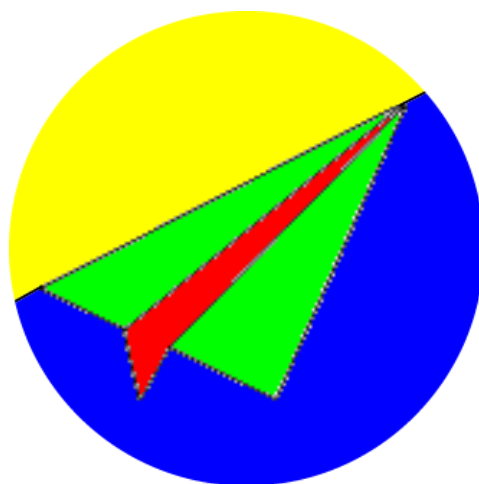




# PROSIMUL



## Logiciel graphique de simulation de procédés et de flux





<b>NOM DU PRODUIT : PROSIMUL</b>	<b>SOCIETE : CIMI</b> (www.cimi.fr)
<b>ADRESSE : 8, rue de l'Azin - 41018 BLOIS CEDEX</b>	
<b>TEL : 02.54 74 65 15</b>	<b>INTERLOCUTEURS : Sylvain LISJAK</b>
<b><a href="https://www.cimi.fr/contact.html">https://www.cimi.fr/contact.html</a></b>	<b>Ludovic BUTIN</b>

<b>Systèmes supportés :</b>	Windows <b>XP/7/10/10-32/64bits</b>
-----------------------------	-------------------------------------

## CARACTERISTIQUES DU LOGICIEL

Type de raccordement avec l'API	Entrées/sorties TOR, ANA. <b>Modbus</b> (RTU – TCP/IP C/S), Unitelway, P3964, DF1 DH+ (Rslinx), <b>EthernetIP</b> (C/S), Applicom (FIP, Profibus,...), S7-MPI, <b>Client OPC, S7 ISO on TCP</b>
Nature des entrées/sorties	TOR et analogiques, 4096 E/S TOR, 2048 E/S ANA.
Type d'API supportés	Connexion parallèle : tout type Connexion série : suivant protocoles ci-dessus ou cartes Applicom (Fip, Profibus, Ethernet...) ou Client OPC
Modélisation des organes	Bibliothèque d'objets process paramétrables (vannes, vérins, moteurs, cuves, convoyeurs,...), images, sons, vidéos, objets 3D. Bibliothèque utilisateur.
Modélisation de pupitre	Bibliothèque d'objets pupitre paramétrables (lampes, afficheurs, boutons poussoirs,...), bitmaps, curseurs, vumètres, ... Bibliothèque utilisateur.
Langages de modélisation	Langage Ladder : équations, temporisations, compteurs, blocs calcul (1 Main + 8 SR de 256 instructions maximum) Langage Littéral évolué (Structure For, While, If,...) Langage Grafcet (2048 étapes sur 8 pages). Blocs Fonctions pour flux et process continu. Bibliothèque utilisateur.
Visualisation de la Partie Opérative	Graphiques animés, courbes, voyants, messages, images, vidéos,...- 16 pages graphiques.
Génération de défaut	Modifications de variables, touches fonctions (48 combinaisons), Forçage sur écran de capteurs/actionneurs (blocage, impulsion,...). Génération aléatoire possible.
Scénarios de test	Enregistrement et reproduction d'actions manuelles. Scénarios programmables.
Consignation d'évènements	Horodatages sur changement d'état, trace, fil de l'eau en temps absolu et relatif (précision 1 ms). Export au format texte, transmission d' <b>email</b>
Documentation	Documents graphiques et descriptifs de la simulation. Fils de l'eau et tracés des enregistrements. Affectation des entrées/sorties.
Divers	Test de la simulation en autonome possible. Flux de matière simulable en Ladder et Blocs Fonction. <i>Simulation en <b>Temps réel ou fictif réglable</b> (accéléré / ralenti).</i> Outil de traduction par dictionnaire intégré. Protection des applications par mot de passe.





## PRESENTATION

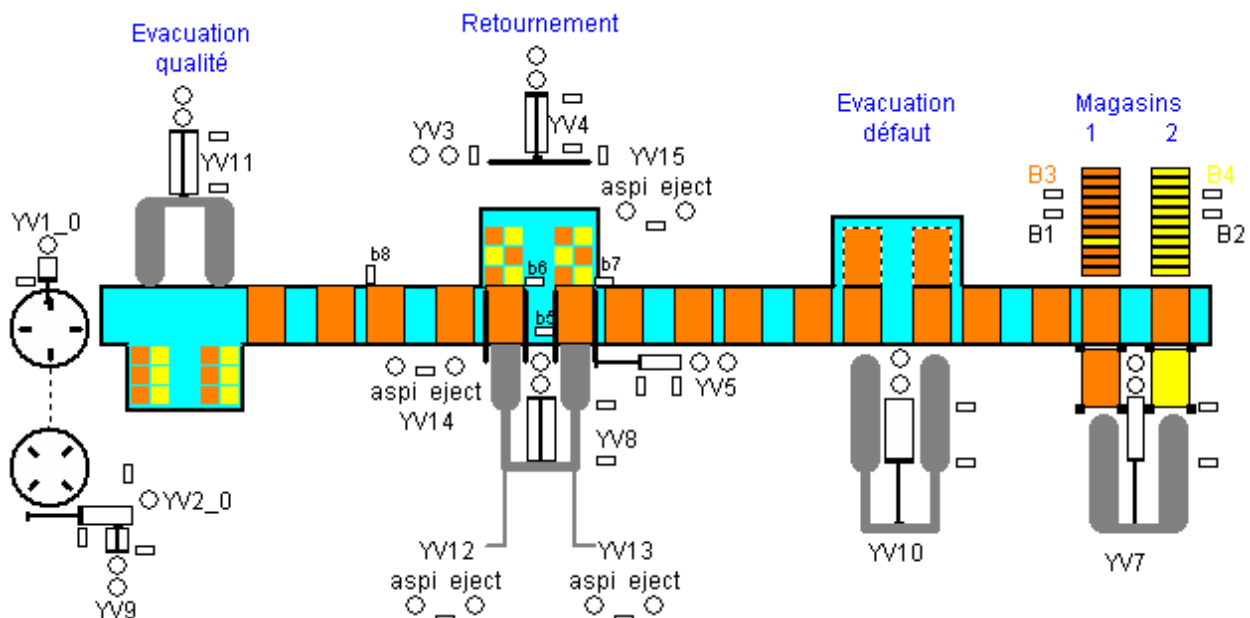
**PROSIMUL** est un outil de simulation de partie opérative développé par le CIMI – CIFOP Val de Loire. La nécessité d'intervenir efficacement sur des applications pilotées par automates programmables existantes ou en cours de mise en service a conduit le CIMI à construire un produit dont l'utilisation industrielle ou pédagogique a déjà fait ses preuves (plus de 400 logiciels en service).

Développé en C++ pour les plateformes Windows (XP/7/10/11-32/64bits), il offre un temps de cycle performant sur tout PC, sans nécessiter de configuration matérielle particulière de l'ordinateur ni de l'API. Cet outil propose à l'utilisateur de larges possibilités de simulation dont le niveau de complexité est déterminé par le choix du concepteur.

L'un de ses intérêts est de s'adapter au cycle de vie du process :

- Avant-projet, étude des choix technologiques.
- Développement, essais, mises au point.
- Maintenance, procédures d'intervention, entraînement hors site.
- Formation des exploitants et conducteurs.

Nous regarderons dans un premier temps les fonctionnalités de **PROSIMUL** puis nous indiquerons quelques aspects méthodologiques, facilitant le développement d'une simulation de partie opérative.

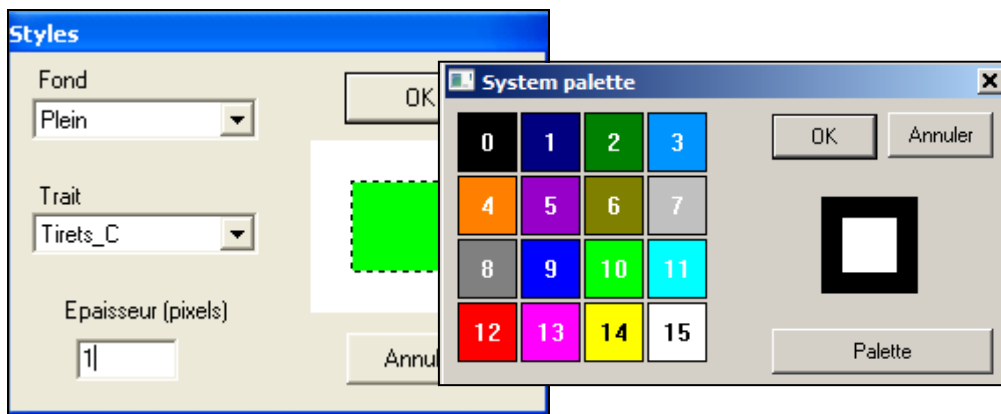




## L'éditeur graphique

C'est l'outil de création graphique de la partie opérative. Il offre une bibliothèque d'objets (vannes, vérins, moteurs, voyants,...) paramétrables par l'utilisateur. Chaque objet possède ses caractéristiques d'animation (changements de couleurs, mouvements,...) qui lui sont propres car définies par leurs fonctions.

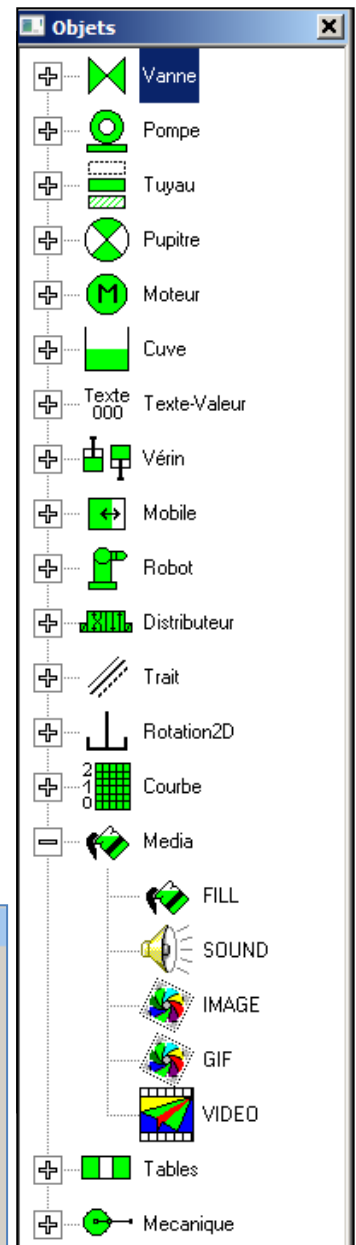
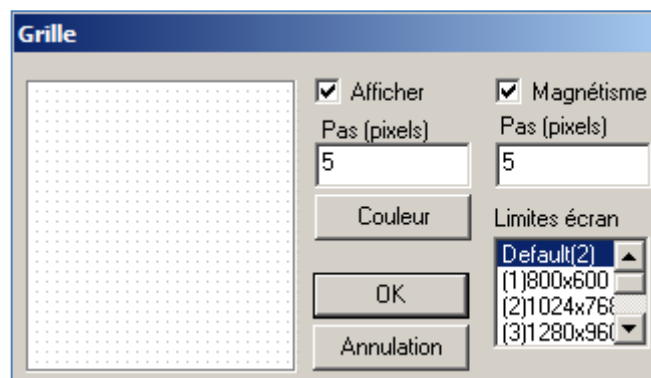
La représentation du process est donc au choix de l'utilisateur, simple sous forme de voyants ou de textes, ou plus complexe sous forme de mouvements, courbes, déplacements d'objets. Elle est de plus indépendante de la modélisation ou description du fonctionnement du système productif et peut être modifiée sans remise en cause de la simulation entière.



Les objets peuvent être composés et sauvegardés sous forme d'une bibliothèque utilisateur et ré exploités pour d'autres applications. Plusieurs vues graphiques sont envisageables pour une même application telles que un pupitre de commande, des alarmes ou différents synoptiques.

L'éditeur complet offre des fonctions de groupement, magnétisme, copier/coller ou re-dimensionnement d'objets permettant un développement des graphiques en un temps record.

Un outil permet de traduire automatiquement les pages dans différentes langues (Unicode) selon un dictionnaire intégré.



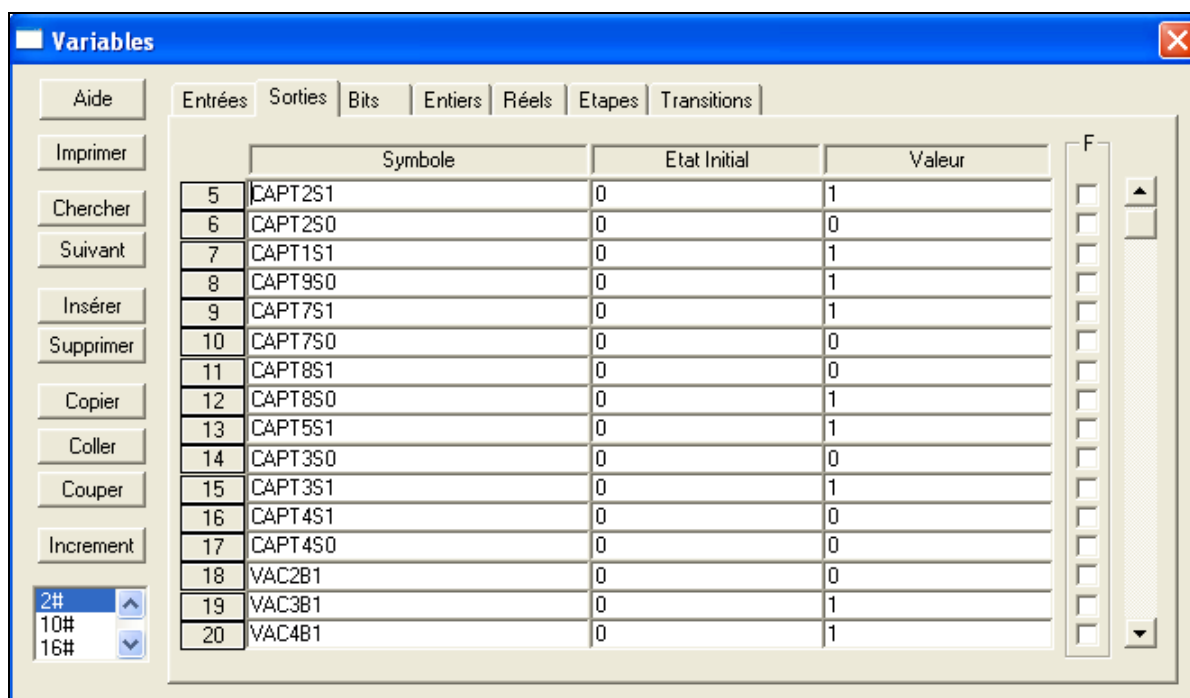


## La base de variables

Cette fonction permet l'accès à la configuration des variables internes de l'application. **PROSIMUL** met à disposition des variables d'entrées (issues de l'API), de sorties (destinées à l'API), des bits et des mots internes pouvant contenir des valeurs numériques (codeurs) ou analogiques. Au total 4096 E/S TOR, 2048 bits, 2048 mots 32bits et 2048 mots réels sont à disposition dans la version standard du logiciel.

La première opération consiste à faire coïncider, en fonction du type d'échange (parallèle ou série), les entrées / sorties **PROSIMUL** avec celles de l'automate. Ces variables peuvent récupérer les symboles API via un fichier texte neutre.

Elles seront exploitées lors de la phase d'animation des synoptiques et de modélisation des procédés et pourront être utilisées sous forme de mémoires, temporisations, compteurs, résultats de calcul ou de blocs fonction. Quelques variables systèmes permettront une gestion plus aisée des temps (notamment un coefficient de réglage du temps écoulé), des pages affichées et du Grafcet.



Un état initial peut leur être conféré qui sera modifiable et forçable pendant la phase d'exploitation des simulations.



## Les animations de synoptiques

Il s'agit dans cette phase de conception de lier aux objets graphiques de l'application, des variables qui viendront modifier leurs états. Les objets, comme décrit précédemment, possèdent leurs propres fonctionnalités et doivent être rattachés aux variables **PROSIMUL** afin d'obtenir une vue animée du process si cela est nécessaire.

L'utilisateur peut bien sûr n'exploiter qu'en partie ces possibilités ou au contraire les augmenter en superposant par exemple deux objets.

Dans le cas concret d'un objet graphique de type "CUVE ", on disposera d'une variable mot d'animation contenant sa valeur courante. Différents seuils, associés à des couleurs, permettront une visualisation pratique de la contenance.

**CUVE (p=0 n=0 g=0)**

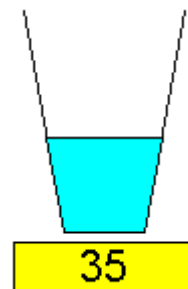
	Size	Point 1	Point 2	Intitulé objet
X	100	535	635	CUVE 4512
Y	79	126	205	Col.Objet

Style    Police

(Variables)----- Paramètres Animation ----- Col.Anim.

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     OK    Annuler



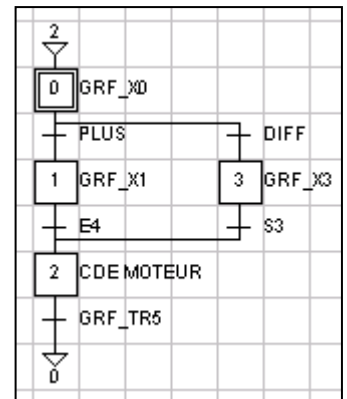
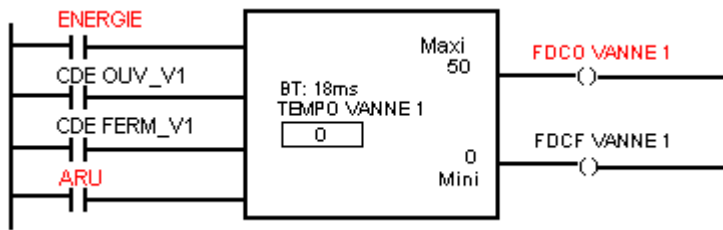
De nombreuses variations sont possibles permettant différentes visualisations graphiques du procédé en évolution tenant compte des technologies mécaniques et électriques, des flux, etc.

Une option « Dev.Only » permet en mode Formation de cacher des objets non nécessaires dans ce mode d'usage (par différence au mode développement).



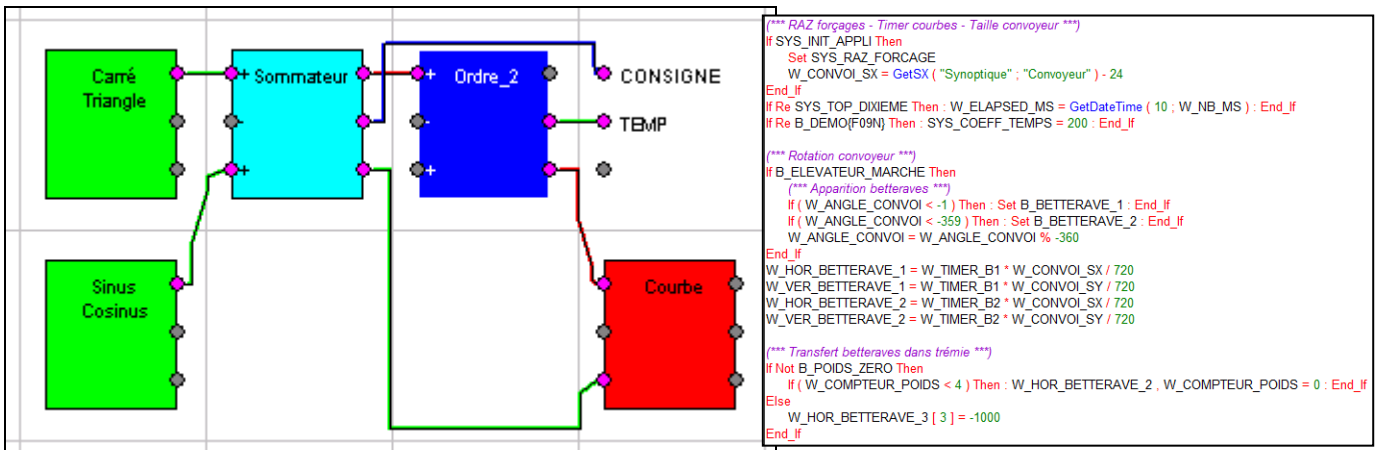
## Les éditeurs Ladder, Littéral, Grafcet et Blocs Fonctions

Il s'agit, à partir d'un jeu d'instructions sur bits ou sur mots, en **langage à contacts** simple ou en **langage littéral**, de décrire le comportement du process en dynamique. La fidélité de reproduction du système (organes mécaniques, flux de matière,...) est déterminée par l'utilisateur à ce moment de la conception. Pour exemple, dans le cas d'un vérin, le mouvement pourra être de type linéaire (position linéairement liée au temps de commande) ou scindé en mouvements linéaires successifs ou encore soumis à un fonctionnement non linéaire (exp, log, sin,...).



Un éditeur **Grafcet** (2048 étapes sur 8 pages, 32 étapes initiales) vient compléter cet outil pour tout problème séquentiel, voire pour tester une partie commande (fonctionnement sans connexion API). Un outil de vérification de structure permet un contrôle de validité des constructions.

Enfin, un jeu de **blocs fonctions** sur 4 niveaux de structuration permet de traiter les problèmes de simulation de systèmes orientés analogiques (capteurs analogiques, fonctions de transfert, régulation,...).



La connaissance la plus exacte possible du comportement de chaque élément technologique du procédé intervient largement lors cette phase de modélisation.



## La configuration des échanges

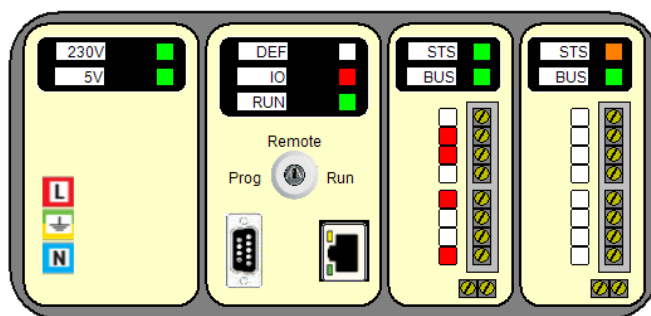
**PROSIMUL** autorise un raccordement sur l'API sous différentes formes :

- Carte parallèle TOR et/ou analogique :
  - Compex,
  - ADLINK,
  - National Instrument,
  - ...
- Liaisons sérieelles natives
  - **MODBUS** (April, Schneider, Siemens, Wago,...) – RTU – **TCP/IP** (client et serveur)
  - UNITELWAY 1 & 2 (Schneider),
  - DF1 (Rockwell SLC & PLC & ControlLogix),
  - P3964 (Siemens S5 & S7).
  - **EthernetIP** (Rockwell – client et serveur)
  - **S7-ISO on TCP** (Siemens)
- Liaison série via pilotes externes :
  - SIEMENS – Liaison S7-ONLINE (MPI, Profibus,...),
  - ROCKWELL – RSLINX
  - **Client OPC** (ROCKWELL RSLINX, SCHNEIDER OFS, SIEMENS SimaticNET, WoodHead, B&R, KEPServer, MATRIKON, ...),
  - APPLICOM – Cartes protocolées (FIP, Profibus, Ethernet,..), Direct-Link SW1000.

En liaison parallèle, une ou plusieurs cartes TOR sont implantées dans le PC et raccordées, grâce à des interfaces, aux entrées/sorties physiques de l'automate. On est alors au plus proche de la réalité, mais ceci nécessite un matériel dont la quantité croît en fonction du nombre d'E/S échangées.

Il s'agit dans ce cas de configurer les cartes du PC en fonction des cartes E/S API raccordées et d'indiquer leur rangement dans les tables d'entrées/sorties **PROSIMUL**.

Dans le cas d'une liaison série, il s'agit d'indiquer les tables des entrées/sorties de l'automate et leurs longueurs. Ce mode d'échange implique parfois quelques lignes de programme au niveau de l'API. L'inconvénient dans le cas d'échanges par voie série est que l'on accède aux bits images et non aux entrées / sorties physiques de l'API (perte des voyants des entrées).



Les liaisons parallèles (TOR et analogiques) et série peuvent co-exister en simultanément pour permettre une petite quantité d'échange très rapide et des volumes plus importants via liaison série.

Dans tous les cas il est possible de supprimer tout échange avec l'extérieur afin de vérifier le comportement du procédé de manière autonome.

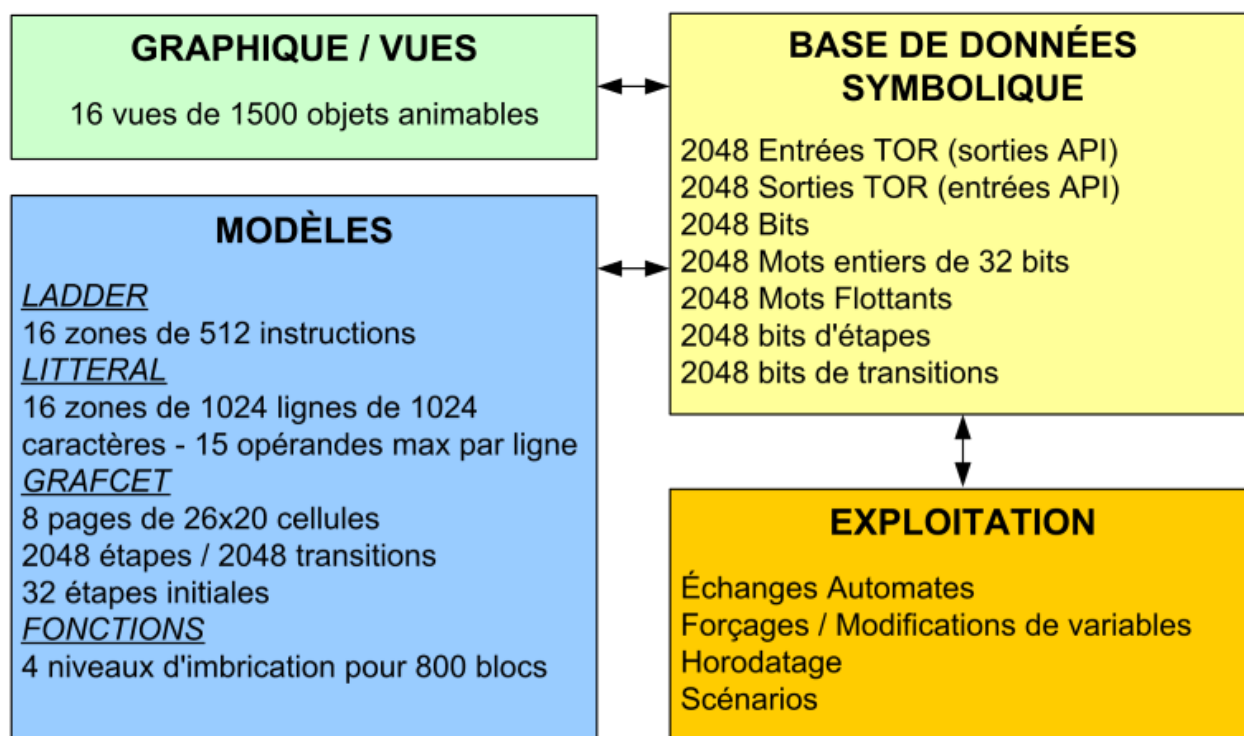




## L'exploitation de la simulation

Lorsque l'application est validée en fonctionnement interne, la connexion de l'API peut être réalisée. **PROSIMUL** effectue alors un cycle de lecture des entrées suivant le mode d'échange retenu, exécution des programmes « modèles », suivie de la mise à jour des sorties. Les animations d'écrans sont effectuées parallèlement.

Toutes les variables **PROSIMUL** peuvent être modifiées ou forcées afin de provoquer des dysfonctionnements de la partie opérative et tester le comportement du programme API face à ces défaillances (gestions de défauts, communication,...). Tous les modes de marche et d'arrêt peuvent aussi être reproduits afin d'assurer une simulation aussi complète que possible (cahier de recettes d'installation). Les étapes actives de l'application ainsi que les valeurs issues des blocs fonction peuvent être visualisées et modifiées en dynamique.



Des scénarios de tests (actions manuelles et/ou automatiques) peuvent être enregistrés et reproduits, afin de limiter les manipulations au maximum. Ces enregistrements de variables horodatées peuvent être analysés à travers un outil spécifique détaillé plus loin. Des états ou alarmes peuvent être transmis par emails.

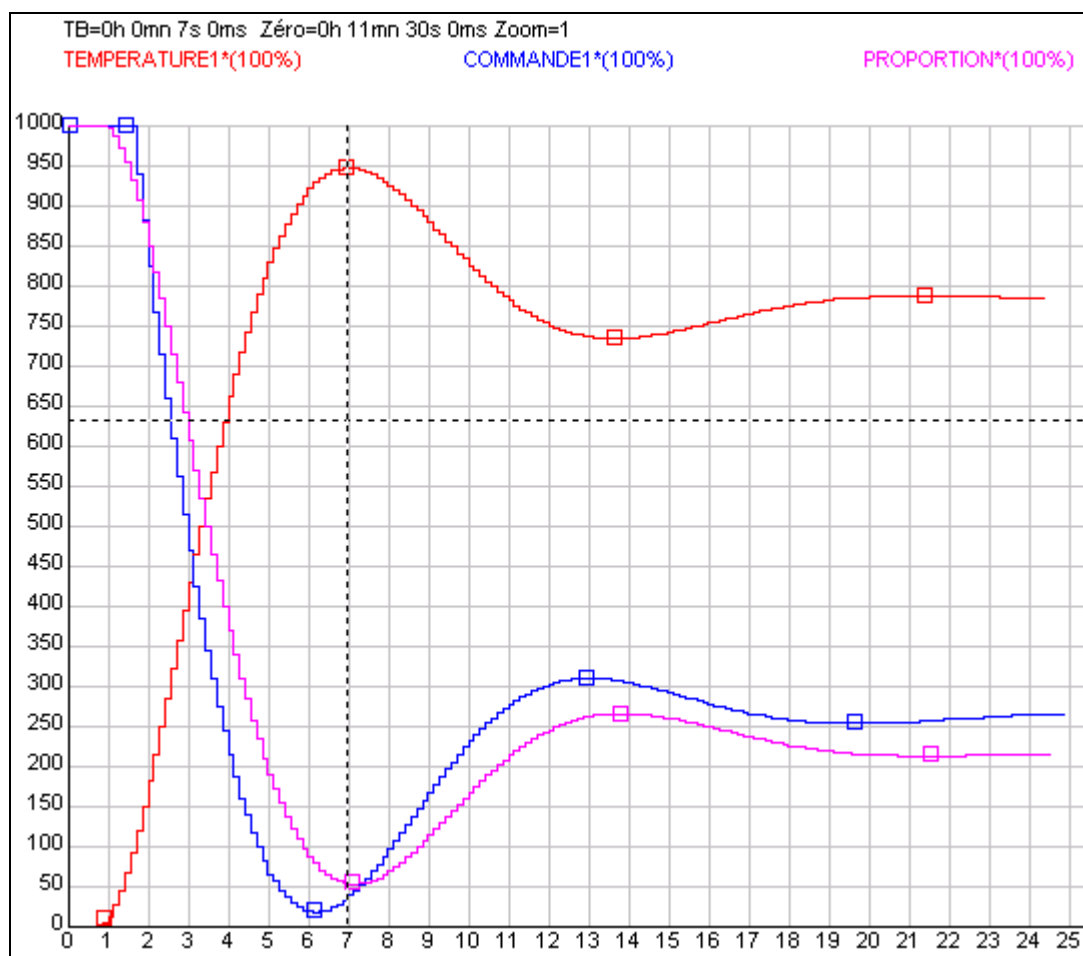
Les essais peuvent s'exécuter en temps réel ou fictif (coefficient de réglage). Ainsi des séquences de tests peuvent être ralenties afin d'examiner certains points en détail, ou à l'inverse accélérées pour réduire les temps de simulation



## La fonction Histogramme.

Afin d'analyser le comportement du procédé dans le temps, dans tous les modes de marche et d'arrêt, la fonction histogramme vous permet d'enregistrer des valeurs (TOR, numériques et analogiques) de l'application à chaque changement d'état (précision d'une milli-seconde).

L'exploitation de ces enregistrements se fait soit via export vers tableurs ou directement sous forme graphique. Les dysfonctionnements sont donc facilement détectables, les temps de commande, et / ou de retour d'informations peuvent être contrôlés visuellement, ainsi que les concordances d'événements.



Des fonctions de Zoom sur temps, détection de points d'extrêmes et d'inflexions sont disponibles pour compléter l'analyse des données enregistrées.

Dans le cas de modélisation de procédé continu, un outil de recherche de polynôme (degré 2 à 12) autorise l'identification d'une série de valeurs. Ce résultat pourra être introduit en bloc fonction afin d'effectuer des tests de régulation PID par exemple.



## MÉTHODOLOGIE DE CONCEPTION

Toute bonne simulation passe par la connaissance la plus exacte des constituants du process, ainsi que leurs modèles de comportement. Il est important pour le concepteur de réaliser que la simulation consiste à modéliser le procédé réel (réaliste), avec toutes ses contraintes, et non le procédé idéal.

La première opération à effectuer consiste à recenser tous les organes connectés à l'automate (étape aisée si l'affectation des E/S existe) et d'en identifier la technologie. Puis, le niveau de vraisemblance du procédé (conditions logiques de fonctionnement des constituants, énergies, aspects mécaniques), ainsi que les représentations graphiques qu'on souhaite en avoir (capteurs, actionneurs, pupitre de contrôle / commande,...) devront être spécifiés.

L'analyse la plus délicate porte sur les flux et transformation de matière. Il est évident qu'un apport de produits en quantité importante, voire « infini » dans le temps n'est pas toujours simulable directement (ex : flux continu de pièces sur un convoyeur). L'utilisateur doit concevoir son modèle en tenant compte des conditions réelles d'introduction, d'avance, et de transformation de la matière, tout en tenant compte que cette logique puisse s'appliquer en continu.

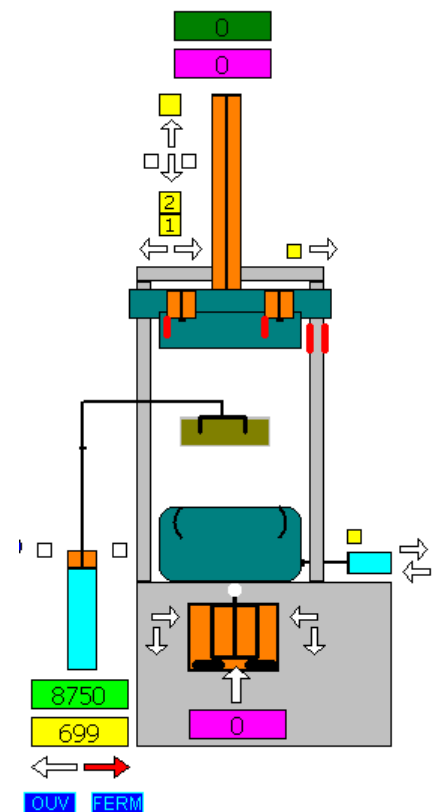
Dans certains cas complexes, on pourra limiter les quantités de produits, voire décrire tout simplement un comportement fonctionnel satisfaisant du point de vue de la partie commande.

Lorsque cette analyse du process est faite, le concepteur peut construire ses représentations graphiques (tenant compte des animations définies) ainsi que les points d'interactions qu'il souhaite avoir sur l'application.

La génération de la base de données se fait tout naturellement en fonction des E/S API, qui peuvent être récupérées depuis un fichier texte, et des variables internes au procédé (temporisations, compteurs, mémoires,...).

La partie liée à la modélisation des comportements du système nécessite des connaissances pluri-technologiques. Les quatre langages à disposition offre à l'utilisation une variété de solution, et l'application peut être testée à tout moment, afin d'éliminer les problèmes progressivement.

Dans le cas de process contenant de multiples objets identiques, l'objet référent doit être testé complètement avant d'être mis en bibliothèque et dupliqué.





## CONCLUSIONS

La réussite d'un projet de simulation est donc avant tout soumise à la connaissance correcte du process et de ses constituants, mais aussi à une capacité d'analyse fonctionnelle du comportement de l'installation.

Cet investissement n'apparaît pas toujours évident. Les méthodes traditionnelles (boîtes à boutons, voyants, tableurs) sont en effet d'une mise en œuvre relativement simple et économique. Toutefois elles ne permettent que des tests simples et peu dynamiques de l'installation et sont sujettes aux erreurs de manipulation.

Il faut donc chercher l'avantage d'un système de simulation tel que **PROSIMUL** à différents niveaux. Notre expérience nous conduit à penser qu'en moyenne une journée/homme est nécessaire pour finaliser une simulation de 64 E/S. Ceci est à mettre en rapport avec les temps d'essais hors site et sur site habituels, ainsi que le temps de prise en main de l'application par le personnel de maintenance et d'exploitation. Les consommations énergétiques, les pertes de matière et les incidents évités interviennent aussi dans le bilan final d'un projet.

L'application développée initialement pour l'étude offre l'avantage d'être un outil de présentation et de validation hors site mais peut aussi être fort intéressante pour les utilisateurs finaux. Elle constitue une base de connaissance des systèmes et offre des capacités de dialogue entre concepteurs et exploitants.

Les gains en sécurité, qualité et disponibilité de l'équipement dépendent pour une bonne part de cette capacité à observer et connaître le process pour intervenir dessus efficacement. La simulation contient certainement en partie la réponse à ces impératifs.

