MODEL-BASED VERIFICATION OF I&C SPECIFICATIONS

Maxime NEYRET - Thibault LEMATTRE
EDF Lab
6 quai Watier – BP 49 – 78401 CHATOU Cedex - FRANCE
maxime.neyret@edf.fr; thibault.lemattre@edf.fr

Gaëtan ROBIN
EDF Lab
6 quai Watier – BP 49 – 78401 CHATOU Cedex - FRANCE
gaetan.robin@edf.fr

ABSTRACT

In the field of nuclear I&C (Instrumentation and Control), the late detection of errors in the specifications and the design of I&C has a strong impact in terms of time and money. The early verification in the V cycle of I&C functions is a key issue to capture inconsistencies and errors before the implementation and on-site tests.

EDF, in the framework of the collaborative project CONNEXION, has investigated methods based on simulation to formalize I&C requirements and to evaluate them at the conceptual and basic design phases. Behavioral models of the physic parts of the system and I&C functions, described as functional diagrams, are co-simulated in an integration environment based on the FMI (Functional Mockup Interface) standard. In this specific environment, test scenarios are performed on these models. These scenarios are defined on the basis of a high level model of the studied system that takes into account the states to cover (in normal and degraded situations). Requirements, once formalized, are translated into observers that evaluate their fulfillment during the simulations. Simulation logs are then a tool for the designer to analyze the causes of the possible non-fulfillment of requirements and to fix the corresponding errors in the specifications.

The proposed methodology enables to detect errors in the specifications of I&C functions, to generate automatically functional test scenarios taking into account the coverage of the requirements and all the different operational modes of the system.

Key Words: I&C – V cycle – MBSE (Model-Based System Engineering)

1 INTRODUCTION

This article presents the results of the R&D (Research and Development) project “CONNEXION” lead by EDF about Systems Engineering applied to the verification of I&C functional specifications. The scope of the study concerns I&C functions at the automation level (the article does not address the supervision functions) for a single plant system. The methodology proposed in the project relies on an enhanced V cycle including complementary verification steps which do not appear in standard V cycle, as used in the nuclear field. Fig. 1 highlights these new steps (in green).

For the design of I&C functions, nuclear engineering currently relies on three main tasks in the V cycle: a textual expression of the functional requirements which are gathered in documents written for each plant system. These documents are based on a common structure with compulsory chapters; as they are written manually, the may contain inconsistencies or errors. The second step is the production of detailed diagrams formalizing I&C functions. These diagrams are used as input data for the programming of the

I&C platform, which constitutes the third task of the “top-down” branch of the V cycle. Thus, the detail level of these diagrams is very near of the realization.

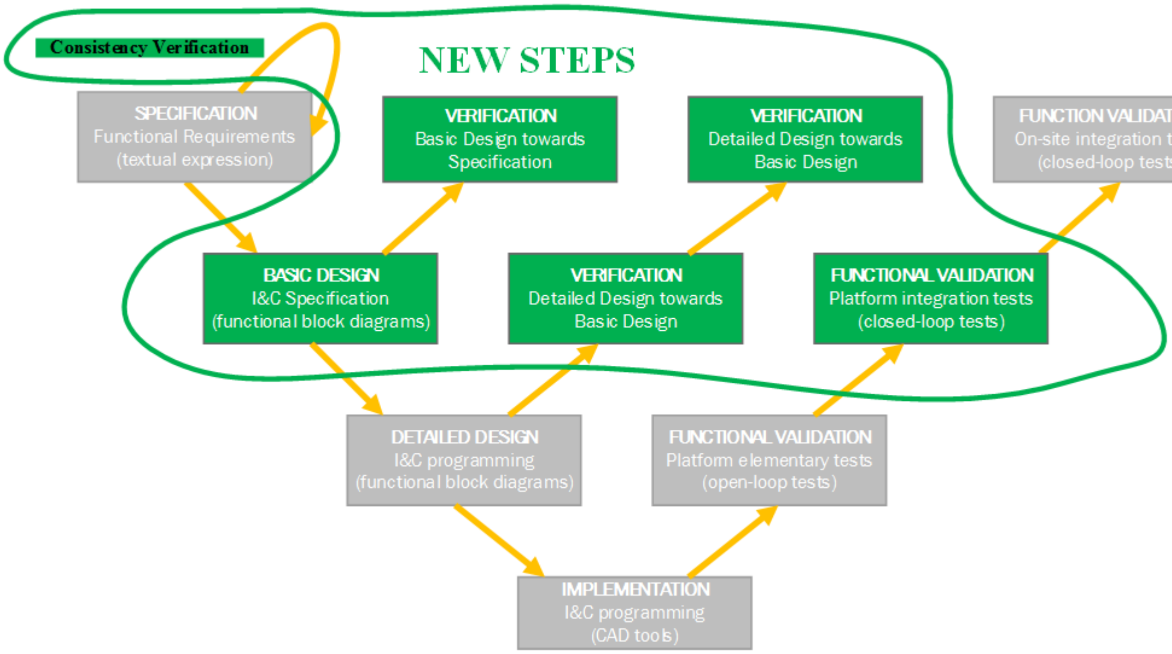


Figure 1. Enhanced V cycle with complementary Verification steps.

The enhanced V cycle distinguishes two different levels of detail in the design phases (basic and detailed design), thus allowing to perform model-aided verifications earlier. Basic design specifications include the main I&C functions expected, at a high level of description. It includes the main regulation, automatisms, but does not precise the detail of implementation. Auxiliary functions (which can be considered as technical functions linked to the technologic choices of realization), such as low-flow circulation lines, redundant sensor measures voters, etc. are not in the scope of modelling. The standard V cycle is also enhanced by three phases of verification (in green) in order to capture defaults as early as possible in the design part of the cycle. Another verification activity is added at the very first step of the V cycle in order to check the consistency of the corpus of requirements.

The key idea of the methodology is to reinforce the early phases of the V cycle in order to avoid wastes of time and money due to late detection of errors during platform and on-site tests. These verifications are based on simulation at different levels of details adapted to the different levels of maturity in the design steps.

2 VERIFICATION OF TEXTUAL REQUIREMENTS

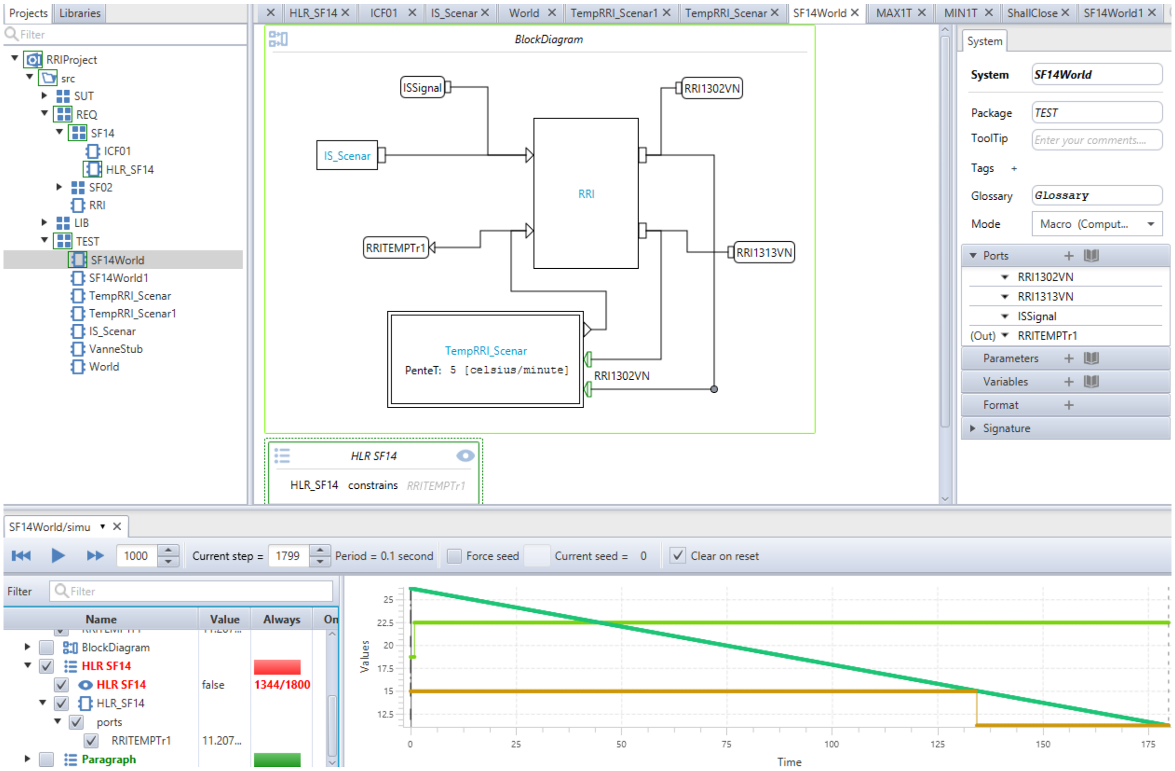
In the first step of the V cycle, I&C requirements of a single plant system are written in literal text, possibly by different experts. The non-automated feature of this activity induces possible errors. That is why, the proposed methodology adds a tool-aided verification at this level. Thus, it contributes to avoid propagating errors throughout the V cycle.

STIMULUS is a tool developed by the French company Argosim [1] which allows to formalize requirements using configurable libraries of expressions (logic, mathematic, temporal, etc.) and configurable data models. The user can develop objects involved in the requirements (sensors, actuators, hardware components, etc.) and logics associated to literal expressions. For instance, the user can declare the expression “the valve shall open” as a tag for the evaluation of the command sent to the valve. Even if STIMULUS formalizes requirements, the interpretation of the requirements models is easy for non-expert people thanks to the use of these tags.

STIMULUS also permits to generate test scenarios on the basis of the requirements. The formalized requirements are then translated into observers which can be plugged to a simulation of the system. These observers evaluate during the simulation whether the requirements are fulfilled or not.

Once the requirements are formalized, test scenarios are performed on the corpus of requirements. The results of the tests possibly highlight inconsistencies between two or more requirements. It can also identify operating ranges for which the behavior of the I&C has not been specified, thus allowing unexpected and unwanted operation situations. This is a help to enhance the relevant requirements in order to fill these lacks.

Fig. 2 shows in the upper-left window, the tree structure with the formalized requirements in the “REQ” folder. The requirements in the displayed example are classified in “I&C Functional Requirements” (ICFR) which contribute to the fulfillment of higher-level requirements i.e. Safety Features (SF). The upper middle window shows the test environment of these requirements: a scenario (based on random stimuli with constraints in order to remain in physical values) is applied to the inputs of the functional requirements. The High Level Requirement (HLR) is formalized as an observer which evaluate the fulfillment of the requirement during the tests. The bottom left window shows that this requirement has been violated during the tests (it is red highlighted) and the bottom right window points on the chronogram when the requirement violation occurred.



PenteT: 5 [celsius/minute]

System: SF14World

Package: TEST

ToolTip: Enter your comments...

Glossary: Glossary

Mode: Macro (Comput...

HLR SF14

HLR_SF14 constrains RRITEMPTr1

1000 Current step = 1799 Period = 0.1 second □ Force seed Current seed = 0 ☑ Clear on reset

Name	Value	Always	On
□ BlockDiagram			
☑ HLR SF14		🟥	
☑ HLR SF14	false	1344/1800	
☑ HLR_SF14			
☑ ports			
☑ RRITEMPTr1	11.207...		
□ Paragraph		🟩	

Figure 2. STIMULUS environment with results of consistency test on requirements

To reach this situation, the tool has generated stochastically input profiles (within the physic limits which have been defined as constraints) until the requirement is violated. If the tool does not identify an input profile that violate the requirement, it is not guaranteed that the requirement can never be violated. However, the use of STIMULUS is considered as a help for verification and not for qualification. In the example displayed on Fig. 2, the analysis shows that an I&C functional requirement concerning a temperature regulation was conflicting with the high-level requirement which states that the temperature of the system must remain above a fixed threshold. Such inconsistencies would have been hard to detect by a non-automated review for instance if the requirements are not written in different chapters.

3 REQUIREMENTS FORMALIZATION

Before the performance of the verification tests, observers are created in order to evaluate the fulfillment of the requirements. These observers are a formalization of the requirements into an executable code, which we call "requirements model". They are connected to the behavioral models of I&C functions and the controlled process with the FMI standard via an exchange table of the inputs and outputs of these models. They evaluate during the tests the fulfillment of the requirements.

The tool used is ARTiMon (CEA / LIST) [2]. It can be used to formalize requirements with the support of libraries of expressions. The broad scope of these libraries allows to express all kind of requirements met in the specifications of I&C functions: logical, temporal, analogic, etc.

Most of the requirements are simple logical expression concerning the belonging of a physical parameter to a specific range. That is the case for the presented study case with the water temperature of the circuit which has to remain between 15° and 38° Celsius in operating modes of the system. In this case, the formalizing of the requirement uses analog thresholds.

More complex requirement may also be expressed thanks to temporal operators (including even-triggered expressions and conditions), such as "while", "as soon as", "during". An example of such requirement has been found in our study case: it states that the water circulation function of the system may not operate with only one of the three redundant pumps active, during more than two seconds.

ARTiMon is an alternative to STIMULUS (cf. Paragraph 3) which can also generate and export requirement observers which can be plugged to behavioral models. ARTiMon has the advantage to be FMI (Functional Mockup Interface) [3] compliant. Consequently, it has been integrated naturally to the other models used in the proposed methodology. However, ARTiMon does not provide any tests generation function.

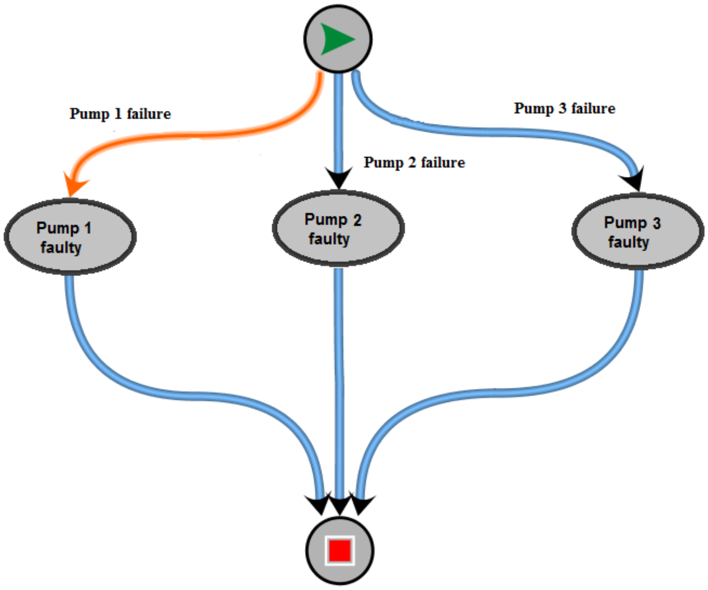
4 TEST GENERATION

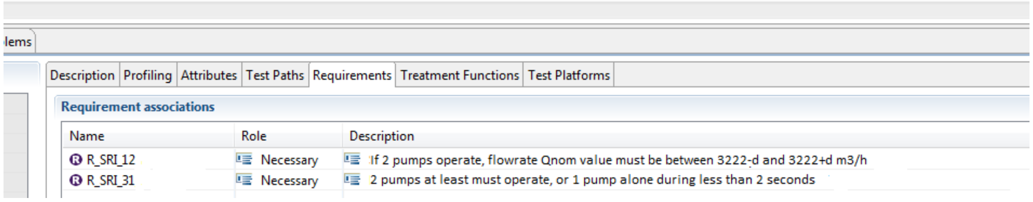
The goal of the test generation is to meet the full coverage of the requirements, that is to say, every requirement must be tested at least once. Whereas usual V&V processes rely on tests specified manually by independent experts, the proposed methodology uses a tool-aided approach insuring that every requirement is taken into account during the test generation.

The different operational modes of the system are first of all listed and formalized into a state-transition diagram which synthesizes all the operating modes to take into account and the events (including elementary failures) allowing transitions between these modes. In our test case, the elementary failures are the loss of each actuator (pump), the loss of a complete cooling line, a leak in the water-feeding tank. Deeper causes of failures (such as power loss, mechanical problem, maintenance shutdown) are not relevant for the modeling, since the process model does not reach this level of detail. In parallel, the requirements are extracted from the textual specification documents and listed into a table with different attributes; among them, the operational modes in which the requirement is relevant are listed.

Fig. 3 shows an example of a part of the testing model, based on MaTeLo by All4Tec [4], for a system including three redundant pumps. The model has several levels of detail: a top level tree is composed of several branches corresponding to the sub-functions (in our example, a cooling function, a water-circulation function, a water-feeding function). For each sub-functions, a tree represents the possible events which can affect the mechanical components on which the function relies. Fig. 3 displays the sub-tree corresponding to the water circulation sub-function. The identified events are linked to the loss of the pumps on which the function is implemented. Requirements are associated to the transition arrow which is integrated in the test scenarios. The small table below shows two requirements R_SRI_12 and R_SRI_31[1] associated to the selected orang arrow. It means that when MaTeLo generates a step linked to this event in a test scenario, these two requirements are evaluated.

The scope of the events must also include changes coming from the environment of the system. Indeed, it may also affect it and involve other requirements to take into account. In our case, the amount of thermal power to extract by the system is an environment parameter that affects the behavior of the system and implies to verify different requirements.





Requirement associations

Name	Role	Description
R_SRI_12	Necessary	If 2 pumps operate, flowrate Qnom value must be between 3222-d and 3222+d m3/h
R_SRI_31	Necessary	2 pumps at least must operate, or 1 pump alone during less than 2 seconds

Figure 3. Example of part of testing model

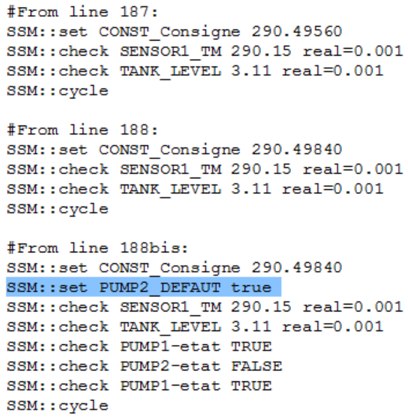
The tool then automatically generates test files which can be performed in the simulation environment thanks to the FMI standard. The scenarios are generated by exploring the branches of the testing model according to different profiles. The user may chose a systematic exploration of all the branches of the model, which guarantee to test at least once each event and each state. But it is also possible to add probabilities of passing through specific states and / or transitions in order to focus on a specific operation mode.

Fig. 4 shows an example of such test file with elementary steps composing the scenario. Each action, described by a "SET" function is associated to a simulation step expressed with number of simulation cycles. ALICES, in which the models are integrated is the master of synchronization: it triggers steps of

[1] SRI is the French acronym which refers to the Conventional Island Closed Cooling Water System.

simulation for each model, including the test scenario. The blue highlighted line shows the activation of a failure on a pump. In our case, the time of the occurrence of this event has been chosen randomly by MaTelo while browsing the testing model branches.

A test scenario may combine several events. As the consequences of an event may require a specific time response, it is necessary that all events do not occur too close in time, in order to analyze more easily the results of the tests. To ensure this, it is possible to change manually the time occurrences of the events in scenarios.



```
#From line 187:
SSM::set CONST_Consigne 290.49560
SSM::check SENSOR1_TM 290.15 real=0.001
SSM::check TANK_LEVEL 3.11 real=0.001
SSM::cycle

#From line 188:
SSM::set CONST_Consigne 290.49840
SSM::check SENSOR1_TM 290.15 real=0.001
SSM::check TANK_LEVEL 3.11 real=0.001
SSM::cycle

#From line 188bis:
SSM::set CONST_Consigne 290.49840
SSM::set PUMP2_DEFAUT true
SSM::check SENSOR1_TM 290.15 real=0.001
SSM::check TANK_LEVEL 3.11 real=0.001
SSM::check PUMP1-etat TRUE
SSM::check PUMP2-etat FALSE
SSM::check PUMP1-etat TRUE
SSM::cycle
```

Figure 4. Example of part of testing model

5 CLOSED-LOOP TESTS

The closed-loop tests performed at early stages of the V cycle enable to detect specification errors in I&C functions. Indeed, open-loop tests, as they are currently performed, can highlight low-level errors (i.e. logic or syntax errors) but are not sufficient to detect higher-level functional errors in which the check-back of the process has a strong influence in the evaluation of I&C functions.

To perform these tests, two models are first of all developed in the “basic design” phase (cf. Fig. 1): a process model and another one with the corresponding I&C functions, described with functional block diagrams. Both are interfaced in an integration environment where the requirements model is also integrated. The proposed methodology is based on FMI standard which allows co-simulation between models developed with heterogeneous tools.

Fig. 5 shows the simulation environment (ALICES by CORYS) in which the different models (process, I&C and observers for requirement fulfillment evaluation) are co-simulated and stimulated by test scenarios.

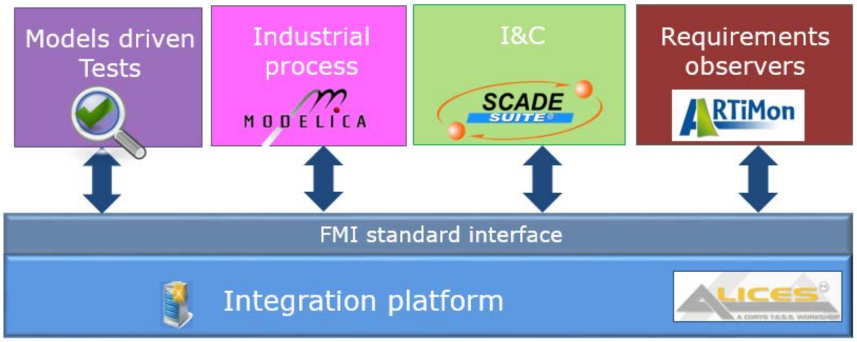


Figure 5. Integration of models in simulation environment

The choice to distinguish Basic Design and Detailed Design verification is made to meet project planning issues. Design choices concerning functions and technologies are not all available during Basic Design so that it is not possible to develop too detailed models. That is why the proposed methodology introduces such tests at the basic design and the detailed design of I&C specification (cf. Fig. 1).

Fig. 6 illustrates a view of the modelling of I&C function diagrams (Basic Design) based on SCADE (ANSYS/ESTEREL) [3]. The formalization is very close to the FBD (Function Block Diagram) I&C programming language. However, SCADE has a complementary feature which allows to order the diagrams in different hierarchic levels. Fig. 5 shows the top-level view with boxes corresponding to the sub-functions of the system and data exchanges between them. At a lower level, diagrams describe the logics and others I&C computing of each sub-function. This structure is very useful to have a functional view of the I&C.

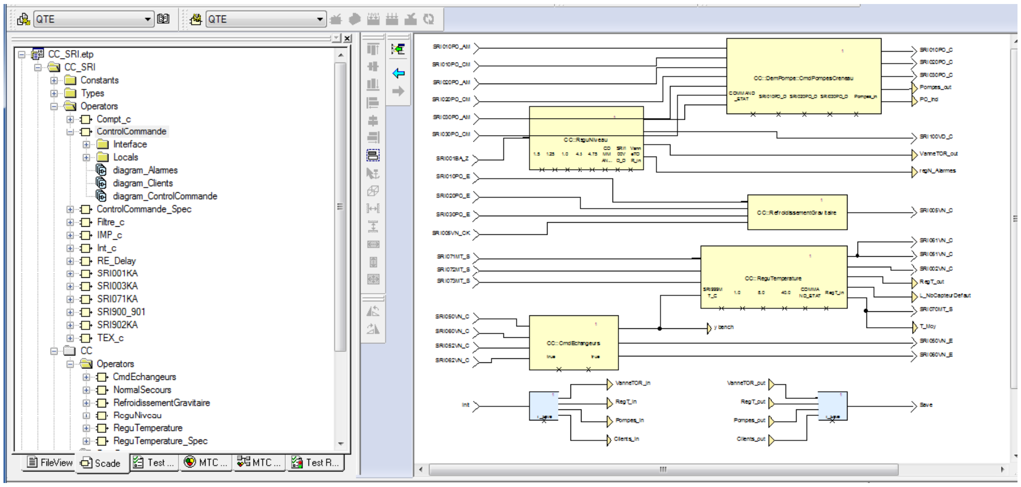


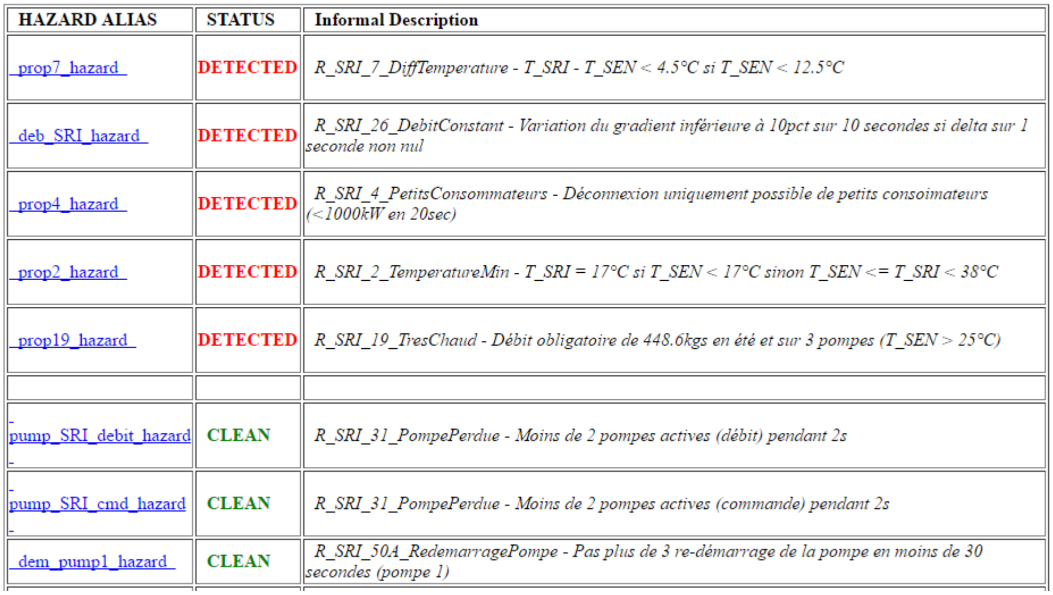
Figure 6. Example of I&C functional diagram in SCADE tool

Furthermore, the functional hierarchy facilitates the coverage analysis of the tests. Indeed, SCADE has a module to measure structural coverage of the tests once they have been all performed. For each box of the set of SCADE diagrams, it states whether the box has been tested or not. If the box has different possible states, the tool specifies which part of these states has been covered. As the diagrams are organized in a functional hierarchy, the structural coverage analysis is linked to the functional coverage. It may help to define complementary tests to cover a functional state which has not been reached.

The log generated by the observers shows which requirements have not been fulfilled and at what time of the simulation. The log generated allows experts to analyze the sequence of events which have led to the hazards. Fig. 7 illustrates the front page of such a log. It synthetizes the requirements (properties) which have not been fulfilled ("Detected" status) during the test. For each requirement, an html link gives access to details about the sequences of events for further expert analysis.

As the methodology deals with early phases of the V cycle, it is possible to use it as an iterative process. When a log states that a requirement has been violated, the analysis can also lead to the conclusion that it is due to an error in the process modelling or even in a test scenario. A step backward is therefore necessary to refine the used models. The structure of the simulation environment allows to perform easily regression tests after the corrections have been realized.

The tool-aided methodology can be considered as a back-up fort experts to examine carefully the specifications of I&C functions but also the conditions and the environment in which they are operated.



HAZARD ALIAS	STATUS	Informal Description
prop7_hazard	DETECTED	*R_SRI_7_DiffTemperature - T_SRI - T_SEN < 4.5°C si T_SEN < 12.5°C*
deb_SRI_hazard	DETECTED	*R_SRI_26_DebitConstant - Variation du gradient inférieure à 10pct sur 10 secondes si delta sur 1 seconde non nul*
prop4_hazard	DETECTED	*R_SRI_4_PetitsConsommateurs - Déconnexion uniquement possible de petits consoimateurs (<1000kW en 20sec)*
prop2_hazard	DETECTED	*R_SRI_2_TemperatureMin - T_SRI = 17°C si T_SEN < 17°C sinon T_SEN <= T_SRI < 38°C*
prop19_hazard	DETECTED	*R_SRI_19_TresChaud - Débit obligatoire de 448.6kgs en été et sur 3 pompes (T_SEN > 25°C)*
pump_SRI_debit_hazard	CLEAN	*R_SRI_31_PompePerdue - Moins de 2 pompes actives (débit) pendant 2s*
pump_SRI_cmd_hazard	CLEAN	*R_SRI_31_PompePerdue - Moins de 2 pompes actives (commande) pendant 2s*
dem_pump1_hazard	CLEAN	*R_SRI_50A_RedemarragePompe - Pas plus de 3 re-démarrage de la pompe en moins de 30 secondes (pompe 1)*

Figure 7. Extract of log showing requirements (properties) fulfillment after test

The level of detail at the Basic Design phase is such that only the functional aspects are modelled and not the technical features linked to the technology. Once the Basic Design verification are performed and the possible errors fixed, the activity is refined in the Detailed Design phase (cf. Fig. 1). The I&C and process models are enhanced to go deeper in detail and includes new features such as sub-functions (voters on redundant measures for example) or technical function linked to the technologic choices concerning the process (actuator characteristics for instance) or the I&C system (sample time for instance).

The performed verifications are the same as for the Basic Design phase. The tests are in this case considered as regression tests which has to verify if I&C functions, as they are here specified in detailed manner, still fulfill the requirements.

6 CONCLUSIONS

The presented MBSE (Model-based System Engineering) methodology is the result of the large scale R&D project lead by EDF, CONNEXION. It aims at reinforcing the confidence in the upstream phases of the standard V cycle for the production of functional I&C data. The use of simulation at the very beginning allows to capture specification errors and to avoid to propagate them in the later design phases. The gain in time and money may be significant as they avoid steps backwards in design and testing phases.

The first step of this innovative methodology can be performed as a stand-alone task since it targets only the textual requirements and does not require any behavioral model. The others verification activities are supported by different models which are all integrated thanks to the FMI standard in a unique simulation environment, from the requirement formalization to the closed-loop tests analysis.

This methodology, has capture the interest of French nuclear engineering and use cases have been initiated to evaluate the gains in the framework of operational projects. Future developments of the methodology shall include the verification of requirements concerning also supervision and control room functions.

7 REFERENCES

1. F. Gaucher, Yves Genevaux, “Validating Embedded Systems Specifications”, *ERTSS*, Toulouse, January 2016 (2016).
2. N. Rapin, “ARTiMon Un outil de monitoring de propriétés de logique temporisée” *Génie Logiciel*, Hors série mai 2014, pp.26-31 (2014).
3. T. Blochwitz et al., “The Functional Mockup Interface for Tool independent Exchange of Simulation Models”, *8th International Modelica Conference*, Dresden (Germany), 2011, pp. 20-22.
4. F. Chastrette, Luc Coyette, “Application of MBT to validation of new nuclear I&C architectures”, *UCAAT*, Paris, October 22nd-24th 2013 (2013).