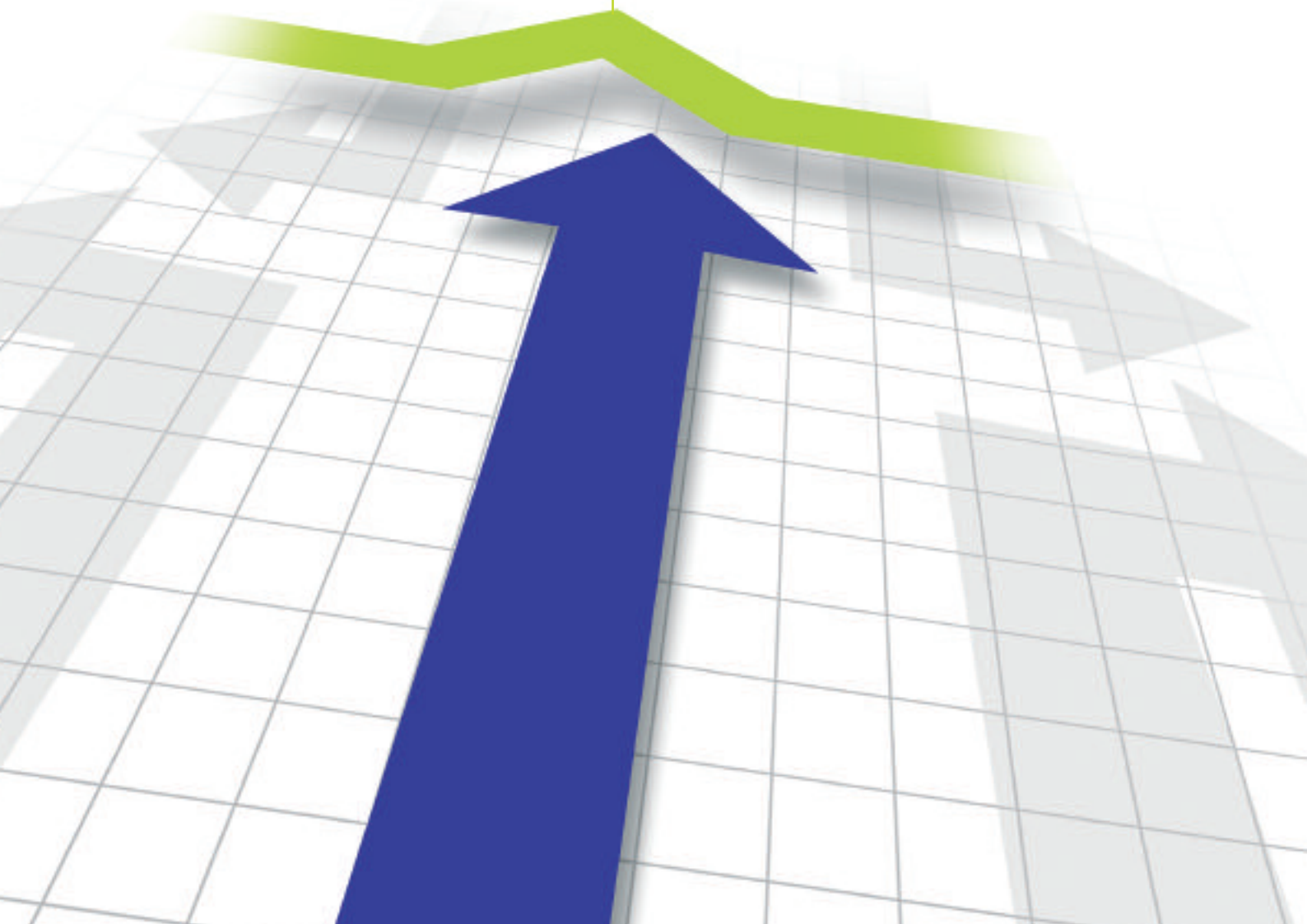


10

chapter

Data treatment and software

- *Presentation of actual architecture examples (schematics, products and softwares)*
- *Presentation of an application generator in a collaborative environment*



10.1	Définition	234
10.2	Introduction	234
10.3	Programming, configuration and languages	235
10.4	Application categories	236
10.5	UAG: Application generators	250
10.6	Definition of the main abbreviations used	254

1

2

3

4

5

6

7

8

9

10

11

12

M

10. Data treatment and software

- 10.1 Definition
- 10.2 Introduction

This section deals with the processing function discussed in the first section and includes a description of industrial software and its interaction with automation system processes. Unlike other sections, we shall not go into details about concepts such as systems, programming languages, etc. Many publications are available to the readers.

10.1 Definition

Programmable Logic Controller (PLC) is the name used for a programmable electronic device for controlling industrial systems by sequential processing.

It sends operators (Operating Section or OS) commands based on input data (sensors), setpoints and a program.

A PLC is a device similar to a computer and is used for instance to control machines on an industrial assembly line. A single PLC is enough to do what older automation systems did with hundreds or thousands of relays and cams. The people who program PLC's are called automated systems engineers.

10.2 Introduction

Programmable Logic Controllers (PLC's) were first developed in the 1970s. They were initially designed to deal with the sequential logic required to run machines and processes. At first with, their cost confined them to large systems. Major technological developments have radically restructured the processing function:

- the languages have been unified and standardised; the IEC 61131-3 standard defines the different types;
- the system approach is now widespread and the diagram principle has been superseded by function blocks;
- digital systems are now widely used to process digital values as well as analogue values with prior analogue-to-digital conversion;
- the cost of electronic components has dropped so much that PLC's can now be used instead of relays even in simple systems;
- the communication buses used for data exchange are a competitive alternative to conventional wiring;
- the software technology used in offices and business are increasingly used in industrial automation systems;
- human-machine interfaces have also progressed in becoming programmable for greater flexibility.

10.3 Programming, configuration and languages

An automation program consists of a set of instructions to be run in a specific order by a processor. The word program is often used as a synonym for software. Though software does largely consist of programs, it often requires resource files containing all sorts of data which are not part of the program.

This is where configuration comes in. Configuration is not programming, it completes software by giving it the data it requires to run properly.

As an example, a water treatment system can range from very basic to highly complex, starting with a simple program to maintain the level of water in a reservoir between two limits by opening and closing an electric valve.

A slightly more complex arrangement could involve a flow controller (incoming) and a flow controller (outgoing) to keep the water flowing at a set rate. An industrial application, such as wastewater treatment, will control several reservoirs. Each reservoir must meet a set of conditions, such as:

- level between the minimum and maximum limits;
- pH factor within a given range;
- have a certain output rate, etc.

■ Standard languages

The International Electrotechnical Commission (IEC) has developed the standard IEC 61131 for Programmable Logic Controllers. Part 3 of this standard (IEC 61131-3) defines the programming languages:

- **IL** (Instruction List) is very similar to assembler language, working in close touch with the processor by using the arithmetical and logical unit, its registers and accumulators;
- **ST** (Structured Text) is similar to C language used in computing;
- **LD** (Ladder Diagram) resembles electrical diagrams and can quickly convert an old electromechanical relay program. This way of programming gives a visual approach to problems;
- **FBD** (Function Block Diagram) is a suite of blocks which can be linked together and perform any type of function from the simplest to the most advanced;
- **GRAFCET** (acronym for “GRAphe Fonctionnel de Commande Etapes/Transitions” or Step/Transition Control Function Chart) is an automation system representation and analysis mode particularly well adapted to sequential systems because it can be broken down into steps. In PLC programming, **SFC** can be used in a very similar way to G (Grafcet IEC848 became an international standard in 1988 with the name of “Sequential Function Chart” (SFC)). Behind each action there is an associated program written in IL, ST, LD or FBD.

10.4 Application categories

Technological progress impelled by user requirements has given rise to a wide range of PLCs which can feature:

- hardware such as processing power, the number and characteristics of inputs/outputs, execution speed, special modules (axis control, communication, etc.);
- software which, apart from the programming language, has higher functions and capacities for communication and interaction with other business software.

These will be described through typical applications to help direct the reader's choice. Our advice is then to refer to the individual documentation of each product.

In the introduction to this guide, we looked at the principle of automation system architecture and implementation based on the customer profile. The solutions described can be divided into four categories.

A - « Electrician » solutions

Applications are simple, standalone and fixed. The choice criteria should be based on products that are easy to use, inexpensive and undemanding in maintenance.

B - « Automated/mechanical systems engineer » solutions

Applications are demanding with regard to mechanical performance (precision, rapidity, movement control, range changes, etc.). Their architecture and processing systems will largely be chosen for performance.

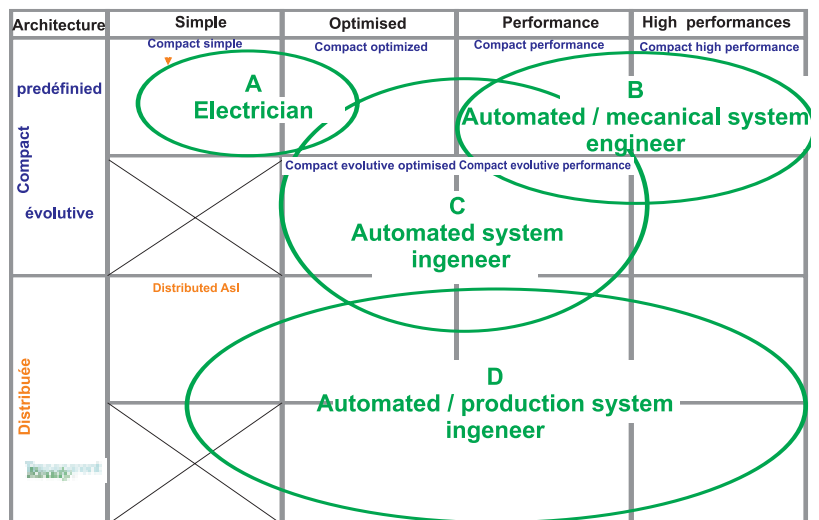
C - « Automated systems engineer » solutions

Automated systems are made complex by the volume and variety of the information to process such as adjustment, interconnections between PLCs, number of inputs/outputs, etc.

D - « Automated/production systems engineer » solutions

Automated production systems must be integrated into the plant's management system architecture. They must interface with each other and exchange data with production and management software, etc.

In *figure 1*, these categories are positioned over the implementations described in section 1 of the Guide to Industrial Automation Systems.



↑ Fig. 1 Applications categories and users profiles

■ “Electrician” solutions




Simple solutions use a few electromechanical relays to run automation sequences. The latest generation of small PLCs are easy to use, competitively priced starting from a few inputs/outputs and offer new capacities without requiring any expertise in programming.

Typical applications are in the following sectors:

- industry: simple machines and additional functions in decentralised systems;
- buildings and services: lighting management, access, control, premises surveillance, heating, ventilation, air-conditioning.

□ Application with a Zelio PLC

This configuration is suited to the following applications *figure 2*.

Application	Description	Example
Mobile pumping station	An application to fill and empty tanks. Use of a speed controller helps to adjust to degrees of viscosity in fluids.	
Automatic gate	To open and close factory gates.	
Electric window	To control the air in a garden centre.	

↑ Fig. 2 Examples with Zelio PLC



↑ Fig. 3 Application based on Zelio PLC

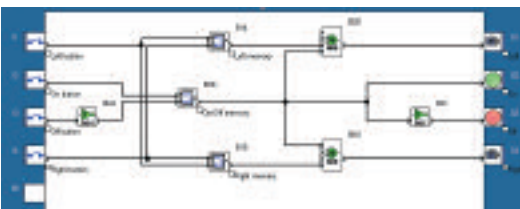
In the diagram (⇒ Fig. 3), motor operation is governed by a speed controller. For discrete control, all it requires is a contactor linked to its thermo-relay.

This unit comprises:

- a Zelio Logic PLC;
- 24V DC Phaseo power supply;
- an Altivar 11 speed controller;
- a GV2 motor circuit breaker;
- an XVB light tower;
- a Vario VCF switch.

The variables of the speed controller (time, speed, control) can be set directly on the Altivar 11 or with Powersuite software.




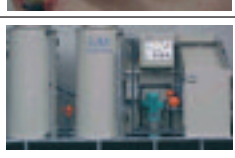
The Zelio can be programmed directly on the module or with Zeliosoft software installed on a PC. The latter option is illustrated by the screenshot in *figure 4* which shows a logical process run by FBD (Function Block Diagram).



↑ Fig. 4 Screen shot of a logical FBD

□ Application with a Twido PLC

This type of PLC is used to build simple applications which can be monitored or controlled remotely via a modem connected to the telephone network (PSTN). *Figure 5* present some examples:

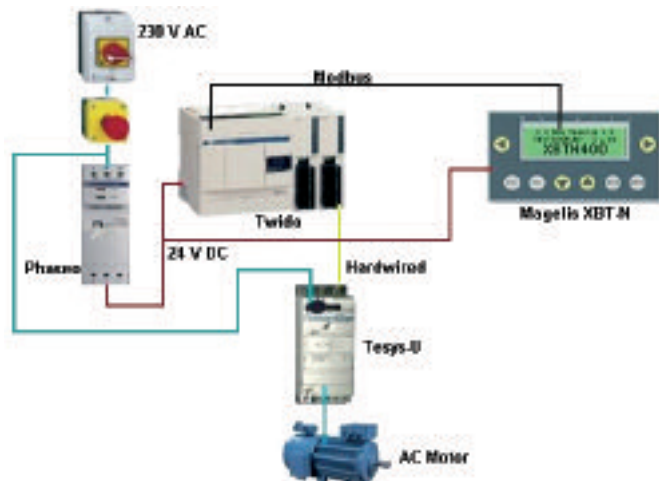
Application	Description	Example
Ventilation	Control of a ventilation system in an industrial building. Temperature measurement governs the starting and stopping of the system.	
Heating	Heating system control in a building.	
Remote control of a fountain	Control of a fountain infrastructure of a service company. The system is remote controlled via a modem.	
Control of filter cleaning in a water distribution plant.	The application controls and cleans the filter in a water distribution plant with an air-cleaning sequence followed by clean water. The system can also be remotely controlled via a modem.	

↑ *Fig. 5* Examples with Twido PLC

□ Typical diagram

The system is developed from a Twido PLC (⇒ *Fig. 6*) and controlled and viewed via a Magelis keyboard/screen. Security is ensured by an emergency stop on the main switch.

The system is hardwired and the PLC controls the starter and the messages from the alarm module.



↑ *Fig. 6* Application based on Twido PLC

The following components make up the system:

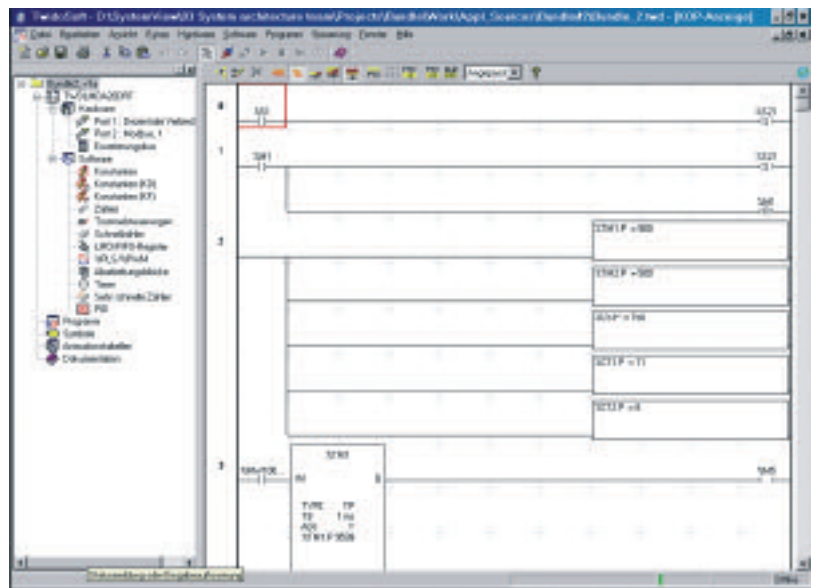
Hardware:

- Twido Modular (PLC);
- Phaseo power supply (PS);
- TeSys-U Starter Controller (SC);
- Magelis XBT-N (HMI);
- Standard 3-phase motor.

Software:

- Twidosoft Version 2.0;
- Magelis XBTL1003M V4.2.

The screenshot in *figure 7* of the Twidosoft program illustrates programming in Ladder which can be switched to List. The software includes a large set of instructions and an embedded browser is used to access all the objects directly.



↑ Fig. 7 Screen shot of Twido Ladder program

■ “Automated/mechanical systems engineer” solution

Some applications require performances that are difficult to achieve without their being closely linked to processing, acquisition and power control functions. For this reason, linkage is directly processed by the power control function, either in analogue, by fast bus (CANopen, etc.) or special bus (Sercos, etc.). This type of architecture is found in speed controllers for both asynchronous and synchronous motors (close loop systems).

The need to link a sequential process of analogue and discrete variables means that manufacturers have had to add to speed controllers:

- input/output cards;
- communication cards;
- PLC processing cards.

These solutions can be used equally for industrial applications and in infrastructure.

We will illustrate this solution by a speed controller with a built-in PLC.

10. Data treatment and software

10.4 Application categories



↑ Fig. 8 Altivar 71 with Controller Inside card

□ Processing built into a speed controller

A programmable card option (⇒ Fig. 8) can be used to convert Altivar speed controllers into proper automation production cells. This card, called "Controller Inside" has all the PLC functions built into it:

- the Codesys programming software with standardised IEC 61131-3 graphic language capacity;
- processing close to the motor controls for rapidity of movement;
- master CANopen bus to govern other speed controllers (Altivar 31, Altivar 61 and Altivar 71) and exchange all requisite data;
- the card has its own inputs/outputs and access to Altivar I/Os;
- graphic terminal display functions are stored to inform and configure;
- supervision available via Ethernet, Modbus or other communication networks.

CoDeSys is a programming tool used with Windows. It supports the five standard IEC 61131-3 languages. CoDeSys produces a native code for most current processors and can be used on different controller platforms.

It combines the power of advanced programming languages like C and Pascal and PLC programming system functions. The programming kit includes a manual and online help and is available in English, French and German.

Many manufacturers use it and Schneider Electric has chosen it for its Altivar and Lexium speed controllers.

□ Applications

This configuration is suited to the following applications (⇒ Fig 9).

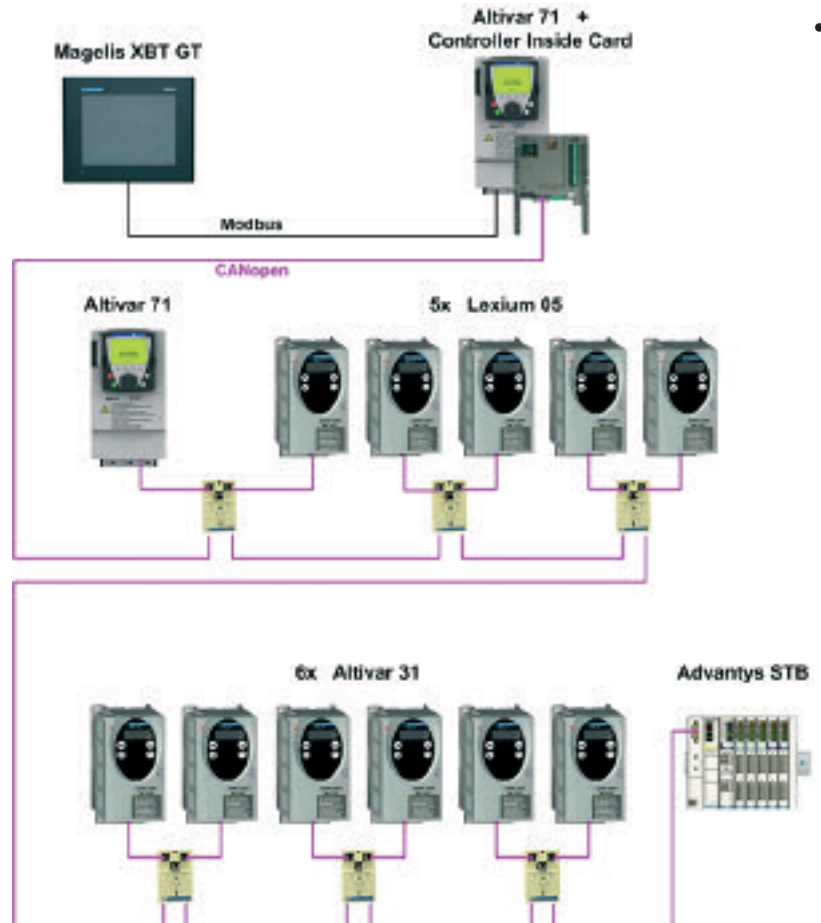
Application	Description	Exemple
Infrastructure network	Used in pumping station to feed user's with fresh water	
Dedicated machines	Several applications: - winders - automatic assembly machines - wood working machines.	
Conveyors	Used in combination with hosting equipment's and shuttles.	

↑ Fig. 9 Applications for PLC embedded in the Speed drive

□ Typical diagram

To make the illustration of this solution clearer, the power section and its supply are not shown *figure 10*.

In the diagrammatic illustration, the system comprises:



↑ Fig. 10 Application with a ATV71 + Controller Inside

Hardware

- Controller Inside card installed in an Altivar 71; the speed controller is the master on a CANopen bus;
- ATV31 and ATV71 speed controllers with built-in CANopen interface;
- Lexium05 servo-drive with built-in CANopen interface. The HMI is managed by a Magelis XBT-GT graphic terminal and linked to the production cell by a Modbus link;
- Advantys STB distributed input/output production cells.

• Software

- PS1131 (CoDeSys V2.3);
- PowerSuite for ATV31, ATV71 and Lexium05;
- Vijeo-Designer V4.30 for Magelis;
- Advantys Configuration Tool V2.0.

■ “Automated systems engineer” solution

Modern automation systems require a great many inputs and outputs of different types. They must be able to process automation sequences and provide the information needed for management and maintenance.




The complexity of the systems entails lengthy and costly design and implementation. Manufacturers offer a dual approach to help cut costs:

- modular offers of hardware and software to reduce the end cost of learning all the applications;
- software tools to boost productivity, manage logs and ease system maintenance and upgrading.

10. Data treatment and software

10.4 Application categories

This type of automation solution is used in industrial processes where several machines are linked together or in infrastructures. Some examples are shown in the table *figure 11*.

Application	Description	Example
Handling equipment.	Used in a process with several conveyor systems and which uses external information.	
Packaging machines, textile machines, special machines.	Used for cutting and folding machines integrated into a production line.	
Pumps and fans.	Used for water circulation and refrigeration systems governed by external measurements such as output rates.	

↑ *Fig. 11* Industrial process applications

□ Applications

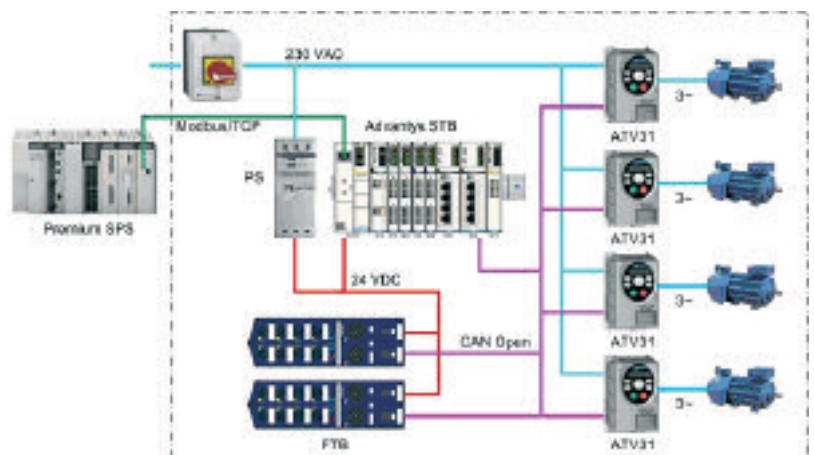
We do not intend to describe an application from end to end, but to illustrate its working principle we will make a description of a significant part of it.

A Premium PLC is used to control a local production cell (⇒ *Fig. 12*). This is a platform made up of Advantys STB inputs/outputs, four speed controllers and external input/output modules. Each element is connected to a CAN bus. This implementation can easily be expanded by adding more speed controllers and inputs/outputs. The PLC is linked to the production cell by a Modbus/TCP bus. The controllers and motors are powered from a 230VAC network. Another source is used for a 24VDC supply.

□ Typical diagram (⇒ *Fig. 12*)

List of system components:

- TSX Premium (PLC),
- Phaseo (24V supply),
- ATV31 (speed controllers),
- Advantys STB (input/output cell),
- Advantys FTB IP67 input/output module,



↑ *Fig. 12* Application based on PLC Premium

- 3-phase squirrel cage motors.

Software:

- Unity Pro V2.0.2 (PLC),
- Advantys configuration software V1.20 (I/O cell),
- PowerSuite V2.0 (ATV31 speed controller configuration).

□ UNITY PRO software workshop

Unity Pro is the common programming software for debugging and operating Modicon Premium, Atrium and Quantum PLCs. Based on the standards of PL7 and Concept, Unity Pro opens the way to a comprehensive set of new functions (⇒ Fig. 13) for a greater productivity and software collaboration.



↑ Fig. 13 Screen shot of Unity Pro

• Main features of Unity Pro

- Windows 2000/XP graphic interfaces;
- “Custom” icons and toolbars;
- user profiles;
- graphic design of hardware configurations;
- integrated PL7 and Concept converters;
- automatic generation of synchronisation variables on Ethernet (Global Data);
- 5 native IEC61131-3 languages supported with graphic editors;
- integration and synchronisation of program editors, data, user function blocks;
- drag & drop reuse of library objects;
- XML import/export and automatic data reassignment;
- automation of repetitive tasks by VBA macros;
- plug & play Hot Standby redundancy system.

Unity Pro offers a comprehensive set of functions and tools to match the structure of the application to the structure of the process or the machine. The program is divided into functional modules which, assembled with hierarchical priority, form the functional view and contain:

- program sections;
- animation tables;
- operator screens;
- hyperlinks.

The basic functions, used repetitively, can be encapsulated into user function blocks (DFB) in IEC61131-3 language.

To create an application reference database, Unity Pro supports project and application libraries locally or on server.

It has around 800 standard function, and can be enhanced with customer’s standards (variables, data types, function blocks).

It also includes:

- symbolic variables independent of the physical memory;
- structured user-defined data types (DDT);
- DDT and DFB function block version management in the library;
- a library of pre-animated graphic objects for operator screens;
- read/write protection of programming data to prevent any modification;
- function block development available in C++ with the Unity EFB Toolkit option.

Put in libraries on a local PC or a remote server, the application objects and their properties are used and shared by all programs and any changes made to an object in a library is effective in the programs that use it:

- the functional modules can be used in the application or between projects by XML import/export;
- function blocks are instantiated by drag & drop from the library;
- all instances automatically inherit library changes (as the user chooses);
- graphic objects for operator screens are selected and configured by drag & drop.

A PLC simulator on PC is integrated into Unity Pro and is used to fine-tune the application as much as possible before it is commissioned on site. It exactly reproduces the behaviour of the target program. All the debugging tools can be used in simulation:

- step by step program execution;
- break and view point;
- dynamic animations to view the status of variables and the logic in execution.

Operating screens facilitate debugging by views representing variable status in graphic object form: indicators, trend curves, etc.

The same as for configuration, application-specific modules have special screens to debug them: the functions available are adapted to the type of module implemented (discrete, analogue, counter, communication I/Os, etc.).

Operator actions are logged and archived in a standard secured Windows file. Hypertext links are used to link the application to all the documents and tools required for operation and maintenance.

• Diagnostics tools

Unity Pro provides a library of application diagnostic DFBs. These are integrated into the program and, depending on their function, are used to monitor the permanent safety conditions and the progress of the process.

A viewing window displays any system and application defects explicitly and chronologically in real time from the source. A click on the window opens the editor of the program where the error was triggered (search for conditions missing at the source).

Online changes can be grouped consistently in local mode on a PC and transferred directly to the PLC in one operation to be included in the same cycle run.

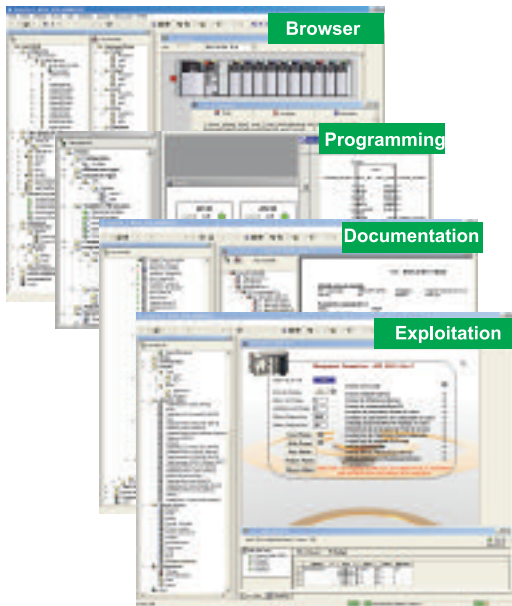
Hypertext links integrated into the application give remote or local access to working resources (documentation, additional tools, etc.) to cut stopping time.

There is a full range of functions to control operations:

- unity Pro operator action log in a secured file;
- user profile with a choice of accessible functions and password protection.

10. Data treatment and software

10.4 Application categories



↑ Fig. 14 Screen shot of various Unity Pro vues

Operating screens and functional views for direct graphic access to application elements (⇒ Fig. 14).

The Unity client/server architecture gives access to the software resources via programming interfaces in VBA, VB or C++; here are two examples:

- automation of repetitive tasks (input, configuration, translation, etc.);
- integration of specific applications (code generator, etc.).

• **Cross-software exchange**

The XML format, the universal W3C standard for data exchange via the internet, is used as the source format for Unity applications such as variables, programs, inputs/outputs, configurations, etc. (⇒ Fig. 15).

Simple import/export is used to exchange all or part of the application with other software in the project (CAD, etc.).

Unity Developer's Edition (UDE) and its programming interfaces in C++, Visual Basic and VBA can be used to develop custom solutions such as interfaces with electrical CAD, a variables generator, a PLC program or repetitive design task automation. Many software publishers use UDE to simplify data exchange with Unity Pro; here are some examples (⇒ Fig. 16).



↑ Fig. 15 Screen shot of steps for cross software exchange

Domain	Company	Product
Electrical CAD	ECT	Promise
Electrical CAD	EPLAN	EPLAN
Electrical CAD	IGE-XAO	SEE Electrical Expert
Electrical CAD	AutoDesk	AutoCAD Electrical
Electrical CAD	SDProget	SPAC Automazione
Process Simulation	Mynah	Mimic
Change Management	MDT Software	AutoSave
Application Generator	TNI	Control Build
SCADA/Reporting	Iconics	GENESIS BizViz Suite
SCADA	EuropSupervision	Panorama
SCADA	Arc Informatique	PCVue32
Graphical User Interface	ErgoTech	ErgoVU
SCADA	Areal	Topkapi
SCADA	Afcon	P-CIM
MES	Tecnomatix/UGS	XFactory
Historian/RtPM	OSISoft	PI
Web Services	Anyware	PLC Animator

↑ Fig. 16 Software publisher using UDE

- **Compatibility with existing applications**

PL7 and Concept IEC61131 applications are imported into Unity Pro by an integrated converter as a standard feature. Operating system update provided with Unity Pro is available for most Premium and Quantum PLC processors.

Existing I/O peripherals, application-specific, communication and field bus modules remain compatible with Unity Pro.

- **“Automated/production systems engineer” solutions**

Here we discuss complex architectures implementing several PLCs which communicate with each other and with supervisory workstations (SCADA). These architectures also interface with corporate management systems.

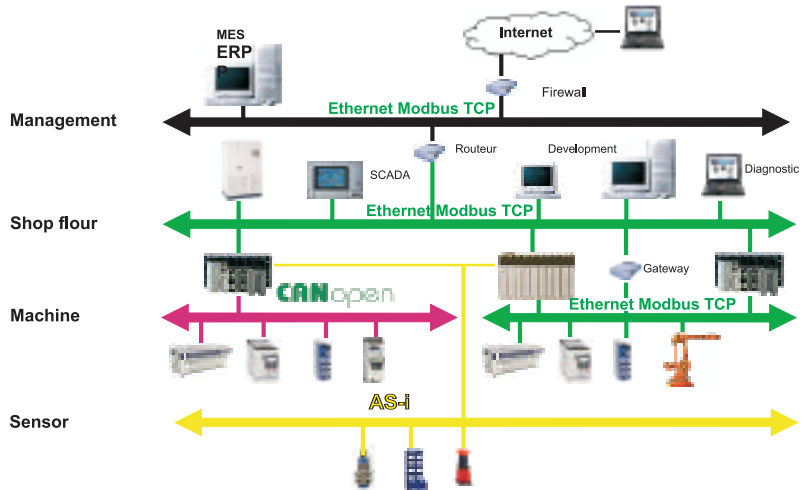
System optimisation requires a global approach to interrelate all the business departments, partners and means. We can distinguish two types of interaction:

- real time: in the operating stage, this characterises the link with customer relationship management systems (CRM), stock and production management (MES) and accounts management for optimising flows;
- collaborative: software tools support the relationship between the relevant players in the design, building, operating and maintenance stages to cut the time and cost of the production tool and improve its quality.

Applications are so diversified that it is not easy to grasp their individual positions in their environment. The need for cross-software exchange leads to a collaborative approach throughout the business.

- **Real time and corporate software**

The illustration *figure 17* shows the most common software used in a workshop or factory. It can be separated into four levels:



↑ Fig. 17 Software used in a factory

- the **corporate level** is characterised by a very large flow of information. Office automation and internet standards are now basic requirements. The software is run on PC or on more powerful servers;
- the **shop floor level** is mainly characterised by supervisory tools and the PLCs which govern the process. Ethernet is now the standard means of communication between the computer and PLC domains;
- the **machine level** where the principle of real time conditions the choice of communication tools. Tasks are allotted to industrial PC's and PLC's, links are made by field buses (CANOpen for Schneider Electric) or by Ethernet associated to specific application layers;
- the **sensor level** where hardwired links compete with the ASi bus which is particularly well suited to this kind of use.

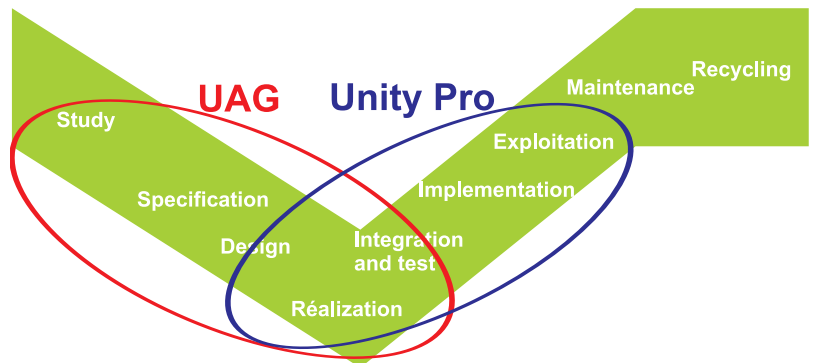
□ Collaborative design and building

The PLC software described above is confined to the control of assemblies designed separately to work as standalone units, even though communication links are sometimes required.

New tools have been developed around the programming software to deal with the constraints of working in parallel on design and building. These help developers to simplify and speed up their work, keep track and reduce the number of errors.

Consider the cycle from the study stage to the end of the life of a machine or process (⇒ Fig. 18). The range of different types of work involved in this lifecycle requires collaboration between departments. This collaboration is made possible by tools such as mechanical and electrical CAD, ERP, MES, SCADA and others.

Unity Pro and UAG (application generator) at the core of the Schneider Electric offer provide a software and hardware automation platform based on openness and collaboration between the tools which cover the range of needs from the study stage right up to system maintenance (⇒ Fig. 18).



↑ Fig. 18 The two platforms UnityPro and UAG

The Unity Pro software workshop has already been described so we will devote the next paragraph to UAG and how it works.

Several additional software tools have been developed to ensure collaboration. The table *figure 19* describing their functions is followed by a brief explanation of how they work.

The main computing standards such as Ethernet TCP/IP, Web, OPC, SOAP, XML, etc. are used to facilitate vertical collaboration at every corporate level and offer:

- more visible information in real time;
- interoperability between the process and the information systems (MES, ERP, etc.);
- exchange with design tools such as CAD.

Components	Function
Unity Pro	Single PLC application development workshop in a collaborative environment
UAG Unity Application Generator	Object-oriented multiple PLC application development and SCADA Complies with standard ISA S88
UDE Unity Developer Edition	Software for development in VBA, VB and C++ programming languages
OFS OPC Factory System	Schneider OPC server to interrelate the desktop and PLC environments
Factory Cast Web Environnement	Ensure that information passes between a PLC environment and a desktop environment
CITEC SCADA	SCADA software
AMPLA	MES software
Ethernet card	Cards using Factory Cast services
Organisation of additional configuration and setting software	
XBT L1000	HMI creation
Vijeo designer	HMI creation
Vijeo Look	Mini SCADA
Power Suite	Configuration of Altivar speed controllers and Altistart and Tesys U starter units

↑ *Fig. 19* Complementary software tools

UDE: Unity Developer's Edition

The Unity range is enhanced with Unity Developer's Edition (UDE), specialist software for programmers in VBA, VB or C++. It provides access to all the object servers in Unity Pro software for developing custom solutions such as interfaces with an electrical CAD or an automatic application generator.

UAG: Unity Application Generator

UAG is a design tool based on a reusable module approach (PID, valve, motor, etc.) and is compliant with standard ISA 88. UAG generates the code of the PLCs in the architecture and the SCADA graphics. In addition, with a single input it manages a database, common to all the elements, to ensure application consistency.

Single data input ensures speed and consistency between the two environments.

With this structured modular design approach, UAG offers significant savings in development costs and facilitates validation and maintenance of automation projects.

OFS: OPC Factory System

OFS, OPC (OLE for Process Control) adapted to the Schneider Electric environment, is a program for communication between the desktop environment and the industrial automation systems. It originated with Microsoft and derives from DDE, then OLE to OLE Automation using Windows COM/DCOM.

A foundation made up of software providers and publishers manages OPC upgrades and guarantees upward compatibility and interoperability between different software products.

Upgrades are conditioned by the following industrial requirements:

- application interfaces based on open standards offering simple common access to shop floor data;
- greater interoperability between automation and control applications, site equipment and IT applications;
- multiple Client/Server architecture;
- access to a local or remote server;
- information flow in real time.

Factory Cast: Web environment

A set of tools to enable applications to communicate via internet and meeting the following requirements:

- communication between applications;
- web and internet compatibility;
- standard-based non-proprietary solutions;
- easier implementation.

The applications must be able to communicate whatever:

- language they were developed in;
- operating system they run on.

The internet-compatible technology is based on a standard XML SOAP protocol (*Simple Object Access Protocol*) cohabiting with HTTP and enables applications to communicate with each other.

A standard description of services and interfaces is provided by a WSDL application (*Web Service Description Language*) which is a library of standard and self-documented data access functions.

The combination of these technologies is known as 'WEB SERVICES' and is independent of platforms, languages and operating systems.

SCADA: supervision software

SCADA (Supervisory Control And Data Acquisition) is industrial software designed to optimise production management. It is used to run a production workshop in real time, based on production requirements and the means available.

Ethernet cards

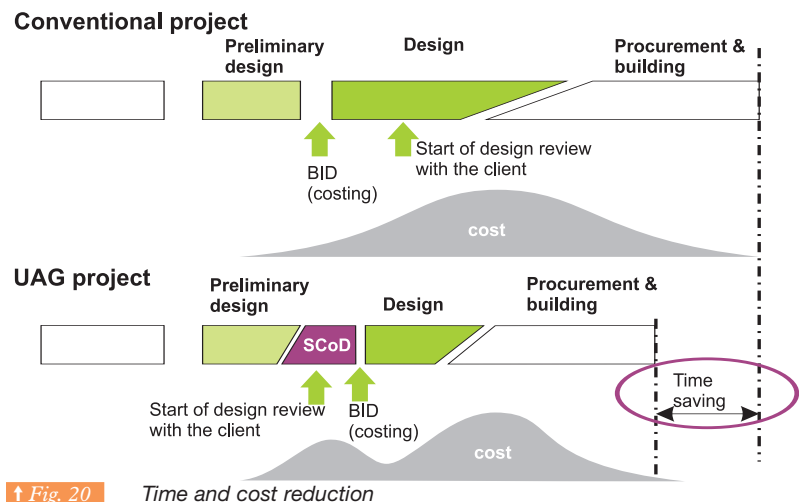
The ranges of Ethernet cards offer modern architectures open to different current software technologies and provide users with tools to build their own functions. This makes it possible to organise objects in a way fully compatible with MES and ERP IT environments.

10.5 UAG: Application generators

Unity Application Generator (UAG) is both design software and a generation tool to integrate PLCs and supervisory systems (SCADA) or human-machine interfaces (HMI). To do this, it uses the basic single database technique of hybrid control systems combining DCS and PLC functions. Though UAG is used for single PLC applications, it is especially useful in multiple PLC architectures. It uses the full power of UNITY Pro which it closely associates with supervisory systems such as Monitor Pro, I-Fix and so on.

■ Significant savings in development time

UAG and its underlying concepts enable developers to get involved with the customers at a very early stage. Reusable objects and easy updating and testing can cut project time by 20 to 30% (⇒ Fig. 20), which can mean several months for big projects.



↑ Fig. 20 Time and cost reduction

■ Main features of UAG

- A single input for the PLC/HMI database (SCADA);
- use and reuse of library objects;
- process application configuration;
- incremental generation for SCADA;
- global resource mapping;
- PLC application (code, variables, configuration, communication, etc.);
- SCADA application (symbols, mimic diagrams, variables, attributes, alarms, etc.);
- communication management (Ethernet, ModBus+);
- generic interface in XML.

Note: "Generic Plug-in Interface" is an open interface for other SCADA systems.

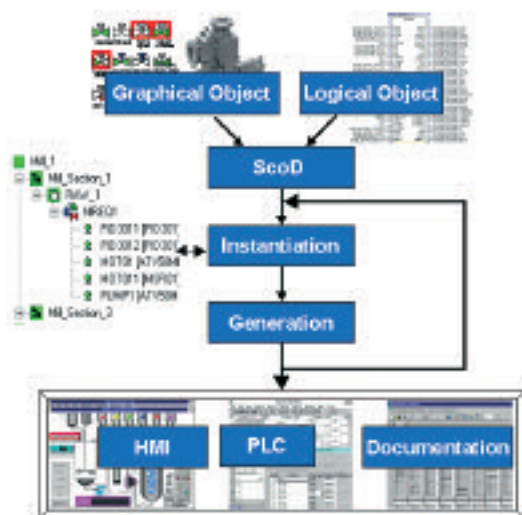
■ UAG operation

Unity Application Generator is made up of three tools (⇒ Fig.21).

Tool	Field	Library
SCoD Editor.	Libraries.	Specification of control modules, DFB or EFB import, attribute specification and screen configuration definition (within UAG).
UAG Customisation	Customisation	Definition of user profiles in a project including: naming rules, hardware specification catalogues, libraries.
Unity Application Generator	Projet	Project design, functional analysis and application generator.

↑ Fig. 21 UAG tools

The links between tools is summed up in figure 22.



↑ Fig. 22 UAG synoptic

□ SCoD object editor

Unity Application Generator (UAG) is an object-oriented tool based on control modules. A control module describes a process unit and covers all aspects of the automation task:

- PLC logic;
- representation for the operator in the supervision system;
- mechanical and electrical properties of the unit;
- maintenance and troubleshooting;
- multi-facet representation of these elements in UAG is called Smart Control Device -(SCoD), the equivalent of control modules in standard ISA 88.

Control instantiation has an equivalent physical representation which can be:

- an actual component, which can be held and inspected, such as a motor, a valve, a temperature transmitter, etc.;
- a control element used to adjust other functions such as a PID loop, timer or counter.

Control modules are defined and used in type libraries; the SCoD editor is the tool which creates, updates and groups specific customer controls in the objects (Smart Control Device).

Definition of rules and properties in the SCoD editor based on the DFB/EFB interface includes:

- graphic user interface (GUI);
- mandatory configuration of the SCoD instance;
- optional configuration of the SCoD instance;
- SCoD instance inheritance;
- simple and complex relations inherited by the selected module and other SCoDs;
- definition of data transmitted to and from the HMI;
- definition of data related to the topological model;
- specific HMI information such as alarm texts, measurement units;
- access levels:
 - per module reference to specific HMI information such as ActiveXs and symbols;
 - per module reference to specific PLC information such as DFBs/EFBs;
- SCoD documentation.

10. Data treatment and software

10.5 UAG: Application generators



↑ Fig. 23 Screen shot of UAG editor

An example of a Smart Control Device could be a valve. A valve is generally used as a cut-off device to prevent or allow the flow of a fluid or gas in a pipe. It is usually linked to three digital signals:

- limit switch open or closed (2 signals);
- the signal for the actuator.

There are many different standard valves ranging from the smallest actuated by solenoids to the largest actuated by motors.

The properties are assigned for the type of valve from the PLC interface (API).

The default valve insert mode is “Energise-to-Open”, though the user can specify “Energise-to-Close”.

The “Travel Time-out” operating time must be within the [min Value.. max Value] interval.

Figure 23 shows a screenshot of the editor.

□ UAG Customisation Editor

This editor (⇒ Fig. 24) is used to define a common language for different people (operation, automation, maintenance). It describes the list of elements used and their definition, objects (SCoD) and human-machine interfaces (HMI).

Customisation starts with:

- the list of SCoD libraries;
- the list of authorised hardware modules;
- the access level;
- definition of process element names;
- specific HMI or SCADA properties.

To return to the valve example, the user has to define the valve exactly so that it cannot be confused with another in the system. The user is going to standardise the structure of the system component names: <position in 4 digits> <English abbreviation in 3 digits> <Type of element from 1 to 6>. A given valve will have a single name such as 2311VAL4.

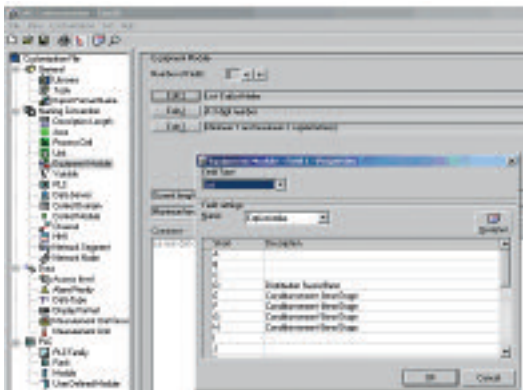
All elements can thus be defined by:

- the location of the SCoD corresponding to the valve;
- the section of the program in the PLC;
- the hardware modules linked to the PLC, etc.

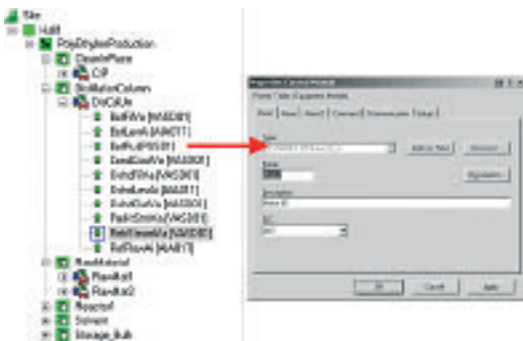
In the PLC and HMI applications, standardisation is defined for all operators and experts in automation and the process. The project follows the rules without exception and maintenance staff can be trained with the tools. Project management is facilitated by knowing the number of valves.

□ Application generator

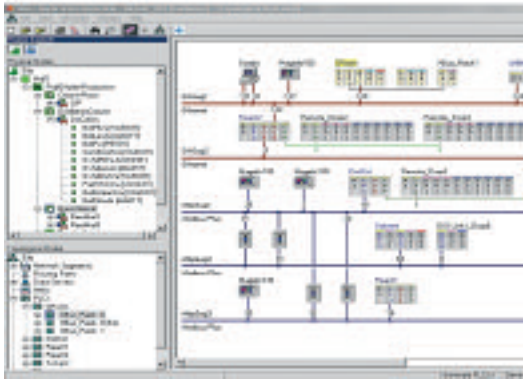
Unity Application Generator is a design and functional analysis program generating applications for PLCs and SCADAs (⇒ Fig. 25).



↑ Fig. 24 List elements description



↑ Fig. 25 Screen shot UAG physical model and control model parameter setting



↑ Fig. 26 Screen shot of UAG topological model

There are two independent tasks to generate the physical and topological models:

- The physical model describes the process in a tree structure of elements as shown *figure 25*.
- The topological model describes the hardware of the automation process, including the PLCs, inputs/outputs, networks, PCs, etc. as shown *figure 26*.

■ UAG and ISA88 standard

□ Standard ISA 88: “Advanced” process control organisation

“Advanced control” focuses on the basic algorithm used to enhance the behaviour of automatic control devices. It also takes into account organisational and economic disruptions and can go as far as casting doubt on the equipment, methods and results of the manufacturing process.

The ANSI/ISA-88 standard offers efficient concepts for the functional organisation of applications to achieve the degree of robustness required for ongoing adaptation of systems subjected to such constraints.

This standard focuses on a number of essential points:

- flexibility, development that is less complex and shorter and mastery of the process;
- object-oriented approach and reusable object classes;
- separation of the procedure and the process;
- separation of process control and equipment automation control.

Detailed information on this standard is available on [http:// www.s88.nl](http://www.s88.nl).

Unity Application Generator uses the ISA 88 batch processing terminology in « Part 1: Models and Terminology ». With the structure of this standard, UAG users can break down process tasks to fit their own rules and then use UAG and the elements defined to rebuild the process.

10.6 Definition of the main abbreviations used

DCS Distributed Control System

HMI Human Machine Interface

PLC Programmable Logic Controller

SCADA Supervisory Control And Data Acquisition (see the paragraph on SCADA)

UAG Unity Application Generator

MES Manufacturing Execution System, a computing system primarily designed to collect complete or partial factory production data in real time. These data are then used for a number of analysis exercises:

- tracking, genealogy;
- quality control;
- production monitoring;
- sequencing;
- preventive and remedial maintenance.

ISA has standardised the MES structure.

An MES is often situated between the industrial automation system and the ERP systems like SAP.

ERP Enterprise Resources Planning, software to manage all the processes in a company by integrating all its functions such as human resources management, accounting and financial management, decision-aiding tools, sales, distribution, supplies and e-commerce.

CRM Customer Relationship Management, a new paradigm in the field of marketing.

The purpose of CRM is to create and maintain a mutually beneficial relationship between a company and its customers. In this commercial relationship system, the company focuses on winning customer loyalty by offering them service of a quality they cannot find elsewhere.

The image frequently used to illustrate this concept is the relationship between small shopkeepers and their customers. Customers are rewarded for their loyalty by the fact that the shopkeeper knows their habits and acts accordingly (like a baker who puts aside a loaf baked the way you like it and allows you credit).

