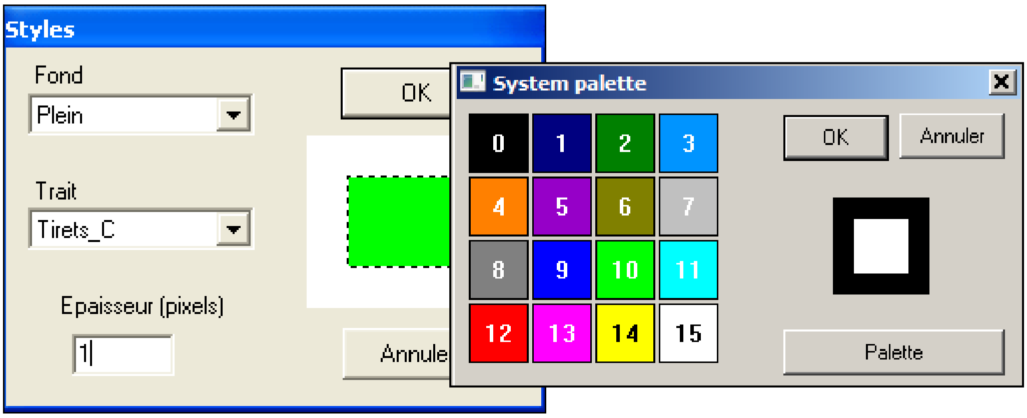
In a first step we describe main functions of **PROSIMUL** and then we look at some methodological elements that could aid process simulation development..



Graphic editor

That’s the graphic creation tool for process visualization.. It offers an objects library (valves, actuators, motors, tanks, conveyors,...). All objects are configurable and include their own animation properties (colours, movements, values, …) which are predefined as easy as possible.

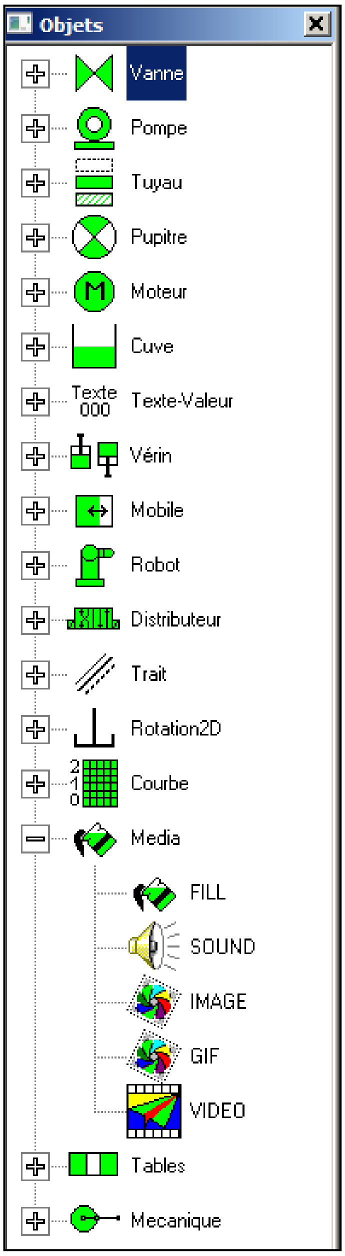
The complexity of process visualization is chosen by user. It can be very simple with texts or lamps for example, but it can be more complex with objects movement, curves, flow contents, ... This visualization is independent of process function modelling and can be modified without changing all the simulation. Furthermore, this tool can be used for only process supervision.



Fond: Plein

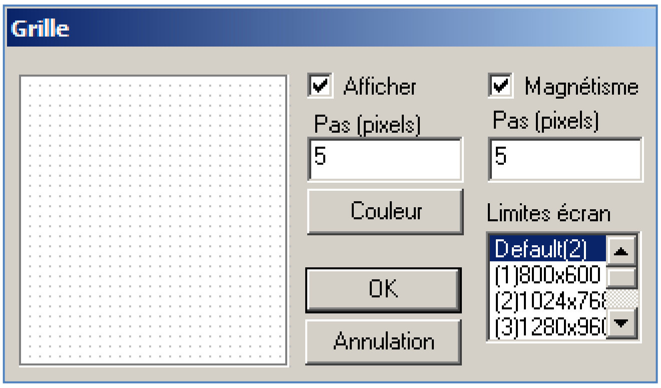
Trait: Tirets_C

Epaisseur (pixels): 1



All objects can be combined and stored as an user library used by others applications. Many views can be produced for one application like alarm pages or synoptics.

The editor includes grouping, magnetism, copy/paste and dimensioning functions to allow rapid development phase.



☑ Afficher

Pas (pixels): 5

☑ Magnétisme

Pas (pixels): 5

Limites écran: Default(2)
(1)800x600
(2)1024x76
(3)1280x96

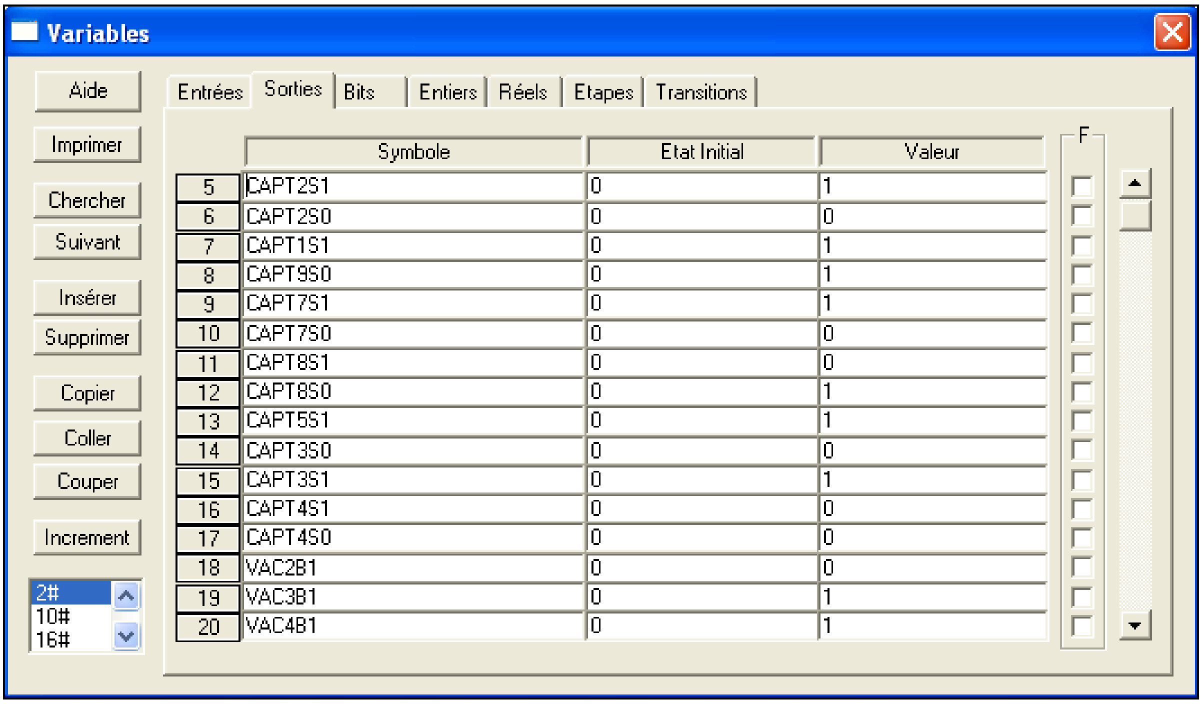
A tool is available for automatic language translation based on an internal dictionary.

The Data Base

This function is dedicated for internal variables configuration. **PROSIMUL** provides view tables for inputs (coming from PLC), outputs (to PLC), bits and internals words (numeric or analog). Finally, 4096 Binary I/O, 2048 bits, 2048 integers (32 bits), 2048 reals (32 bits), 2048 chart steps and transition are available in the standard version of software.

The first operation is to configure **PROSIMUL** I/O (parallels and or serials) in relation with PLC I/O (same order and types). Those variables can receive PLC symbols thru a neutral text format file.

Those variables will be used for graphic animation and process modelling to create memory functions, timers, counters, calculation results or discrete function blocks. Some variables are reserved for system runtime like time base, virtual time factor, system and chart initialization, predefined colours or programmed pages access.



	Symbole	Etat Initial	Valeur	F
5	CAPT2S1	0	1	☐
6	CAPT2S0	0	0	☐
7	CAPT1S1	0	1	☐
8	CAPT9S0	0	1	☐
9	CAPT7S1	0	1	☐
10	CAPT7S0	0	0	☐
11	CAPT8S1	0	0	☐
12	CAPT8S0	0	1	☐
13	CAPT5S1	0	1	☐
14	CAPT3S0	0	0	☐
15	CAPT3S1	0	1	☐
16	CAPT4S1	0	0	☐
17	CAPT4S0	0	0	☐
18	VAC2B1	0	0	☐
19	VAC3B1	0	1	☐
20	VAC4B1	0	1	☐

An initial state can be given to each variable, that can be modified or stored during running state application.

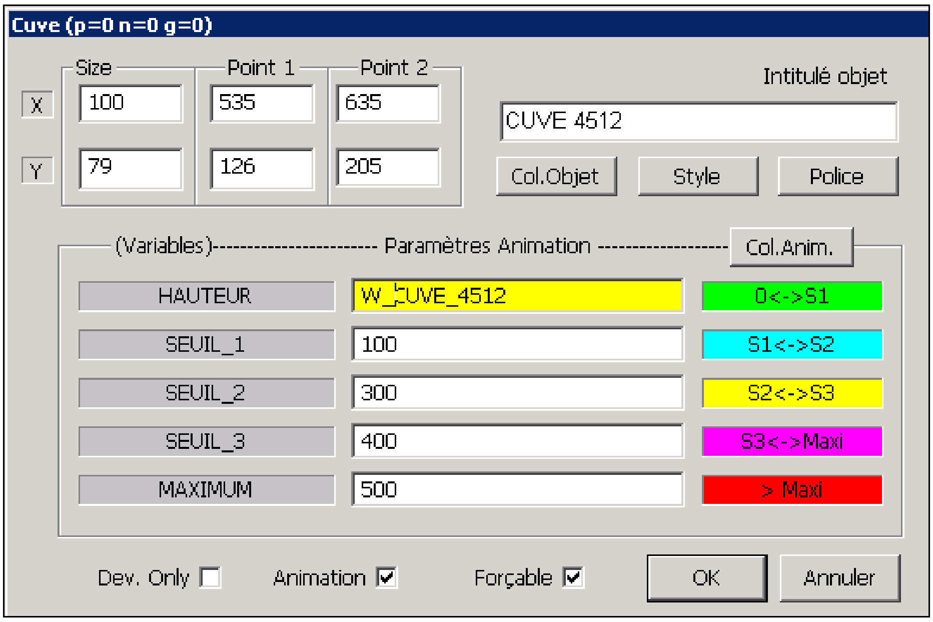
Additional to programmed actions, the user can adjust all variables by screen clicks or functions keys. They can also be selected to appear in special windows box for forcing or timestamps function.

Graphic animations

This phase consist in making relation between graphic objects and database variables depending from the predefined rules established.. Graphic objects integrate there own animation properties that are trigged by **PROSIMUL** variables state evolution. It’s possible to quickly create simply views or more realistic synoptics integrating for example bitmaps (GDI library).

The animation properties can be fully or partially used, and some special effects can be produced by objects superposition or combination.

For example in the case of “tank”, user can use one word containing the level of the object depending of the maximum value. Different thresholds are available for changing colour object. This object can be use for tank representation but also for bargraph value or flow contents in pipe.



Cuve (p=0 n=0 g=0)

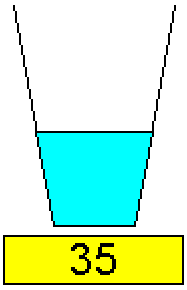
	Size	Point 1	Point 2
X	100	535	635
Y	79	126	205

Intitulé objet: CUVE 4512

(Variables)---------------------- Paramètres Animation ------------------

HAUTEUR	W_CUVE_4512	0<->S1
SEUIL_1	100	S1<->S2
SEUIL_2	300	S2<->S3
SEUIL_3	400	S3<->Maxi
MAXIMUM	500	> Maxi

Dev. Only □ Animation ☑ Forçable ☑

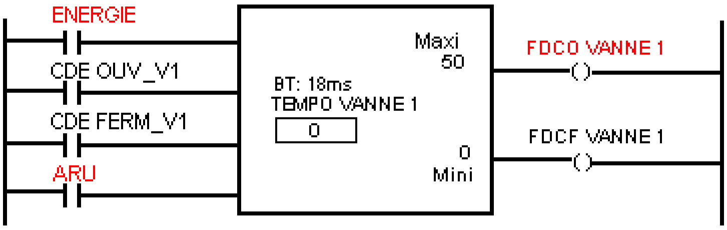


Various dynamics views are available depending of process technologic components like mechanic systems, electric devices or flows.

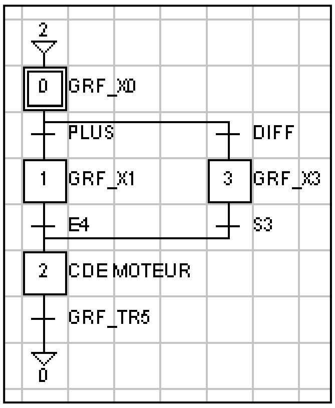
Some objects can be controlled by mouse actions (value, pushbutton, switch, ..) and one configurable system parameter allow to hide objects when application is running in training mode (by opposite of development mode).

The Ladder, Structured Text, Chart and Functions blocks editors

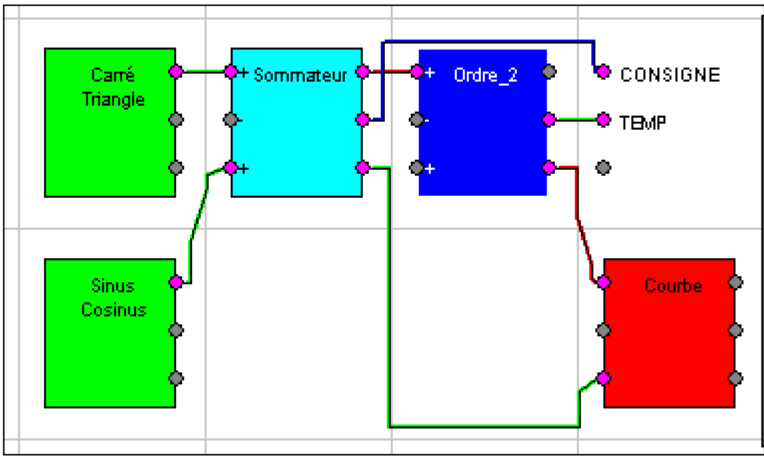
This phase is the translation of process to emulation program. Four languages are available for modelling movements and product flows. The LADDER language can easily describe actuators (timers with 2 commands and 2 sensors for example) or logical combinations.

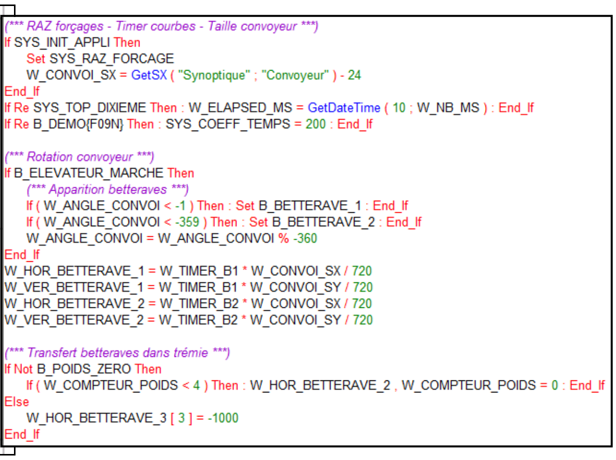


The CHART language (2048 steps and transitions) is well adapted for sequential functions. It can also help to test the control program without connected PLC (standalone configuration).
One tool is available to valid the built structure of chart.



For continuous process, with temperature or flow control, the DISCRETE FUNCTION BLOCKS language offers many possibilities to simply work.
It offers 4 hierarchical levels of discrete blocks like integrator, derivator, filters and order N functions..





```
(*** RAZ forçages - Timer courbes - Taille convoyeur ***)
If SYS_INIT_APPLI Then
    Set SYS_RAZ_FORCAGE
    W_CONVOI_SX = GetSX ( "Synoptique" ; "Convoyeur" ) - 24
End_If
If Re SYS_TOP_DIXIEME Then : W_ELAPSED_MS = GetDateTime ( 10 ; W_NB_MS ) : End_If
If Re B_DEMO{F09N} Then : SYS_COEFF_TEMPS = 200 : End_If

(*** Rotation convoyeur ***)
If B_ELEVATEUR_MARCHE Then
    (*** Apparition betteraves ***)
    If ( W_ANGLE_CONVOI < -1 ) Then : Set B_BETTERAVE_1 : End_If
    If ( W_ANGLE_CONVOI < -359 ) Then : Set B_BETTERAVE_2 : End_If
    W_ANGLE_CONVOI = W_ANGLE_CONVOI % -360
End_If
W_HOR_BETTERAVE_1 = W_TIMER_B1 * W_CONVOI_SX / 720
W_VER_BETTERAVE_1 = W_TIMER_B1 * W_CONVOI_SY / 720
W_HOR_BETTERAVE_2 = W_TIMER_B2 * W_CONVOI_SX / 720
W_VER_BETTERAVE_2 = W_TIMER_B2 * W_CONVOI_SY / 720

(*** Transfert betteraves dans trémie ***)
If Not B_POIDS_ZERO Then
    If ( W_COMPTEUR_POIDS < 4 ) Then : W_HOR_BETTERAVE_2 , W_COMPTEUR_POIDS = 0 : End_If
Else
    W_HOR_BETTERAVE_3 [ 3 ] = -1000
End_If
```

The STRUCTURED TEXT language can solve any problem including iterative elements and discrete flow with a classic programming language.

All those editors are the toolbox for dynamic process description depending from the chosen complexity level. For example for a translation actuator, the movement can be linear (only time proportional), or divided in some linear movements. It can also be modelled by an non linear function if necessary.
The best process components knowledge is very important during this modelling phase. One visible benefit is also to store this knowledge in the simulator.

PLC communication configuration

PROSIMUL allow PLC connecting by few ways ::

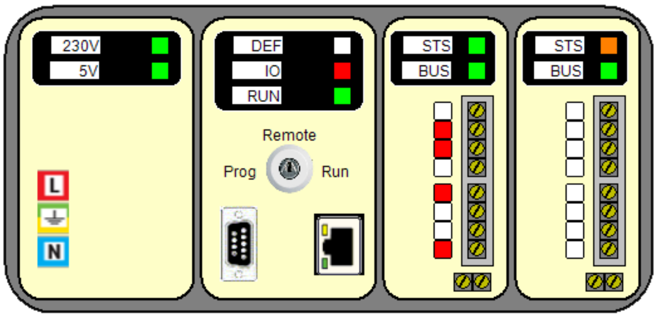
- Binary or analog physical cards :
  - Compex,
  - ADLINK,
  - National Instrument,
  - …

- Serial integrated communication
  - **MODBUS** (April, Schneider, Siemens, Wago...) – RTU – **TCP/IP** (client and server)
  - UNITELWAY 1 & 2 (Schneider),
  - DF1 (Rockwell SLC & PLC & ControlLogix),
  - P3964 (Siemens S5 & S7).
  - **EthernetIP** (Rockwell – client and server)
  - **S7-ISO on TCP** (Siemens)

- Serial communication with extern drivers :
  - SIEMENS –S7-ONLINE link (MPI, Profibus,…),
  - ROCKWELL – RSLINX,
  - **Client OPC** (ROCKWELL RSLINX, SCHNEIDER OFS, SIEMENS SimaticNET, WoodHead, B&R, KEPserver, MATRIKON, …),
  - APPLICOM –cards with drivers (FIP, Profibus, Ethernet,..), Direct-Link SW1000.

In parallel link, one or more binary cards are plugged in the computer and I/O are directly connected to PLC cards. That is the more realistic possibility which offer fast time cycle. However this solution is limited to low quantity I/O (cost and space). The avantages is to work with I/O LEDS for diagnostic.
C cards are simply configured in relation with **PROSIMUL** database order.

In case of serial communication, developer have to configure an exchange table including binary value and numeric words that are Read / Write on the PLC. This mode requires a sub-routine for organizing and delivering the table in the internal PLC data.



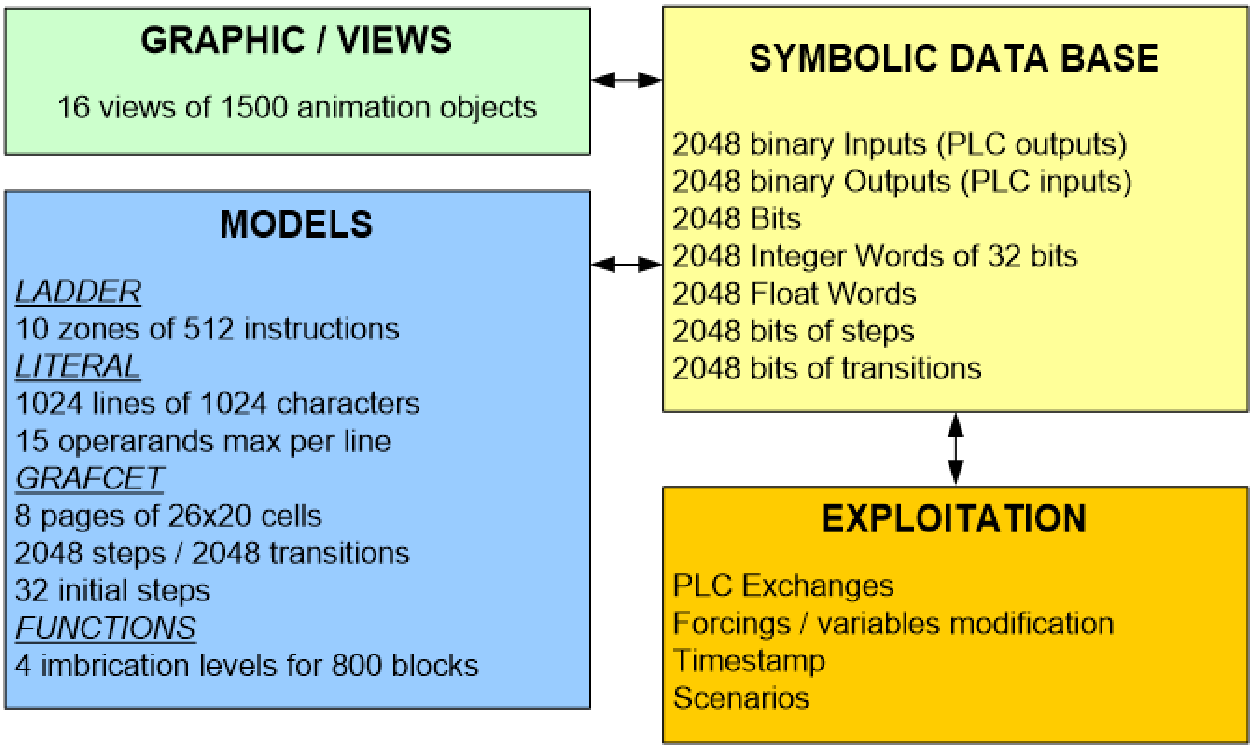
Parallel and serial links can work simultaneously, to allow fast exchanges (< 1ms) for some I/O and slowest communications (20 to 100 ms depending upon used protocol) for large quantity data.

Furthermore, it is possible to avoid any PLC communication in order to verify standalone process (actions done by forcing) or simulator with integrated control program.

Runtime exploitation

After standalone simulation validation, the PLC connexion can be done and **PROSIMUL** switched in running mode. The running cycle is composed by few operations : it first reads PLC commands, then computes models program, and finally delivers sensors to PLC. All views are refreshed in parallel.

All **PROSIMUL** variables can be modified or forced to test component failure and PLC response at any dysfunction (like alarm display for example). All automatic or manual process working modes can be reproduced to fully valid the control program before on-site operation. All process variables and active steps can be visualised and dynamically modified.



GRAPHIC / VIEWS

16 views of 1500 animation objects

MODELS

LADDER
10 zones of 512 instructions
LITERAL
1024 lines of 1024 characters
15 operarands max per line
GRAFCET
8 pages of 26x20 cells
2048 steps / 2048 transitions
32 initial steps
FUNCTIONS
4 imbrication levels for 800 blocks

SYMBOLIC DATA BASE

2048 binary Inputs (PLC outputs)
2048 binary Outputs (PLC inputs)
2048 Bits
2048 Integer Words of 32 bits
2048 Float Words
2048 bits of steps
2048 bits of transitions

EXPLOITATION

PLC Exchanges
Forcings / variables modification
Timestamp
Scenarios

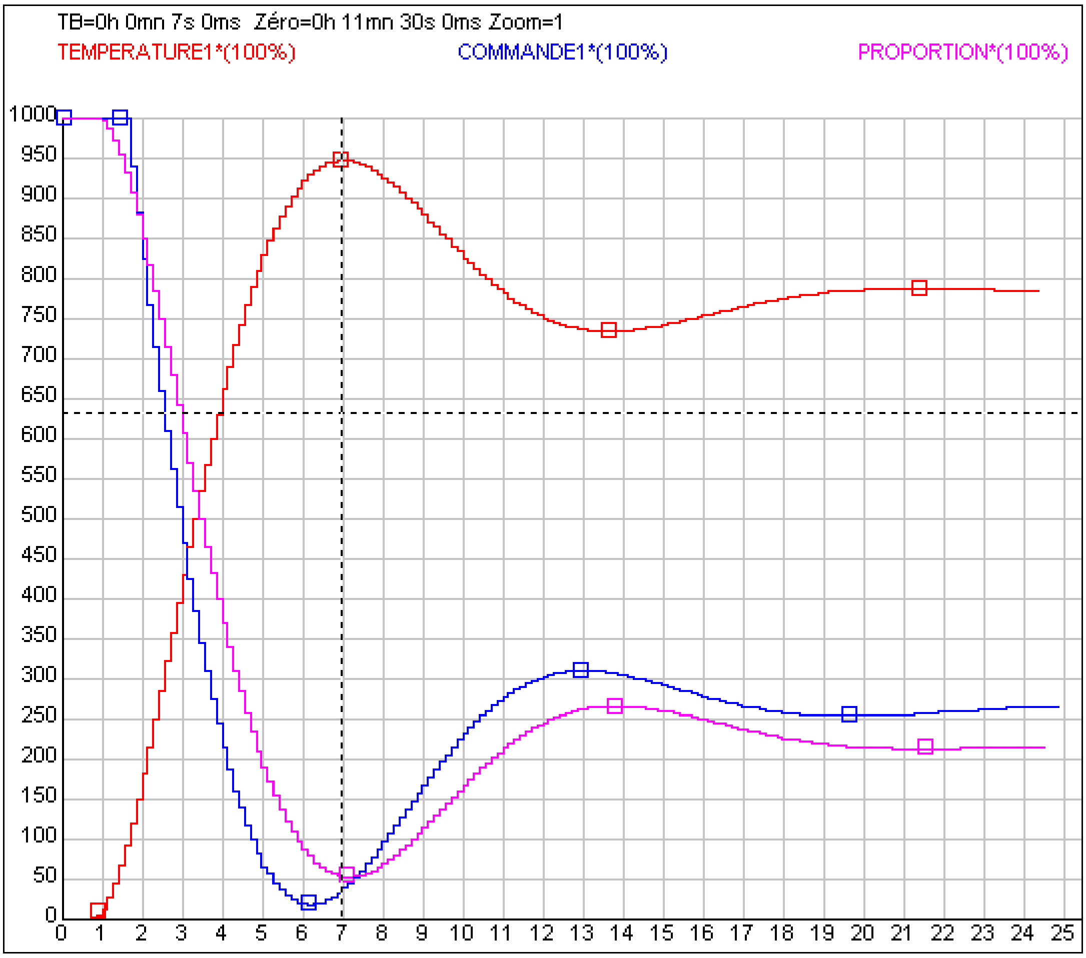
Some test sequence (manual or automatic actions) can be stored and played, to limit screens or keyboard function actions. Those monitoring records can be analyzed thru an integrated tool explained next page or used with external tool. Rapports or alarms events can be transmitted by emails.

The testing phase can be executed in real or virtual time through a system parameter. Some tests can be accelerated (processes with large time factor) or decelerated to catch or observe specific process states.

The Monitoring function

In order to thinly analyze process states, the timestamps function allows to store all applications values (binary and words) on each database evolution with a time accuracy of 1ms.

Records can be exported or exploited thru an integrated graphic tool. Time scale is automatically adjusted depending of the selected variables. Abnormal states can be detected by analyzing command time, sensor duration, securities, and undesirable or dangerous events combination.



Time axis can be zoomed and some special points (minima, maxima, inflection) are available for display if necessary.

In case of continuous process modelling, an identification function is available to compare the process response as nearly as possible to a N (2 to 12) degrees polynôme. This result can be introduced in a polynôme function block in order to evaluate PID regulator efficiency for example.

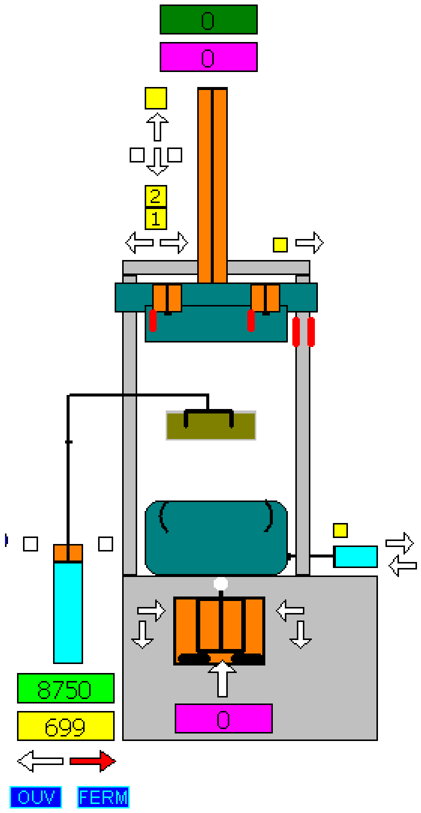
Elements of methodology

Simulation project requires a fine knowledge of process components and functions. Developers must integrate this first important point in order to build a realistic approximated model but not the ideal process forgetting some important physical constraints.

The first step consists to list up all actuators and sensors connected to the PLC (this step is easy if I/O organization is done) and to identify their technology. Then the realistic modelling level (logical conditions for working components, energies, mechanic relations, securities), and the graphics views with interactions (oriented for development or training courses) must be specified.

Flow contents and product transformations are the most difficult parts to analyze. It is an evidence that a large or infinite product flow can not be simulated according to the limited computer resources (for example : continuous flow of products on a conveyor). Developers must build its simulation including real conditions of insertion, progression and product transformation, according to the necessity to divide process by flow relations or to “recycle” final product.
In some complex case, process flow contents can be limited if non essential, and the simulation can only deliver an emulated sensor function, according to the information expected by PLC.

The database generation, in relation with PLC I/O, is a very fast step. It can be done by text file export/import operation. Some internal value (timers, counters, conditions) can be directly created in the database to facilitate the next steps.



After process analysis, developer can design and draw graphic views according to predefined rules and graphic usable interactions.

The modelling program step requires a large technologic knowledge to allow a lot of process simplifications without changing the main functions. The four languages available offer various solutions and the application can be tested every time in standalone configuration. The purpose is to verify gradually and one by one the object functions as to prevent superposing incorrect simulations and controller faults.

Also, in case of process containing multiple identical sub-systems, referent one must be fully tested before being duplicated and stored in library.

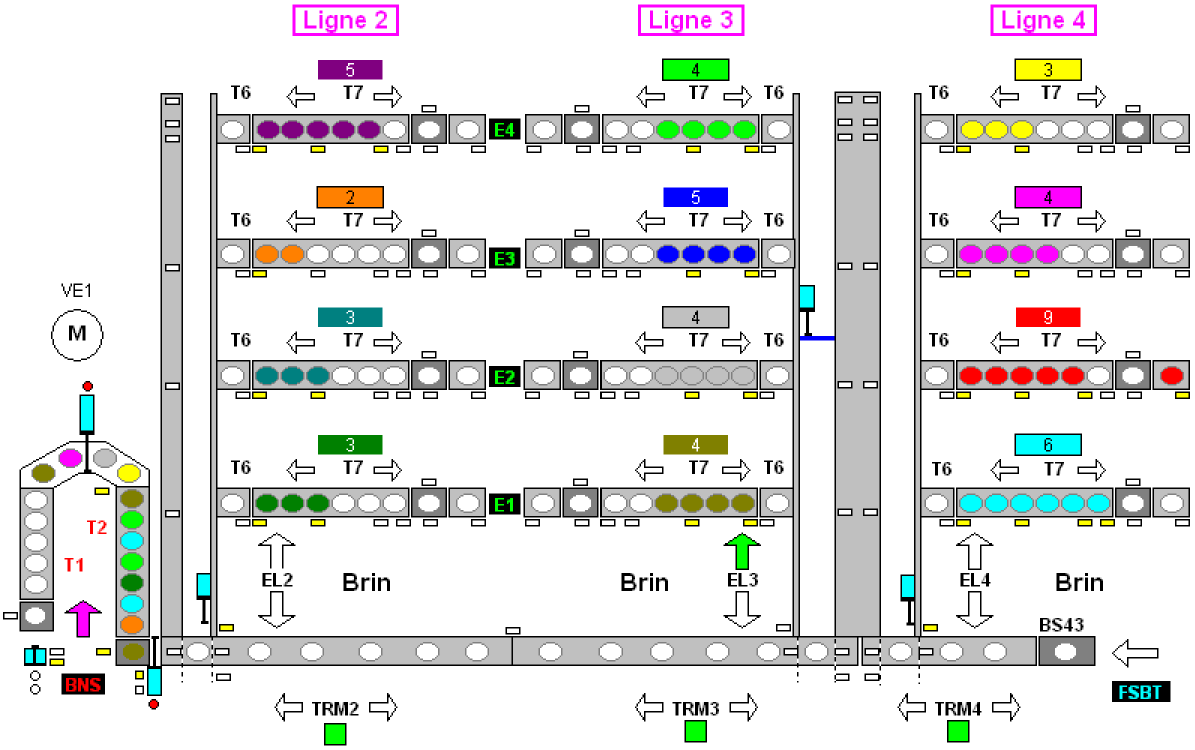
CONCLUSIONS

The success of a simulation project is based upon a good understanding of the processes and their components, but also on the ability to produce an efficiency function analysis.
This investment does not easily appear economically or technically attractive. The traditional tools for testing (button box, leds, …) are quickly operable and appear economic and easy to use. However, they allow only simple an non dynamic tests of process and are conditioned by a lot of manual actions.

The advantages of a software simulation like **PROSIMUL** can be displayed in many ways. Based upon our experiences, only one developer-day is necessary to build one application involving 64 binary I/O.
This point of view has to be in relation with standard off-site and on-site time test (lasting many weeks sometimes, with stressful conditions). We also do not forget the time required by operators and maintenance staff to learn procedure and drive efficiently their process.
Free energy consumptions, avoided product losses, and human or pollution risks reduced are also factors that improve simulation project interest.

The initial development, usually design for technical services can be used for testing an validating the control program, and may interest the final users of the process. It is a system knowledge base that offers ability of learning and dialog between designers and operators.

A large part of security, quality and process efficiency benefits depends upon the ability of users to observe and understand the processes they are working on. Simulation is certainly the best answer for your queries…


Ligne 2
Ligne 3
Ligne 4
5
4
3
T6
T7
E4
2
5
4
E3
VE1
M
3
4
9
E2
3
4
6
E1
T2
T1
BNS
EL2
Brin
Brin
EL3
EL4
Brin
BS43
FSBT
TRM2
TRM3
TRM4