



**HAL**  
open science

## **Towards Automation Component Based Design Hands-on Training**

Arnaud Lelevé, Mady Guillemot, Nabil Fakhfakh, Didier Noterman

► **To cite this version:**

Arnaud Lelevé, Mady Guillemot, Nabil Fakhfakh, Didier Noterman. Towards Automation Component Based Design Hands-on Training. International Conference on Integrated, Virtual and Interactive Engineering for fostering Industrial Innovation, Oct 2008, Beijing, China. pp.1-8. <hal-00331581>

**HAL Id: hal-00331581**

**<https://hal.science/hal-00331581v1>**

Submitted on 1 Jun 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Towards Automation Component Based Design Hands-on Training

A. Lelevé, M. Guillemot, N. Fakhfakh, D. Noterman

Université de Lyon  
INSA-Lyon, LIESP, F69621, Villeurbanne, France  
+33(0)4.72.43.60.47 / +33(0)4.72.43.85.38

*E-mail* : {arnaud.leleve , mady.guillemot, nabil.fakhfakh, didier.noterman}@insa-lyon.fr

**Abstract:** Automation world is living a change of design methodologies. Old top-down approaches are being replaced by component based design allowing design, realization and exploitation cost gains. Automation training in universities also evolves in order to train future automation industrialists up to date on new design paradigms and technologies. This paper presents three projects illustrating this evolution, based on training component based design applied to simulation of operative and control parts use.

**Key words:** Automation, component based design, UML, operative part simulation

This paper presents three hands-on answers to this industrial evolution: the design of a ventilation system, a sequential gearbox and an elevator. These three samples introduce component based design and appeal to simulation of parts of the studied systems. The gearbox and the elevator hands-on resources are available on AIP-Primeca-RAO remote laboratory platform<sup>1</sup>. The remainder of this paper is structured as follows : we present at first this industrial evolution. This is followed by details on the use of simulation in automation. Next, we depict these three samples and finally, we discuss this evolution and describe future directions for this work.

## 1- Introduction

Nowadays, a modern automated system is a networked distributed control system featuring intelligent components (sensors, pre-actuators, networks, ...). Thus, their design methodologies incline towards approaches which reduce design and coding time, allow reusability of parts of programs and enable capitalizing on know-hows. These approaches are based on Object Oriented paradigm inspired from computer scientists. Design modeling languages based on UML, yet used for real-time and embedded systems, now become used in automation world to model factory automated systems.

This little revolution also affects how automation is taught in universities. As this technical discipline is largely taught by means of hands-on sessions, corresponding hands-on trainings have to be updated with recent hardware, field buses, modern programming software, ... designed with these new approaches. As every french educational institute can not afford industrial up to date equipment to train their teachers and students on, these resources have been shared in a country-wide network of workshops : *AIP-Primeca*. AIPs provide, among other services, hands-on resources to train people (teachers, students, industrialists, ...) on automated production systems (design, use, maintenance, ...). These resources are available locally in each center (and also remotely through Internet, as for ARI project presented in [CG1]). Nevertheless, sharing heavy systems has constraints that operative and control part simulation modern techniques can bypass to help in providing the best service to universities.

## 2- Automated Production System Design

Nowadays, design methodologies used traditionally by automation experts are evolving towards approaches which reduce design and coding time, allow reusability of parts of programs and enable capitalizing on know-hows. Indeed, traditional methodologies lead to monolithic hardly reconfigurable and modifiable control systems, while modern systems are organized as Distributed Control Systems (DCS) leaning on industrial field networks hosting (more and more) intelligent (able to talk directly on a field bus, to self-diagnose, to be autonomous, ... according to the component) sensors, pre-actuators, motion drivers, business specific components, ...

### 2.1 Chronology of an automation project

Whether top-down or a bottom-up methodology is used, an automation project follows a lifecycle modeled on the V-cycle used by computer scientists (except in more recent methodologies such as eXtrem Programming, Value Based Software Engineering... [B1]). This latter :

- starts from global specifications and refine them (detailed specifications),
- designs the software (in two steps: general and then detailed design),
- codes it,

<sup>1</sup> See <http://aipportail.insa-lyon.fr:8085/aiprao/index.html>

- tests it,
- integrates it in production environment,
- validates it in production mode.

Automation projects use the same global chronology. The difference is one has to design (and then to make) operative part and control part simultaneously. Compared to traditional automated systems, modern ones include Human Machine Interactions and communication with higher level software (Manufacturing Execution System, Enterprise Resource Planner, Maintenance Management Softwares, ...).

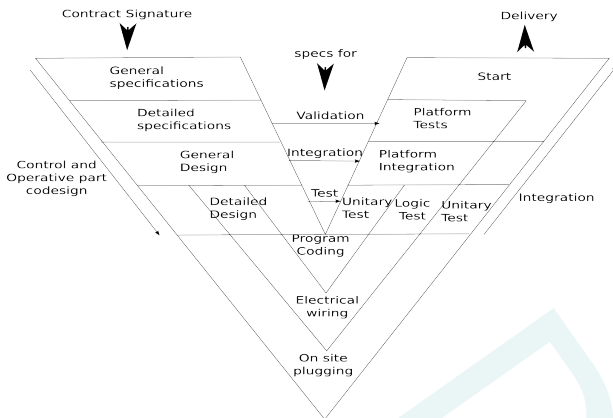


Figure 1: traditional Automation V Cycle

### 2.2 Classical design methodology

Traditional tools of functional analysis used by automation designers are axed on a logic of specification defining :

- functioning modes,
- anomalies and their treatment,

The most popular tool in France (which is even more popular in universities than in industry) is the GEMMA which is a study guide of run and stop functioning modes. It has been developed by ADEPA which is a national agency which aims at developing applied industrial engineering. The GEMMA consists in a graphical analysis procedure enabling to determine different functioning modes and paths and conditions to pass from a mode to another one. This generic diagram is generally a preliminary step before elaboration of functional specification using IEC 60848 GRAFCET charts (which were preliminary designed to be a specification tool and has become a programming language ; more recent IEC 61131 Sequential Functional Chart (SFC) is only a programming language). Then follows the coding in one out of the four IEC 61131 standard languages (Ladder Diagram, Structured Text, Function Block Diagram and Instruction List). This top-down approach leads to a hierarchical control structure characterized by:

- a synthetic representation which facilitates comprehension and application analysis,
- a great efficiency and a quick start-up and easy maintenance.

However, every analysis is dedicated to a new system and a

new application requires to reiterate the whole study. So there is no reusability. Moreover, this approach favors control structuring but neglects other facets of the study (electrical wiring, mechanical and energy study, Human-Machine Interactions, ...) and its exploitation (supervision, alarm management, ...).

### 2.3 Component-Based approach

Nowadays, design methodology used traditionally by automation experts is evolving towards an approach which :

- reduces design and coding time,
- allows reusability of parts of programs and,
- enables capitalizing on know-hows.

This new approach is based on Object Oriented (OO) software paradigm. It keeps a part of its interesting properties: class and object notion, encapsulation, aggregation). But it leaves apart more evolved properties: heritage and polymorphism.

Recent new standards such as the IEC 61499 open standard for distributed control and automation reflect this trend [HF1]. In automation, the term “Component” is used in place of “Class” as it features more information and process than simply programming data and methods.

Automation Component Based Approach [J1, T2] leans on a methodology structuring an equipment in operative subsets. This analysis is realized according to two complementary dimensions

- the physical dimension,
- the functional dimension.

An automation component is representative as well of an operative part of an equipment to be automated (associated actuators and sensors) as a functional control module [BG1]. Moreover, an automation component is relatively generic (it can be imported from a standard or a specific library). It is then later directly instantiated (as many times as necessary) in an application or specialized (and then instantiated). It answers to two main needs:

- reusability :which enables capitalizing know-how,
- structuration which complements top-down traditional approaches by bottom-up ones.

Automation Components are autonomous in a functional point of view and multi-facets (it features, for instance, part list, electrical wiring plans, mechanical plans, Human-Machine Interactions, maintenance procedures, ...). Each newly designed set of business components is to be recorded and administered in a library which is enriched as needs and studies go on. This perpetual recording participates in know-how capitalization. The versioning of recorded components can be managed through a Product Lifecycle Management (PLM) software, for instance.

As in OO approach, a component is specified (as a class) by

a set of parameters, inputs, outputs and a specific program linking the whole to create the research behavior (see last section for samples). Its instances (objects) will share the same properties, same behaviors and same semantics but they will each one live their own life according to the way they are stimulated [VH1]. One can find in these methods:

- a static aspect which describes an object characteristics and its links with other objects,
- a functional aspect which defines its method behaviors
- a dynamical aspect which describes interactions between objects and the different states which every object can be in.

This new way of programming has modified project organization. New businesses appear :

- business specifiers : design standard components, their facets (programming interface (inputs/outputs/parameters), HMI widgets, electrical and mechanical models and constraints, part list, network configuration, ...), their data structure and interfaces,
- prototypers : make every facet of standard components, test each unit,
- component administrators : manage component structure rules, program and electrical wiring plan generation, documentation, validation cycles,
- application architects : reuse standard and business components to integrate them into a global application which they finally validate.

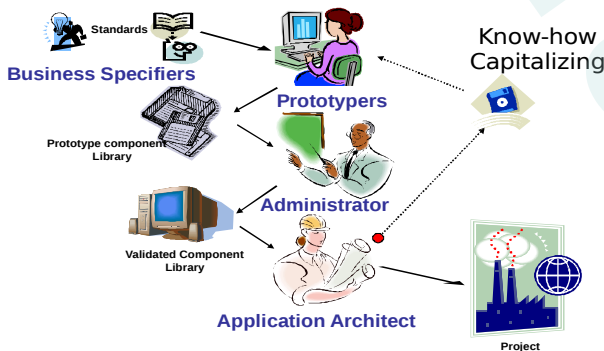


Figure 2: future automation component life cycle ?

### 2-4 UML modelling in Automation

As automated system design is moving from document centric to model centric, modeling languages such as UML become available to automation designers [SD1].

UML (Unified Modeling Language) is a graphical language modeling data and process. It is an open formalism very accomplished for object oriented modeling broadly used in software engineering.

UML does not define model elaboration process : it is not a methodology nor a toolbox, uniquely a graphical modeling language. In 2003 the Object, Management Group (OMG) released the version 2.0 of the UML. UML2 defines thirteen

diagrams which use is free according to one's appreciation [BR1]. UML software modeling is lead according to three points of view which appeal to some of the 13 diagrams:

- functional point of view : use-case diagram
- static point of view : class diagram
- dynamical point of view : state diagrams, sequence diagrams.

The strength of UML2 is that it becomes possible to create profiles for specific business fields. Profiles are both specializations of UML2 generic diagrams and extensions of initial formalism, in order to specify clearer behaviors (in each business field), such as OMG SysML [OM2, CK1]. SysML aims to support the specification, analysis, design, verification, and validation of systems that include hardware, software, data, people, procedures, and facilities.

Even if UML was at first targeted towards pure computer science projects, its popularity and genericity have rapidly seduced designers for real-time systems [OM1], embedded systems [AF1], mechatronic systems [BG2] and then automated systems [KN1, RK1, OM2].

### 2-5 Training on automation design

The evolution of these methodologies requires to keep university and technical college teachers up to date so that they can train efficiently students who will have to use them later in industry.

To train teachers and then students to these new methodologies, as for many technical disciplines, it is most efficient to illustrate typical use-cases by hands-on trainings on manufacturing equipments [FI1]. But such resources are few and their availability, linked to platform office opening hours, is an obstacle to their use in lifelong training as well as with an international audience.

But, every educational institute can not afford industrial up to date equipment to train their teachers and students on.

This is why, in France, these resources are shared from a country-wide network of workshops : AIP-Primeca

The A.I.P. (Inter-Establishment Production Engineering Workshops) are regional resource centers, created in 1984, and used as support for high-level training in the Industrial Automation field. AIP-PRIMECA (Computing Resource Centers for Mechanical Engineering) were created in 1991 to promote the use of computing tools in the design of mechanical products and establish training programs in Computer-Aided Mechanical Design. Its missions, allotted by the french education ministry concern:

- initial learning,
- life long learning,
- technological transfer (from research laboratories to industry),
- knowledge diffusion (towards universities and industries).

Out of ten regional clusters, AIP-Primeca-RAO (Rhone Alpes West) is a resource for various high schools in french Rhône-Alpes Region. Its eight partner universities are also its main users, in :

- Lyon (INSA, Ecole Centrale, Université Claude Bernard Lyon 1, ECAM, IUT Lumière of University Lyon 2),
- Saint-Etienne (ENI, Ecole des Mines, Université Jean Monnet Saint-Etienne).

The resources at users' disposal are mainly composed of :

- industrial equipment (such as CNC machine-tool, conveyors, assembly automation systems, robots, ...),
- computer systems featuring industrial software (Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Manufacturing Execution System (MES), process simulation, shop control, ...),
- communication and cooperation systems (LAN, WAN access, video-communication, Computer Supported Cooperative Work (CSCW), ...).

Concerning this paper topic, AIP-Primeca-RAO has been proposing resources for automation hands-on training as a local usage in its buildings (and also remotely through Internet [CG1]).

Today, RAO works on the “Collaborative Enterprise” and the “Product Lifecycle Management” (PLM). These topics include the “Collaborative Design” and the Automated Production System Building.

### 3- Operative Part Simulation

#### 3.1- Industrial needs

In Automated System design, programming requires testing on the real system. This involves:

- to have access to the target system (in case of retrofitting, the production system hardly stops for maintenance activities, so it is not available for programming tests),
- that its prohibitive cost generally prevents from creating a clone just for testing,
- that test can be destructive so very risked for hardware and also possibly for surrounding people,
- when several systems are embedded, it is difficult to isolate one to test it without making trouble with others.

#### 3.2 Educational needs

In a educational context, more constraints appear:

- potentially destructive tests should not hurt anybody nor destroy educative hardware where students are people who are not experts of systems and opportunities to break it are more present,
- industrial hardware is very expensive so very few apparatuses are available for a growing number of students, moreover through remote laboratories,

- when it is not possible to work on it remotely by way of a remote laboratory platform, it is only available at office opening hours.

#### 3.3 Advantages of appealing to simulation:

A solution to these constraints is both industrial and educative contexts is to appeal to software simulation of Operative Part. Its advantages are :

- this solution is cheaper than real heavy hardware so there is no heavy investment in specialized machines,
- the material quantity is infinite,
- the simulator is always available and it can be run,
- independently in parallel on several desktops,
- it can be done anywhere as soon as one owns the simulation application,
- no human and hardware risk of loss or destruction,
- possibility to reduce the complexity of the situation to focus testing on specific aspects (and limiting blockings hazards),
- possibility to enrich information picked up from the system (modifying spacial and timing parameters, displaying normally invisible informations and mechanisms, infinite replay...).

Moreover, in educational context:

- the absence of risk enables making this educational resource in self service,
- this solution can be relatively easily proposed remotely as a virtual laboratory, which answers needs of scalability and raises opening time to 24h/24h,

Nevertheless, simulation is always based on a model which represents reality with a certain degree of approximation. The more the model is realistic (taking into account frictions, environmental perturbations, ...), the more it is heavy and difficult to design and run.

Also, software simulation mainly requires abstraction : no touch, no smell, no taste, ... Every return of information from the system is transmitted by vision (2D or 3D) and in some cases by haptic return (returned efforts on a joystick) [CN1].

#### 3.4 How Simulation is used

Typical industrial use of operative part simulation is when testing one's program before uploading it on the real system.

It can be used in various educative situation :

- in courses, to illustrate phenomena
- in exercises, to accompany students in their learning
- in virtual laboratories
- in remote laboratories before testing programs on real shared systems

#### 3.5 How Simulation is done

Simulating operative part can be done either by specific softwares programmed commonly in *C* or *VisualBasic* languages; possibly on a real-time system.

It may be also done by means of scientific software such as *MathWorks MATLAB+Simulink* and more likely (in automation area) by National Instruments *LabVIEW*.

Another approach consists in reusing automation software such as supervision ones which are more likely to communicate with Programmable Logic Controllers (PLC) and can be more easily (generally graphically) programmed than in *C/VisualBasic*.

A last solution consists in appealing to specialized simulation software, such as *Geensys ControlBuild*.

Nowadays, main automation suppliers propose software simulators which simulate a control part: their PLCs can so be emulated by a PC. For instance, Schneider Electric Unity enables simulating the target PLC in the development stage.

*Geensys ControlBuild* enables to simulate both Control Part (the PLCs) and Operative Part. An hybrid simulation can be created with certain parts of both parts which are simulated and others which are the real ones.

Moreover, the standardization of field bus communication over Ethernet enable to easily simulate both operative and control parts on one or several networked PCs.

### 3.6 Educational use

In education context, resorting to simulation brings numerous important advantages. It can be used:

- to dynamically illustrate a course,
- to enhance interactivity of classroom exercises,
- in local laboratories to replace part or all apparatuses,
- in remote laboratories to compensate for the difficulties due to the remote handling.

This is partly why first implementations of distant hands-on laboratories were based on simulations [CN1].

## 4- Three incremental approaches

We propose in this section to illustrate previous discussions with three educational projects aiming at training students on automation component design with support of UML diagrams. First one uses a PLC to simulate lacking operative part components. Second and third ones use a supervision software to simulate operative part.

### 4.1- Industrial ventilation project

Since 2007, students of Industrial Engineering at INSA Lyon have been given a new automation platform for hands-on training [CG1] featuring real PLCs, electric motors, motion drivers, ... In this project, nine groups of students had to work on the design of an automated global system. They had access to the apparatuses during three sessions of 4h each. They used this time to get used with the development software Unity Pro and to test their programs. As apparatus sets are generic (1 PLC, 1 field network (Modbus, Profibus, CANOpen, AsI, Ethernet), 1 motion driver, 1 motor starter and two motors),

they have had to make hybrid simulations where :

- one part of the operative part is real: two motors, one controlled by a motor starter and the other one by a motion driver,
- the other part is virtual: it consists of models designed and run in the PLC in order to simulate the rest of the operative part.

Component Based Design has been used. Some basic components (one interfacing a cylinder and the other simulating it) had been provided in a tutorial. Students could reuse them as is and modify them when necessary.

To illustrate this application, we present one out of these projects, which consists in managing the ventilation of an industrial system which temperature has to be controlled. This system is depicted in figure 3. It consists of a fan which requires to be lubricated. A lubrication circuit must be started before turning the fan. A control law is designed to set the fan speed at the right value according to the tank desired temperature.

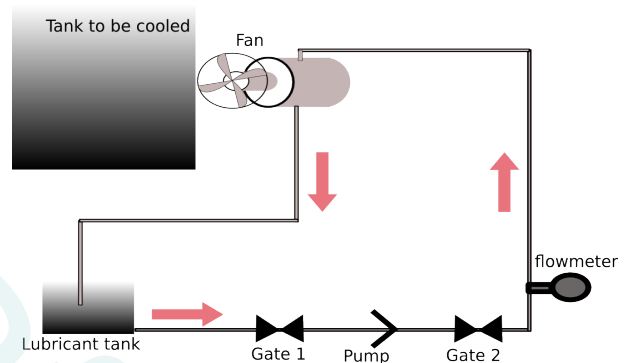


Figure 3: ventilation system to control and simulate

In development phase, students use a PLC simulator (provided by Unity Pro) which enables to debug without constraints of apparatus busy planning. In this phase, they have to design a whole simulated operative part. These two parts are split into different sections in order to disable the simulation sections earlier.

In the test phase, they have access to real apparatuses. In this phase :

- the fan is simulated by a real motor controlled by a motion controller,
- the pump is simulated by a motor fitted with a motor starter,
- the gates and the model of the system to cool remain software simulated (the temperature evolves according to the speed) : see fig. 4.

One real PLC hosts the control part and the software simulated operative parts.

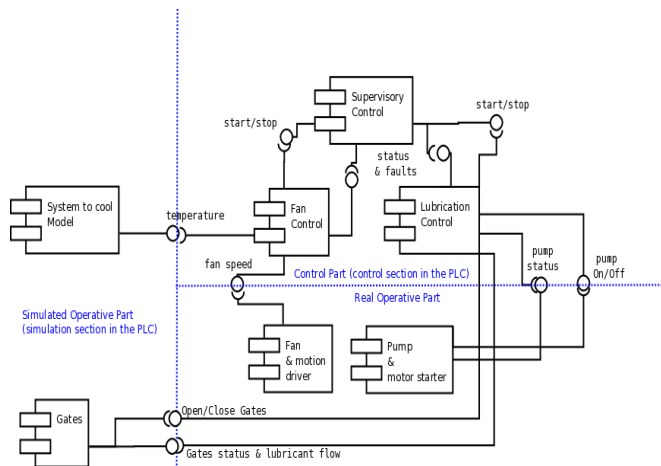


Figure 4: ventilation system components

In this case, the programming of a few real systems enabled students to deal with real industrial systems. The low number of real apparatuses was compensated by the use of simulation. The restricted time (3x4h) given to learn how to use available softwares, to resolve attributed problem and to test it is actually borderline. Four supplementary hours are necessary for students to finish correctly their work. Sixteen supplementary hours would have been necessary without simulation features. Moreover, the fact that the simulator is integrated into the development software made things easier for students as they had not to learn different dedicated softwares.

4.2- Sequential gearbox control

This project concerns on-board control of a mechatronic system which study can be approached in different ways:

- by a whole simulation. It can be used as a pedagogical support to a case study in exercises,
- by a real system dealing with technological aspects : featuring a PLC, pre-actuators (motion control, pneumatical distributors), actuators (electrical motor, pneumatical cylinders) field network (*CANOpen*), Human-Machine Interface (HMI).

In local or remote laboratories, a test system with the real gear-box is available.

Whatever approach, a functional study is led through UML diagrams conforming to previously exposed methodology. Figure 5 shows a sequence diagram corresponding to gear changes.

In the case of a global simulation, a single PC computer enables :

- to emulate control part (in the Schneider Electric programming software Unity Pro),
- to simulate the operative part (in the HMI emulator featured by Schneider Electric HMI programming software Vijeo-Designer).

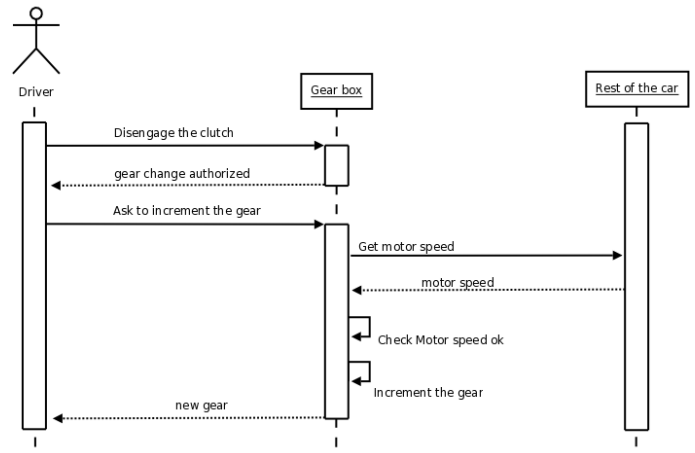


Figure 5: sequence diagram corresponding to a gear change.

The liveness is obtained by way of scripts written in *JAVA* which are associated with graphical objects. Simulation facet is realized and integrated into a component library. Basic components can then be put together in order to create more complex components such as double tandem cylinders with their own sensors and transducers.

Global system is architected on Ethernet network at high level (PLC – HMI panel – programming computer) and *CANOpen* for low level communication (PLC – Motion driver – networked cylinder distributors).

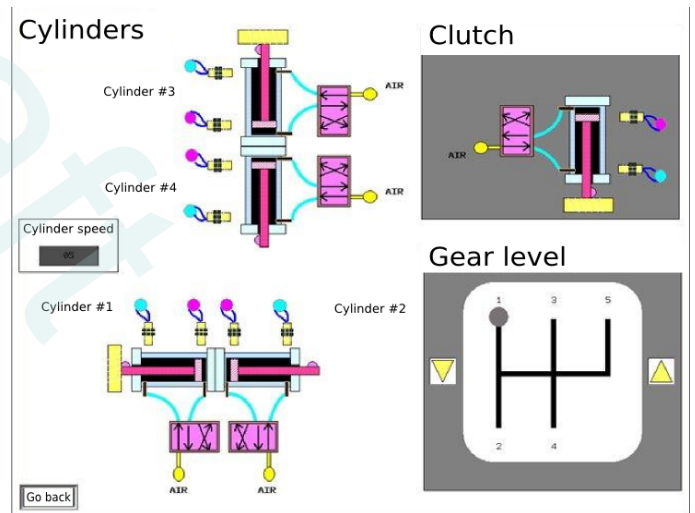


Figure 6: screenshot of the simulated operative part

In this project, simulation was necessary for students in order to test their program before applying it on the real system. In the previously introduced project, the system could not be broken as there was just one or two rotating independent axes per system. In this project, actuating cylinders have to be coordinated to engage the good gear. If students make mistakes in this movement, pieces of the gear box may brake themselves. Moreover, if students engage wrong gears (5th gear to first one in place of 4<sup>th</sup> one), this may also break pieces as in a real car, even if lower speeds are used in this system.

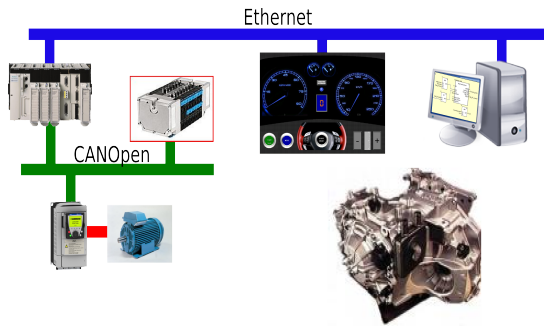


Figure 7: gearbox system architecture

### 4.3- Elevator

In this application, the study focuses on the functional division of an elevator. This elevator is composed of :

- a lift shaft including conductor rails, electric wires and sensors,
- a cabin featuring a set of indoor doors and an HMI panel,
- a machinery moving the cabin.

The analysis enables to create a library of elementary components required for the control. These components are designed as Derived Function Blocks (DFB) :

- door component dealing with opening and closing of doors,
- winch component controlling cabin moves,
- call priority management dealing with calls from people at different floors waiting for the cabin and the people inside the cabin.

These three DFB are then put together in a new component dealing with the control of a whole elevator.

Once the elevator DFB checked, it can be used in many situations and coding time will then be much decreased compared to a new study. This new DFB can then be instantiated several times to control several synchronized elevators.

Concerning operative part simulation, a 2 cabin elevator simulator has been developed by using *Vijeo Designer*, such as for the previous gear box. This model is based on two instances of a component simulating operative parts for a single cabin. Figure 8 is a screen shot of the resulting HMI.

This simulator enables user to drive the elevators by mean of indoor keyboards and call buttons on each floor. Security behaviors can be simulated such as people preventing the closing of doors or overweight in case of too many people getting inside the cabin.

This project aims at illustrating the advantages of Component Based Programming towards students. Indeed, as soon as a floor control (resp. operative part) model is validated (featuring the control of external doors, call buttons, external display, ...), students (resp. tutors) just have to instantiate it as times as necessary to create a control model for a whole elevator covering several floors. In the same way, as soon as the control

(resp. operative) part of an internal door is modeled, students (resp. tutors) can use this model for elevators with two internal doors.

Thus, a tutor can prepare a model of the operative part for an elevator with an arbitrary number of floors and then make students design neatly their control model. The second part of the exercise consists in asking to students for a control model of a slightly different operative part. Good models would require very few adjustments to fit the new operative part.

This level of freedom would be difficult to obtain with a scale model. Experience shows that students would have opted for a traditional way of programming as they work on a frozen configuration and then to fiddle it to fit the second configuration, with some difficulties and much more time.

The gearbox and the elevator hands-on resources are available on AIP-Primeca-RAO remote laboratory platform<sup>2</sup>. Development softwares are available online and students can use them exactly as if they were *in situ*.

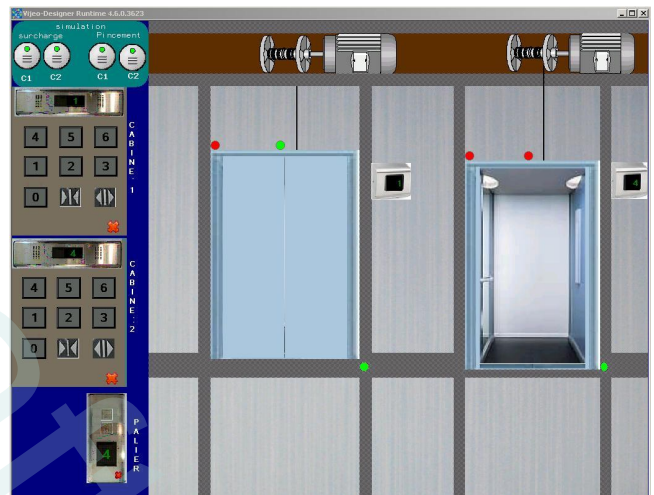


Figure 8: double cabin elevator HMI

### 5- Conclusion

Nowadays, design methodologies used traditionally by automation experts are evolving towards approaches which reduce design and coding time, allow reusability of parts of programs and enable capitalizing on know-hows. This paper summed up main aspects of this evolution from an industrial point of view and more accurately on an educational point of view. Component Based Design associated with UML (or a specific profile) are becoming the way to design factory automation systems. Along this evolution, computer tools also evolve and operative and control part simulations are now integrated in automation programming suites. This is why educative institutes are also evolving in order to produce future automation experts up to date on today design methodologies, tools and technologies. Three samples were presented involving the use of simulation of part or whole of

<sup>2</sup> See <http://aipportail.insa-lyon.fr:8085/aiprao/index.html>

the operative and control part.

As this evolution progressively leads to new businesses (business component furnisher, component library manager), universities will have to prepare their students to these skills.

We are working on hands-on training projects delimiting each role in the automated system design chronology to help students to build skills in each design step.

## 6- References

- [AF1] Agnès M., Frati P., André C., Rigault J.P., UML et le paradigme synchrone: application à la conception de contrôleurs embarqués, In SPORTS Project Report, <http://www.i3s.unice.fr/%7Emh/RR/2002/RR-02.02-C.ANDRE.pdf>, february 2002.
- [B1] Boehm B., A view of 20th and 21st century software engineering, In Proceedings of the 28th international conference on Software engineering, Shanghai, China, Pages: 12 – 29, 2006
- [BG1] Buckley J., LeGear A. P., Exton C., Cadogan R., Johnston T., Looby B., Koschke R., Encapsulating targeted component abstractions using software Reflexing Modelling, In Journal of software maintenance and evolution, 20(2), pp. 107–134, march 2008.
- [BG2] Burmester S., Giese H., and Tichy M., Model-Driven Development of Reconfigurable Mechatronic Systems with Mechatronic UML, In Model Driven Architecture: Foundations and Applications, LNCS 3599, (ed.) Springer Berlin/Heidelberg, pp. 47-61. Springer-Verlag, doi 10.1007/11538097\_4, ISSN 0302-9743 (Print) 1611-3349 (Online), august 2005.
- [BR1] Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User Guide (2nd Edition), The Addison-Wesley Object Technology Series
- [CG1] Coquard P., Guillemot M., Leleve A., Noterman N. and Benmohamed H., AIP-Primeca RAO Remote Laboratories in Automation, In International journal of online engineering, <http://www.i-joe.org/ojs/viewarticle.php?id=206&layout=abstract>, 4(1), 2008
- [CK1] Chiron F. and Kouiss K., Design of IEC 61131-3 function blocks using SysML, In Proc of Mediterranean Conference on Control & Automation (MED '07), 27-29 June 2007, Athens, Greece, pp. 1-5, ISBN: 978-1-4244-1282-2, doi 10.1109/MED.2007.4433695, 2007
- [CN1] Corter J. E., Nickerson J. V., Esche S. K., Chassapis C.; Im S. & Ma J, Constructing reality: A study of remote, hands-on, and simulated laboratories, In ACM Trans. Comput.-Hum. Interact., 14(7), ACM, 2007
- [FI1] Fiesel L. D. and Rosa A. J., The Role of the Laboratory in Undergraduate Engineering Education, In Journal of Engineering Education, pp 121-130, January 2005.
- [HF1] Hussain T. and Frey G., Migration of a PLC Controller to an IEC 61499 Compliant Distributed Control System: Hands-on Experiences, In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2005), 18-22 April 2005, pp. 3984-3989, 2005
- [J1] Jeong, H., Interactive Technologies and Sociotechnical Systems, The Component Based Factory Automation Control in the Internet Environment, Springer Berlin / Heidelberg, pp. 290-299, 2006.
- [KN1] Köhler Hans J., Nickel U., Niere J., Zündorf A., Integrating UML Diagrams for Production Control Systems, In Proceedings of the 22nd IEEE International Conference on Software Engineering (ICSE'00), ISBN:1-58113-206-9, Limerick, Ireland, pp. 241 – 251, June 4-11, 2000.
- [OM1] Object Management Group, UML Profile for Schedulability, Performance and Time Specification, Version 1.1, 2005
- [OM2] Object Management Group, OMG System Modelling Language (SysML) Specification, Final Adopted Specification, ptc/2006-05-04.
- [RK1] Ritala T., Kuikka S., UML Automation Profile: Enhancing the Efficiency of Software Development in the Automation Industry, In Proc. of 5th Int. IEEE Conf. on Industrial Informatics (INDIN 2007), Vienna, Austria, July 23-26, 2007
- [SD1] Servat D., Chiron F., Kouiss K., Meurisse T., Souchet G., CLIPS: experiments on model-driven engineering for production systems, In Proc. of 10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2005), 19-22 Sept. 2005, Catania, Italy, ISBN: 0-7803-9401-1, doi 10.1109/ETFA.2005.1612563, pp. 482-489, 2005.
- [T2] Trkaj, K., Users introduce component based automation solutions, In Computing & Control Engineering Journal, 15(6), pp. 32 – 37, Dec. 2004-Jan. 2005
- [VH1] Vyatkin V., Hanisch H. M., Reuse of Components in Formal Modeling and Verification of Distributed Control Systems, In Proc. of 10th IEEE International Conference on Emerging Technologies and Factory Automation, (ETFA 2005), Catania - Italy, September 19 - 22, 2005, pp 129 - 134.