Advanced Tools for the Control Engineer in Industry 4.0

Alexandre Philippot, Bernard Riera, Vinay Kunreddy, Serge Debernard

Advanced Tools for the Control Engineer in Industry 4.0

Alexandre Philippot, Bernard Riera
Research Centre for Science and Information Technology and Communication, CReSTIC (EA3804)
University of Reims Champagne-Ardenne
Reims, France
alexandre.philippot@univ-reims.fr
bernard.riera@univ-reims.fr

Vinay Kunreddy, Serge Debernard
LAMIH - Laboratoire d'Automatique de Mécanique et d'Informatique Industrielles et Humaines,
Univ. Valenciennes, CNRS, UMR 8201
F-59313 Valenciennes, France
VinayReddy.Kunreddy@etu.univ-valenciennes.fr
Serge.Debernard@univ-valenciennes.fr

***Abstract*— To make Industry 4.0 a success, it is necessary to take into account the human component. To design safe controllers, engineers must dispose innovative and human adapted methodologies. The paper proposes two advanced powerful tools (Model-Checking and Virtual Commissioning) which could modify the work of the automatic control engineers in the future. Model-Checking is used as an off-line verification of structural properties of a controller specification. Virtual Commissioning is used on-line to test the functional part.**

Keywords—Model-Checking, Virtual Commissioning, Programmable Logic Controller, Industry 4.0

I. INTRODUCTION

The Factory of the Future is a generic concept that is part of a general awareness of the importance of manufacturing industry for nations' development. This reflection is intended to maintain and develop, a strong industry, generating wealth and job creation. Hence, the Factory of the Future has to take into account several simultaneous transitions: energy, ecological, digital, organizational and societal. Factories have to transform themselves to become more sustainable industry and more respectful of the Earth [1]. Hence, the industry has entered a phase of big change that sees digital technologies as a key factor for the future to design Cyber-Physical Production Systems. These systems are predicted to enable new automation paradigms and improve plant operations in terms of increased effectiveness in facilities. Virtual commissioning, process simulation and techniques like model checking or formal methods are key components of the research agenda for making safe Factory of the Future a reality.

This paper presents 2 advanced tools which could be useful and used by the automatic control engineers in the future: model checking and virtual commissioning. Both methods require models of the plant at different levels of abstraction. They are complementary. The idea is first to check offline formally some properties of a model of the controller and secondly to use a digital twin to perform virtual commissioning in order to check on-line the PLC program.

However, all these methods considerably change the work of the engineer. It is shown how these advanced methods can be used and can considerably modify the work of the automatic control engineers in the future. To make Industry 4.0 a success, it is necessary to take into account the human component and to propose human adapted methodologies to design safe controllers.

The next section of the paper deals with the principle of verification by Model-Checking whereas section III talks about the modelling problem. Section IV presents virtual commissioning as an on-line tool in order to check on line the PLC program. A pedagogical example illustrates the paper in section V before to conclude and propose some work prospects in section VI.

II. VERIFICATION BY MODEL-CHECKING

In automated systems, most of accidents that occur in industry have been discovered to be the result of Programming errors [2]. Therefore, verification of Programmable Logic Controller (PLC) program before its implementation remains a very important task during an automation project and is a "hot topic" of the Factory of the Future. This verification must concern both functional and safety parts and not only for the equipment but also for human operator. In the proposed approach, the first verification ensures that the PLC programs meet the functional specifications, and the second one consists in verifying if the controlled system can be or not exposed to dangerous states leading to human and equipment damage. Nowadays, some verification and validation techniques like tests and simulation are available by using a recipe book for example. The verification consists in executing manually each instruction contained in the recipe book during factory tests, and then comparing the obtained results with the expected ones. This verification is therefore not automatic, and requires too much time because of the length of tests (sometime more than 100 pages of instructions). Moreover, the problem of these techniques is that they are focalized on the functional part of the system but are not exhaustive in terms of safety and there may exist a dangerous untested condition. Tests and simulation are not efficient to check formally safety part of PLC programs insofar as it verifies only if PLC programs meet the requirements specifications.

To solve this trouble of completeness, Model-Checking (MC) approach can be used in an off-line step. This concept

involves in verifying a property on all the branches of a model of the system (or of the PLC code). Our methodology consists in an offline verification of several properties of PLC programs through the model-checker Uppaal [3] on a digital plant. These properties can concern both the structure of the code and the safety.

Generating codes from models has been an industrial objective for many years. However, this phase of a project can be realized if a high and generalized model of the system exists. There exist some modelling approaches like MDD (Model Driven Development) and UML (Unified Modelling Language). Model Driven Development (MDD) is a software engineering approach that uses model to create a product. This approach is sometimes used interchangeably with model-driven engineering, and may refer to specific tools and resources, or a model-driven approach. UML is a widely-used modelling language in the field of software engineering. Experts use UML to analyse, design, and implement software-based systems, along with other business processes. Each of these modelling approaches proposes tools for automatic generation like MDA architecture (Model Driven Architecture) from UML, and DSM architecture (Domain Specific Modelling) from MDD.

The problem to use these approaches on manufacturing systems or Cyber-Physical Systems (CPS) is that these systems are complex and try to be progressively flexible. This flexibility implies to comprehend the system and a suppleness of the automation engineer.

From the specifications, the first step of the engineer is to formally model it with adapted tools such as GRAFCET or Petri Nets [4]. This abstraction is then translated into a PLC language (IEC 61131-3) for implementation [5].

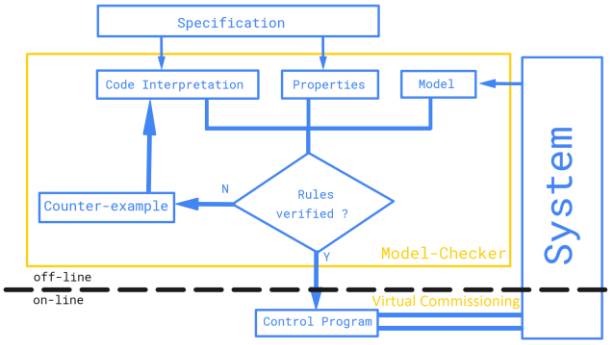

Fig. 1. Model-checking approach

Firstly, authors propose to interpret the GRAFCET specification to check off-line some syntax and structural properties by Model-Checking. For example, GRAFCET model must respect some rules. (i) One syntax rule (Alternation step/transition transition/step) and (ii) five evolution rules (on the initial situation, validated/fireable transition, activated and deactivated step…). All details about GRAFCET are described in the IEC 60848 standard [6].

Secondly, safety properties are identified by an expert through a dysfunctional analysis (using a FMEA for instance) to define all the dangerous states. For this, the behaviour of the plant must be modelled to be an input of the model-checker.

In both cases, if a rule (property) is not satisfied, the model-checker can provide a counter-example to help the operator into its program correction (fig. 1).

III. Modelling For MC

A. Code Interpretation

A PLC program is classically defined in an IEC61131-3 language. However, a model-checker is an off-line tool which cannot receive such language in input but an interpretation of it. To be considered as an input of the model-checker Uppaal, the GRAFCET specification is translated into a textual language resembling Structured Text language. For this, the classical translation of the GRAFCET called self-holding programming (or self-maintaining circuit) is used [7] (Fig. 2).

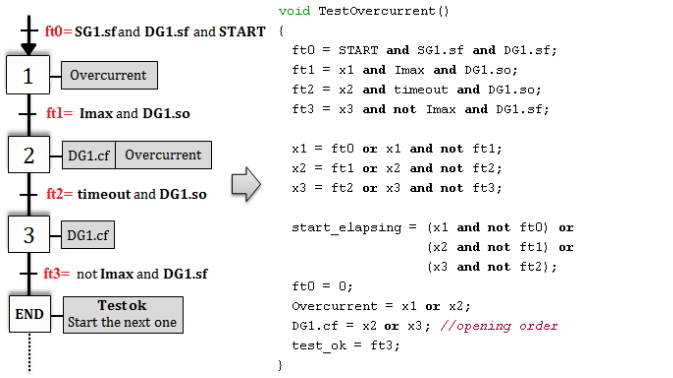

Fig. 2. Extract of the GRAFCET translation into Uppaal program.

B. System Modelling

System modelling requires careful structural analysis and PLC programs because the models must have the same behaviour as real system. If orders are sent by the PLC program, events like sensor change or operator interaction can occur whenever on system. Therefore, authors define a model that generates randomly sensor event (Fig. 3). Thereby, sensor can take two possible Boolean values (**true** or **false**).

Thanks to this model, all the reachable states of system are browsed and studied. And it can verify not only the safety part but also the functional correctness of PLC programs, according to the set of properties.

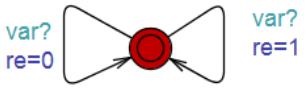

Fig. 3. Uppaal model of boolean input.

Moreover, some events must be tested on the rising edge (RE) or falling edge (FE). Consequently, a new model is defined to observe on a PLC cycle this change of value (fig. 4).

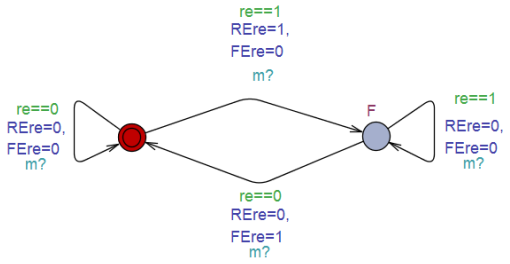

Fig. 4. Uppaal model of a falling and rising edge.

On these 2 models, we can remark that 2 synchronous events (*var* for fig. 3 and *m* for fig. 4) are present (symbol ?). These events permit to introduce a synchronization evolution during the evolution of the PLC cycle that it must be modelled.

C. PLC Cycle Modelling

The model in fig. 5 synchronizes all the other models of the system thanks to broadcast channels (symbol!). PLC cycle is modelled and structured as a loop. Initialisation of the program and parameters is required during the first PLC cycle. Then, the cycle is composed of 3 committed steps (symbol ©):

- Input reading (synchronization by var);
- Edge reading (synchronization by m);
- Execution of main program (GRAFCET function update) followed by output writing.

To reduce states space, most of states are declared as committed, so that time can elapse only during program execution. Therefore, the duration of input acquisition and output emission is negligible.

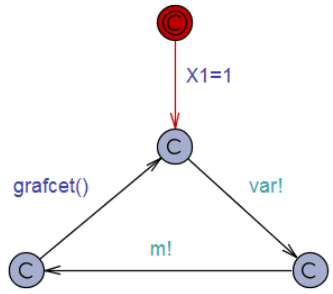

Fig. 5. Uppaal model of the PLC cycle.

D. Properties definition

The query language of Uppaal Model-Checker to specify properties is a subset of CTL (computation tree logic) [8]. Path formulae can express reachability, safety and liveliness properties through several syntaxes (E: possibly exists a path, A: Inevitably for all paths, < >: some state in a path and [ ]: all states in a path) and keyword as *deadlock*.

One can note that the use of Model-Checking techniques modify considerably engineer's work. It is absolutely necessary to think about the way to integrate them in the controller design work-flow. It seems possible for instance to use the trace of the model-checker to give information about the errors. Human-Machine cooperation tools have to be designed for that.

IV. VIRTUAL COMMISSIONING

Virtual commissioning [9] is a process which allows a comprehensive evaluation of production systems before performing physical commissioning. The programmable logic controller (PLC) code can be debugged before using it in a real production system. A growing number of companies has recently started taking interest in this technology as it reduces the time and cost of introducing new products and different scenarios can be performed to validate the manufacturing controllers in the virtual environments prior to the physical commissioning. Virtual commissioning is based on a real time plant simulation. It seems possible to use virtual systems to do virtual commissioning even if theoretically 3D high quality rendering is not necessary.

Virtual commissioning can be seen as a part of Product Lifecycle Management (PLM) which is an integrated and information driven approach to all aspects of a product from its design inception through its manufacture, deployment and maintenance [10]. PLM provides a product information backbone for companies and their extended enterprise which can be used in different applications as shown fig. 6.

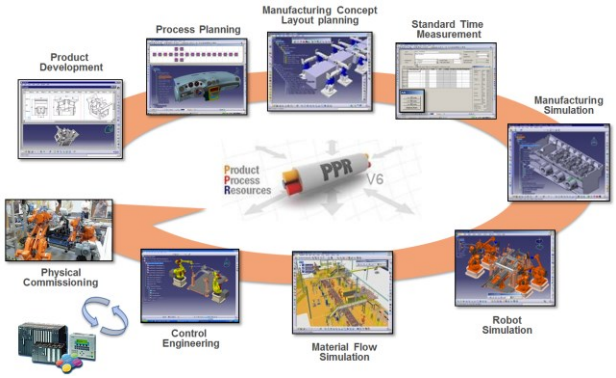

Fig. 6. The phases of PLM project from DELMIA (Dassault Systèmes)

Virtual commissioning can bring several benefits:

- Check the equipment meets the required cycle time,
- Evaluation of PLC code,
- Rapid development of standards.

There are already several simulation software to manage virtual commissioning. However, the proposed solutions are often very expensive and not so easy to use. In other words, virtual commissioning is an interesting solution but today it is risky for SME (Small Medium Enterprise).

It seems that video games technologies can become a solution to get, an easy to use and not so expensive solution for virtual commissioning. However, that involves some requirements for virtual systems.

The physics (objects, actuators, sensors) has to be certified. In other words, the use of video game physics engine can be

seen in a way as a black box. In the case of virtual commissioning applications, it is necessary to guarantee the behaviour of the simulated plant. The progress in the field of physics engine will enable in a near future to extend the possibilities of simulation (temperature, smoke, liquid…). Hence, the validation of virtual systems will be a necessary stage.

The possibility of designing their own training scenarios: hence, it would be interesting if future training software could include a library of industrial machines, parts and devices from where educators could develop their own training plants. Also, interesting would be that the developed plants could be exported and imported allowing trainers to exchange this way their customized training scenarios.

Complex plants with dozens of I/O points are usually required for conducting PLC programming and control applications in industry. Virtual systems have to become synthetic plants which must be modelled as distributed systems, allowing users to visualize and interact with different sections of the plant in different computers or on different screens.

The inclusion of a larger variety of sensors and actuators in the next generation of synthetic environments is another understandable requirement for most training applications in future factories. For instance, although binary sensors and actuators are very common in industrial applications, industrial analogue sensors and actuators are often found in practical applications. Today, the available synthetic systems do not include applications exchanging analogue data with the controlling PLC.

One other requirement is to include models of modern automation devices that communicate with PLCs through different ports (serial or Ethernet for instance). For example, one requirement often coming from PLC training in the industry is for a trainee to program a PLC and later connect it to the serial port of a PC, watching then the PLC exchanging data with a synthetic bar code reader or an AC motor driver, exactly in the same way as it would with a real, but expensive and fragile, similar device.

Soft PLC and PLC emulation software packages become common in PLC programming. Another requirement would be the connection to soft PLCs. Virtual systems, because they integrate a plant simulation, can also be an interesting tool for human operator training using Human-Machine Interface (HMI). However, it is necessary to develop specific tools and methods to use virtual commissioning. Indeed, virtual system is just a simulation. It seems necessary to design specific Human-Machine tools enable to cooperate with the engineer and to supply explanations about the encountered errors and problems.

V. Illustration on a Pedagogical System

To illustrate our previous sections, authors propose to use a sectional garage door (bottom of fig. 11) as a pedagogical study. This system is composed of elements presented in Table I. The specification is classical and the garage door is considered with no inertia. The remote control button is used to open, close or stop the movement of the door in function of its previous statement. However, during the closing, if the infrared sensor detects an obstacle, the door must be stopped. A student proposition is shown on fig. 7.

Starting from this specification, some properties are firstly checked off-line before to implement the PLC code on a virtual model.

TABLE I. Input/Output of the Garage Door

Input	***Output***
os: open sensor	OP: Open order
cs: closed sensor	CL: Close order
ir: presence infrared sensor	
re: remote control button	

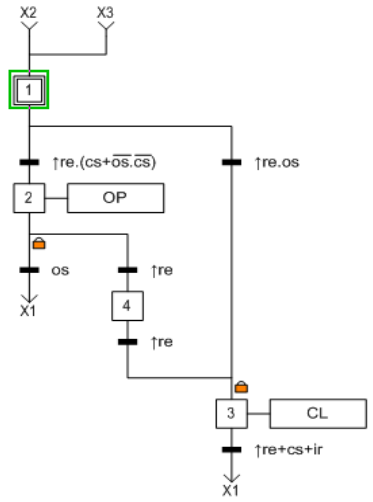

Fig. 7. Student GRAFCET.

A. Off-line verification by Model-Checking

The interpretation of the GRAFCET into self-holding programming method for Uppaal is given fig. 8. Simulation models on Uppaal Model-checker is presented fig. 9. Boolean inputs information of *os*, *cs*, *ir* and *re* are modelled by 4 models. GRAFCET used only one rising edge on the remote control then there is only one model of *re* edges. The PLC cycle model is initialized on step X1. Several properties can be checked as:

- E<>deadlock: There is a path with a deadlock state? (response: property not satisfied);
- E<> OP and CL: Is it possible to have steps OP and CL activated? (response: property satisfied)

We can note that the second property identifies the possibility to have 2 opposite orders (OPEN and CLOSE) in a same PLC cycle. This behaviour is not safe and can cause disturbances or default. To inform the designer of the trouble, a diagnostic trace is used to return a counter-example of the

property and replay the simulation (fig. 10). Thus, user can analyse the trace and finds the origin of this problem comes only when the step X2 is active and the input vertex is (*os*, *cs*, *ir*, *re*) = (1, 0, 1, 1). This "bad" proposition allows to switch on the remote control *re* in a same PLC cycle that the change value of the opening sensor *os*. To solve it, the transition condition between X2 and X4 must be *rising edge of re and not os* to be sure about the exclusivity with the transition between X2 and X1.

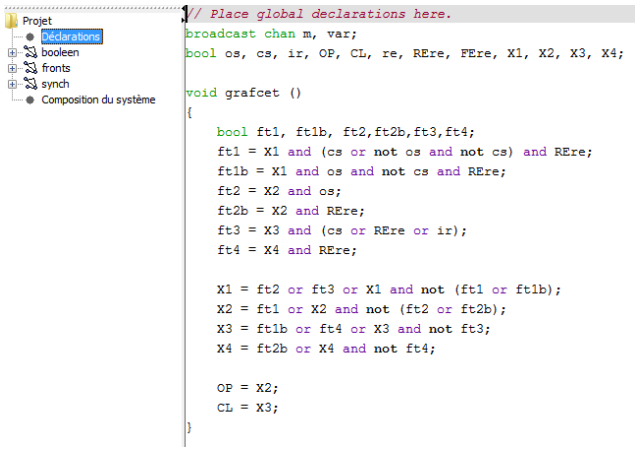

Fig. 8. Uppaal interpretation of the GRAFCET fig.7.

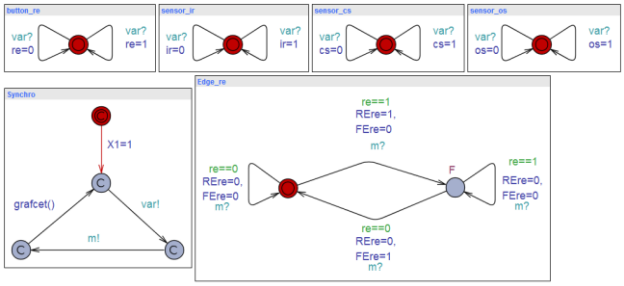

Fig. 9. Uppaal models of the garage door.

The functional part must be then checked thanks to Virtual Commissioning.

B. On-line verification by Virtual Commissioning

The use of technologies based on videogames (Graphical and audio rendering, interactivity and attractiveness) is seen as a means to promote the «Awareness of the situation». Computer simulations can be used in order to reduce risks, costs and optimize time spent on the acquisition of experience. The conception of a digital tool is the vision that has led to the development of software in a scientific and technical cooperation between the CReSTIC of the University of Reims Champagne-Ardenne (URCA) and the Portuguese company Real Games (www.realgames.co), since 2008.

HOME I/O (https://realgames.co/home-io/) is the result of a 3-year research and development project called «DOMUS» (2011-2014) between the CReSTIC lab and Real Games which was partially founded by the French Ministry of National Education, [11]. HOME I/O is real time simulation software (fig. 11) of a smart house and its surrounding environment [12]. This software was built to cover a large spread of educational applications in technology and engineering sciences. HOME I/O was built to study the house with different points of view (automation, energetic efficiency, smart home…), in its entirety or in subsystems.

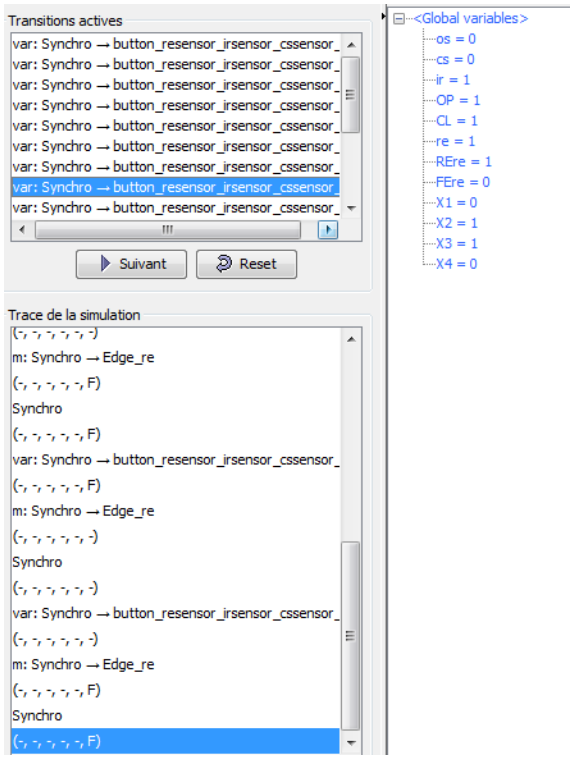

Fig. 10. Diagnostic trace with variables evolution.

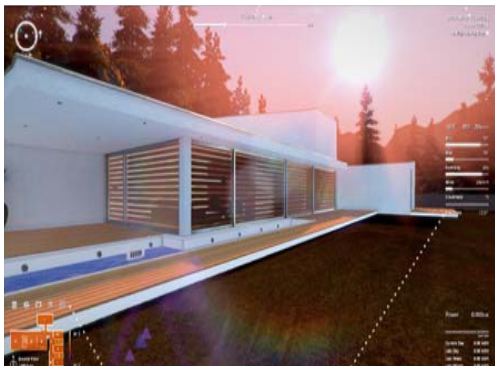

Fig. 11. Illustrative pictures of HOME I/O

To control our garage door, students propose a solution implementation on a Schneider Electric PLC. This proposition is programmed in Ladder Diagram language (IEC 61131-3) which is, may be the most popular language due to its correspondence with electrical schemes. The program under Unity Pro software is shown in fig. 12.

By a Modbus TCP/IP communication between HOME I/O and the simulate PLC of Unity Pro, the Virtual Commissioning can be used. Operator can test several scenarios thanks to a recipe book for example. A scenario is to stop the opening of the door before is extended sensor by a first press hold on the remote control (from X2 to X4). The second press hold on re must then induce the close of the door (Step X3). However, thanks to the Virtual Commissioning, student can note that the door is always in a middle position. Moreover, a third press causes an opening movement. After an analyse, operator should be able to solve their program and see that on the line 6 of the LD program (Fig. 12), a rising edge is missing on *re*.

In addition, one can note that the transition condition between X1 and X2 can be simplified to *rising edge of re and not os*.

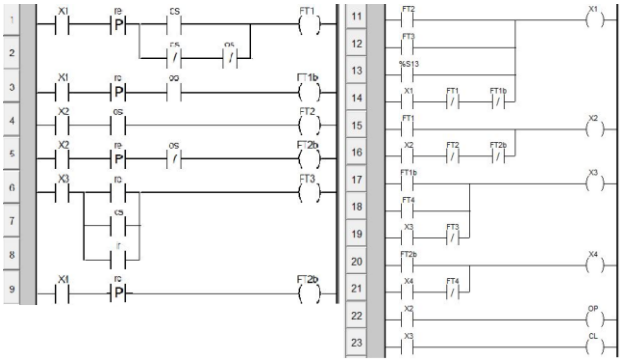

Fig. 12. Ladder Diagram on Unity Pro (Schneider Electric).

VI. Conclusion

This paper proposes to Control engineer to use Model-Checking for an off-line verification of structural properties of a specification and Virtual Commissioning to test the functional part on-line. These 2 tools permit to solve problem before real implementation on the system. It is shown how these advanced methods can considerably modify the work of the automatic control engineers in the future. It is why, the area of control education, training and outreach has to evolve in order to be adapted to the requirements of the Factory of the Future and more generally to our society. It seems to be a very promising new field of Human-Machine Systems research.

In addition, some other advanced tools can be used in the future. Indeed, standardization and the qualimetric analysis of the PLC code have to be developed. For instance, a tool like PLC Checker [1] proposed by Itris Automation Square, automatically analyses PLC programs and verifies, in an exhaustive way, their conformity with generic rules (ISO 9126 [13]). This standard describes the requirements of: (i) Readability (comments, variable naming), (ii) Reliability (all inputs are read, all outputs are written, all defects are evaluated, all sections are present and in the specified order), (iii) Modularity (no dead code and uncalled subroutine, variables are properly handled and in the right place).

A set of good practice programming rules is complementary to Model-Checking and Virtual Commissioning. It can be for instance found in The PLCopen® coding guidelines (www.plcopen.org). It covers the naming, comments and coding practices to improve the quality and consistency of a PLC programs.

[1] www.automationsquare.com/plc-checker.html

Acknowledgment

The research presented in this paper has been carried out in the context of the HUMANISM N° ANR-17-CE10-0009 research program, funded by the ANR ‘‘Agence Nationale de la Recherche”.

References

[1] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. Murray, H. Nijmeijer, T. Samad, D. Tilbury, P. Van den Hof, « Systems & Control for the future of humanity, research agenda: Current and future roles, impact and grand challenges», *Annual Reviews in Control 43 (2017) 1-64.*

[2] H.B. Mokadem, B. Bérard, V. Gourguff, O. De Smet and J.M. Roussel, « Verification of a timed multitask system with UPPAAL », *IEEE Transactions on Automation Science and Engineering, Institute of Electrical and Electronics Engineers, 7 (4), pp.921-932, 2010.*

[3] G. Behrmann, J. Bengtsson, A. David, K.G. Larsen, P. Pettersson and W. Yi, « Uppaal implementation secrets », *7th International Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems, Oldenburg, Germany, 9 -- 12 September 2002*

[4] H. Alla and R. David, « Du GRAFCET aux Réseaux de Petri, 2nd édition », *Hermes Science Publication, 1992.*

[5] IEC 61131-3 (2003). Programmable controllers – Part 3: Programming languages, Reference number CEI/IEC 601131: 2003.

[6] V. Carré-Ménétrier and J. Zaytoon, « GRAFCET: behavioural issues and control synthesis », *European Journal of Control, vol. 8, n°4, p.375-401, 2002.*

[7] William S. Levine. « The Control Handbook, Second Edition ». *CRC Press, December 2010. ISBN 9781420073669*.

[8] E.M. Clarke, E.A., Emerson, and A.P. Sistla, « Automatic verification of finite-state concurrent systems using temporal logic specifications », ACM Transactions on *Programming Languages and Systems 8 (2): 244–263, 1986*.

[9] A. Heidari and O. Salamon, « Virtual Commissioning of an Existing Manufacturing Cell at Volvo Car Corporation Using DELMIA V6 », *Master's Thesis, CHALMERS University of Technology, Goteborg, Sweden 2012, Report No. EX023/2012*

[10] W. Kuhn W, « Digital factory – integration of simulation rom product and production planning towards operative control », *20th European conference on modeling and simulation, ECMS 2006, Germany, 2006*

[11] M.J. Mayo. « Video Games: A Route to Large-Scale STEM Education? », *Science. 2009 Jan 2; 323(5910):79-82. doi: 10.1126/science.1166900.*

[12] B. Riera, F. Emprin, D. Annebicque, M. Colas, B. Vigario. « HOME I/O: a virtual house for control and STEM education from middle schools to Universities ». *11th IFAC Symposium on Advances in Control Education ACE 2016, Bratislava (Slovakia), 1-3 June 2016.*

[13] ISO/IEC 9126 Software engineering — Product quality was an international standard for the evaluation of software quality, 2001