M. Illarramendi, L. Etxeberria and X. Elkorobarrutia, "Reuse in Safety Critical Systems: Educational Use Case Final Results," 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, 2015, pp. 290-297.

DOI: https://doi.org/10.1109/SEAA.2015.43

# Reuse in Safety Critical Systems: Educational Use Case Final Results

Miren Illarramendi
Leire Etxeberria
Xabier Elkorobarrutia
Embedded Systems Research Group
Mondragon Goi Eskola Politeknikoa (MGEP)
Arrasate-Mondragon, Spain
Email: {millarramendi,xelkorobarrutia,letxeberria}@mondragon.edu

***Abstract*—The University of Mondragon, has participated in an European ARTEMIS project called SafeCer during 4 years. The main objective of the project has been to research about the "Reuse of safety related embedded systems and components". Mondragon University has defined an Educational Use Case in order to form the future engineers and has also conducted a first experiment with students of the Master of Embedded Systems. In this paper, the results of this experiment are shown.**

***Keywords: Reusability of SW, Safety Integrity Level, Certification of Safety-Critical Embedded Systems, Education***

## I. Introduction

Reuse of safety related embedded systems and components is one of the research challenges in the European software research community. It is important to maintain Europe as world leader in development of safety relevant systems and for that one of the keys would be to transfer the results and the generated knowledge to the new professional in this area. Having all this into account, in the ARTEMIS SafeCer project industrial, academic and scientific communities have been working together generating new methods and tools. The project has had different use cases, most of them industrial use cases, but one of them developed by the Mondragon University have been an educational one. In this paper, the experiences of Mondragon University in the Master of Embedded Systems and the overall results of the Educational Use Case are shown.

As we present in [5, 6] the University of Mondragon is a cooperative small and private university and it has several master's degrees. One of them is the Embedded Systems Master. The main objective of this master is to train professionals able to innovate, design, develop, assess and maintain products that are based on embedded systems assuring the required safety level during all their life cycle.

The Embedded Systems Master is very practical. The students take their competencies in the area of embedded systems using active methodologies and each student has to take the initiative in his/her studies and decide in which aspects they want to specialize more. There are some theoretical classes (basic concepts) and then the students have to work on practical exercises or real projects.

Part of this research was funded by the ARTEMIS Joint Undertaking, Grant Agreement 295373 nSafeCer.

Some industrial companies and research centers (eTic, Ikerlan, Orona, CAF P&A, Ulma Embedded Solutions, etc.) are also participating in the master courses giving some modules and/or defining real projects.

In this way, the master course joins industrial and academic communities and the students have the option to contribute to the industry and also have the opportunity to work with real problems.

In this context, the educational use case defined in SafeCer is very interesting in order to help in the acquisition of this knowledge by the students. The theoretical part of this type of systems (standards, methods, etc.) will be important, but having the practical aspect is also very important.

The basic knowledge of the first year student's is based on Computer Science, Telecommunications and Electronics: the new students are coming from these bachelor studies. The Embedded Systems Master is constituted by 12 different subjects and the final course project. This final project is developed on industrial companies or in a research team of the university. The Educational Use Case is based on 4 of these 12 subjects: three in the first year and one in the second year. In parallel, there are other practical use cases or practices related with other subjects and technologies like FPGA programming and Communications. The Embedded Systems' Master uses a methodology with a problem based approach.

The planning, objectives, tool framework to be used and some technical concepts of this use case were explained in the previous publications [5] and [6] . In this paper, we are going to show the results of the experience and the results obtained once the use case has been implemented.

In section II of the paper final objectives and the general description of the use case are presented. In section III we present the scenarios that have been considered in the use case. In section IV, the tool framework used in the use case is presented and in section V we will explain the final implementation of the use case. In section VI the results and the first evaluation are presented and finally in the section VII final conclusions are shown.

## II. Objectives and Description of the Educational Use Case

### A. Objectives of the Educational Use Case

As explained on [5], the main technical objectives of this Educational Use Case are to demonstrate that the reusability of SW components in Safety Critical Embedded Systems is possible and also to demonstrate the benefits of reusability (less cost, safer, reduced time to market, etc.) but there are other transversal objectives linked with the learning results of the involved Embedded Systems Master subjects which are *Reliability and Performance Analysis*, *Life cycles of Embedded Systems*, *Real Time Systems* and *Standards and Regulations*.

The use case has been designed to be implemented in two iterations:

- In the first iteration, 1st year of the Master, the system description was based on an elevator system (considered as an automatic roof control) and the applied functional safety standard was IEC 61508 [4]. The students had to do some related works in different courses as explained on [6]: They work on reliability techniques, the SafeCer Generic Process Model and Activity Patterns in the *Reliability and Performance Analysis* subject, on SafeCer Component Level and Contract Based Modeling in the *Life cycles of Embedded Systems* subject and they implemented the system in the *Real Time Systems* subject .
- In the second iteration, some of the requirements of the original system have been updated and the students have had to redesign the original system considering the reusability, activity patterns, contract based design approach and taking into account what the standards say. In this case the students have also learned about different functional safety standards (IEC 61508 [4], CENELEC 50126[1], CENELEC 50128[2], CENELEC 50129[3], ISO 26262[7],...) in the *"Regulations and Standards"* subject. The main learning result of this subject is to learn the basic concepts of the IEC 61508 [4] functional safety standard and others that are specific for domains (Railway, Automotive, Avionics,...) and the work proposed in the experiment fit very well with that objective.

As a result, the students have had an opportunity to learn and practice with new and innovative methods, tools and processes to design and develop Safety Critical Embedded Systems that in the basic bibliography and educational material related with Embedded Systems is missing. It is an extra activity that improves the knowledge of the students. In future, these students will work in industry and the European industrial net will be the final beneficiary.

So, once the experiment is finished, three objectives have been reached:

- The students have checked the benefits of the reusability and its limits in Safety Related Systems Design.
- The students have used and studied on detail one or more Functional Safety Related Standards.
- The students have used new and innovative methods, tools and processes to design and develop Safety Critical Embedded Systems.

For both iterations, some guidelines and material has been created in the context of the SafeCer project and these guidelines have been used in the basic lessons. Once the basis were clear, the students designed the system, defined its contracts and used the tools to do the V&V. For using the tools, some examples were shown in the practical classes. Regarding to the new concepts as definition of contracts, some examples were shown and the teachers' help have been needed.

### B. Description of the Educational Use Case

This is the description of an educational demonstrator aiming to be used in lectures related to safety, real-time, software engineering and embedded system development.

The goal of the demonstrator is to develop an automatic roof or an elevator system control (for simplicity, in the laboratory a mock-up for the elevator system has been implemented). Both system's are composed of 2 or more engines and they close-open the roof or lift or bring down a load in a coordinated way. Each roof-arm /elevator has attached a motor, up and down sensor, and shaft rotation sensor that is used to infer position and speed. As both system's have similar characteristics, for simplicity, we are going to consider the elevator system in the explanations of the use case.

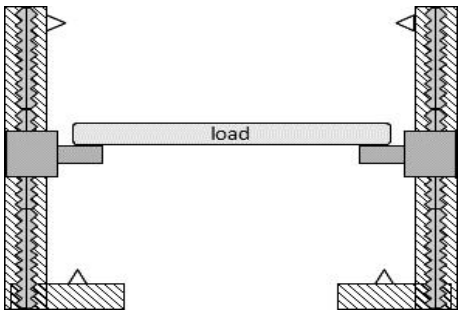


Fig. 1. System Elevator.

The use case is going to focus on software aspects and not on electromechanical ones. The analysis model of the control system is the one depicted in the next figure.

- Each elevator is controlled by an `ElevatorCtrl` software component. It reads from its sensor, actuates on its motor and announces its state to the main controller.
- All elevator coordination is in charge of `ElevatorSystemCtrl`. The one that commands all the elevators coordination response to an operator.
- The operator has an interface for commanding the system.

The main idea of this use case is to demonstrate the benefits of implementing the SW components following a Contract Based Design methodology. First, before implementation each

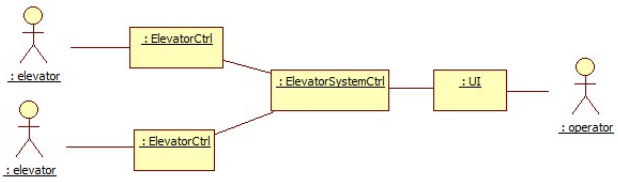


Fig. 2. Elevator control system's software components.

component specification will be used to acquire a sound foundation that the system will work; second, if those components would have been previously developed and their contracts specified, those can be used for a virtual integration and check if the collaboration of the various components fulfill system requirements.

The collaboration among the various components of the system is depicted in Fig. 3. It doesn't show much details but the main focus isn't to check the semantic and syntactic correctness of the various connections; the main focus is to check if other non-functional requirements are met.

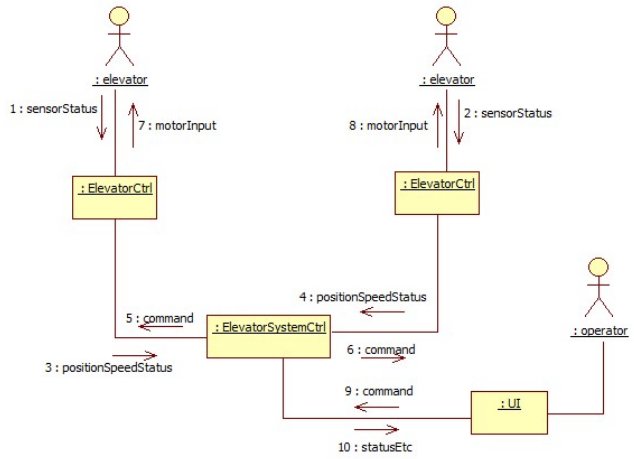


Fig. 3. Elevator control system collaboration.

Although far from a real industrial system, for it is an educational use case, the system is assigned next safety requirements:

- If one elevator stops, the others must stop within 50 millisecond.
- The difference of position between two elevators can't be greater than 10 mm.

## III. Working Scenario and Activity Patterns

### A. Implementation Scenarios

In order to show the benefits of a Contract Based Design methodology, different scenarios must be provided. Those can be different applications, different application domains, different implementations and technologies that involves new components,...

In the table I, we can see the scenarios that has been checked for implementing the Educational Use Case:

| Scenario name | Cross-certification ? | Scenario description | Goals | Main enablers/challenges |
|---|---|---|---|---|
| System evolution/ maintenance due to new system requirements | No | New requirements on the system, leading to changes. Two variants of this scenario have been identified (in many system change scenarios, we would expect both to occur simultaneously):<br>• Re-integration of existing components with some new own functionality ("glue code").<br>• The new requirement is to be forwarded to a specific component. (This is the system developer's view of one variant of the scenario "Component modification" above.) This includes communicating the new requirements to component developer, and later receive the new component (with documentation, verification results, argumentation etc.) Addition of new component. | Cost savings: *Entry not completed* | *Entry not completed.*<br>Easy re-integration, re-verification, regression testing, etc.<br>Easy re-incorporation of component safety argumentation into system safety argumentation<br>The effect of the requirement on the architecture is important to understand. Tracability of architecture changes that are the result of the requirement is needed. |

TABLE I. Educational Use Case Working Scenario.

In our case, two variants of the system implementation have been chosen: a centralized one implemented in one micro-controller (first iteration, 1st year), and a fully distributed one where each software component runs in its own processor and are connected through a field bus (second iteration, 2nd year) (see Fig. 4 and Fig. 5).

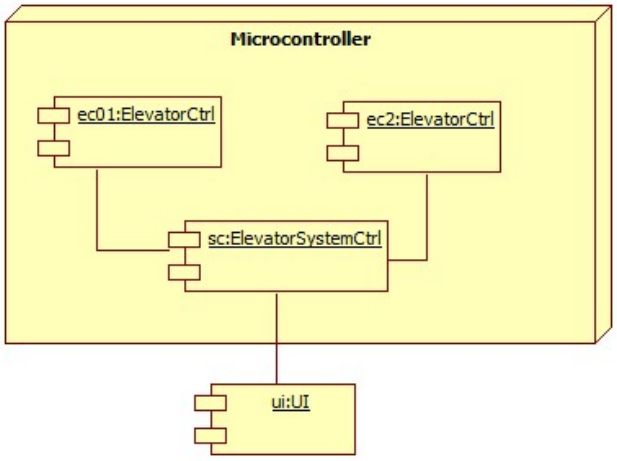


Fig. 4. Deployment in single micro-controller.

Intermediate variants can be easily derived in which some components share the same microprocessor. Even a centralized one that uses I/O modules connected by a field bus to the only microprocessor.

When using software components in different scenarios, their environment changes. In this work, component reutilization is understood as source code reutilization. Also, their environment is required to satisfy some requirements: activation frequency from the scheduler and worst response time that depends upon the processor and other possible tasks in the same microprocessor.

### B. Activity Patterns

The approach adopted by the SafeCer project builds on a number of technical items which are developed within the

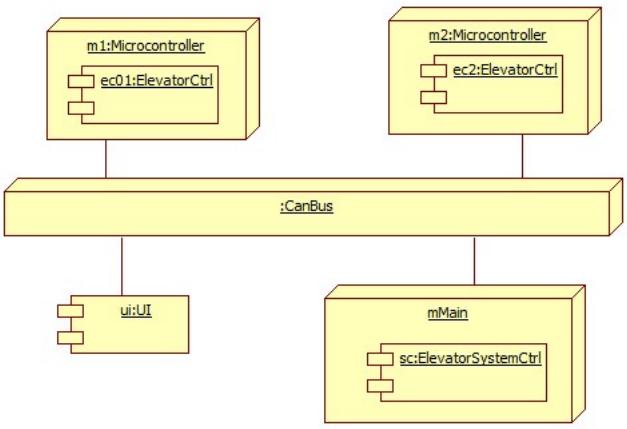


Fig. 5. Distributed implementation with a field bus

project: component contracts, safety argumentation, formal verification techniques, proactive certification documentation, tool support for traceability. While tool support is something that is present throughout the development (i.e. found in every activity), the remaining technical items represent dedicated activities that need to be performed as part of the development process. Therefore, it makes sense to add explicit activities to the pattern to deal with these items.

In a reuse and component based context, the process model needs to deal with the joining of two development paradigms: system development and component development. If the process model is to be applied for both these development paradigms (and more) it should preferably contain the same main activities although possibly with slightly different concretization. In SafeCer the activities has been regrouped and a pattern identified; this is called Activity Pattern [8].

In the Figure 6 we can see which are the Generic SafeCer Activity Patterns.

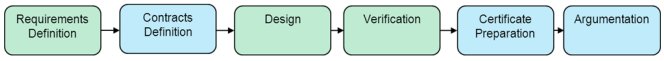


Fig. 6. Generic SafeCer Activity Patterns

Regarding to the use case, the elevator system was designed and modeled using the process defined with the Activity Patterns, but we didn't consider the Certificate Preparation and Argumentation phases because it was an academic exercise. The considered functional safety standard was the IEC 61508 [4] and the system had several SW components. Each of these components were designed and modeled by the process defined by the Activity Patterns (at Component Level) and then, the system has been generated with the composition of the components (Activity Patterns- System Level) [8].

In the second iteration, a new configuration of the system has been considered and new requirements and contracts have been specified. With these updates, the overall system has been changed and the system level requirements and contracts have to be verified and validated.

In the Figure 7 we can see which are the considered Activity Patterns at Component Level and at System Level and they are linked with the tools considered in the use case.

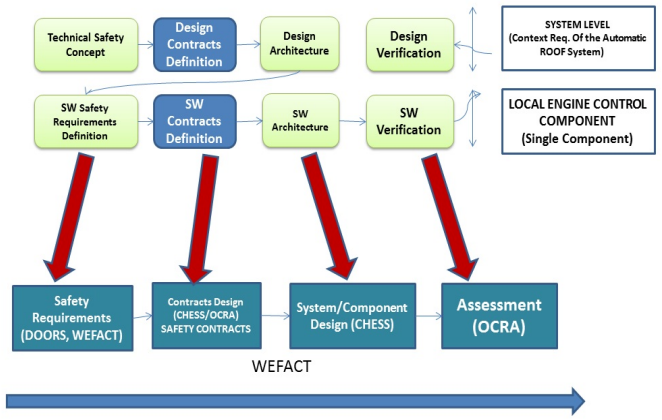


Fig. 7. SafeCer process demonstrated by the Educational Use Case

## IV. Tool Framework Used in the Educational Use Case

As described in [9] and [10], the Certification Tool Framework (CTF) is a framework collecting all the SafeCer consortium partners' tools producing evidence within the process of certification. Each tool, able to produce or manage artifacts and needed to provide certification evidence, will return one or more artifacts as output. Some of these tools have been used in this educational use case.

Workflow Engine For Analysis, Certification and Test (WEFACT) [11] is a tool developed by AIT (Austrian Institute of Technology) and it is one of the tools that has been used in the use case in order to represent the Generic Process Model. The tutors of the courses have used the tool and they have defined the requirements of the use case, the activities involved in the use case (based on the Activity Patterns) and the input and output artifacts of each of these activities.

Another tool that has been used in the educational use case is the extended version of the CHESS tool (Composition with Guarantees for High-integrity Embedded Software Components Assembly)[12] developed in the CHESS and SafeCer projects. In the use case, all the modeling and contract definition of the use case has been done using this tool. This tool has the option to define contracts for the system and it is possible to use a Contract Based Design approach.

There is also a tool called CHESS2OCRA [13] that translates the model of the system defined in CHESS to the OCRA [15] tool (contract based modeling). So in the use case, first the system is modeled using CHESS and then it is translated from CHESS to OCRA using the CHESS2OCRA tool.

As we explained earlier, the design of the control system has been based on contracts. In order to assure that the system fits the contracts, a contract based design has been applied and the tool called NuSMV3 [14]/OCRA has been used to verify the contracts.

## V. Implementation of the Educational Use Case

In this section it will be shown how the implementation has been done focusing how the contracts have been changed and OCRA has been used in order to verify that the collaboration of the software component that constitute the system, based on their contracts, fulfill system requirements. In particular, in this paper, only one safety requirement is considered: **if one elevator stops, the others must also stop within 50 millisecond**.

In the demonstrator, 2 elevators have been used in and for the sake of simplicity, they play a different role:

- the first one stops due to internal reasons, i.e. it has reached the top position.
- the second must follow this circumstance.

Although both elevator roles, and thus their contracts, are different, by symmetry of the system both elevator roles can be exchanged.

The elevator can reach up or down limit sensor, or its associated controller can decide to stop it due to any other detected situations. For analysis purposes, due that in any case the reaction of the system must be the same, these circumstances are modeled using a unique external input.

As we have shown in the implementation scenarios section, in order to show the benefits of the reusability of safety SW components, we have selected the scenario where the system has an evolution (new requirements) and as a result we are going to reuse/reintegrate the existing components in the new version. In the educational use case, the evolution of the system is based on an architectural evolution of the system but not on the safety requirements. The safety requirements are not changed. These are the architectural requirements that have been considered in both iterations:

- First iteration: the final system must have all the SW components in the same micro-controller.
- Second iteration: each of the controls have to be on different micro-controllers and they must be connected by a communication bus.

First, the centralized scenario will be shown and afterward, in the distributed one the same component contract will be used plus the one relative to the bus.

### A. Centralized Case

Each elevator, denoted here as Mobile01 or Mobile02, has next ports:

- stop: external input attached to a top/down limit sensor.
- on: a simplified output commanding the elevator in charge of a particular elevator controller.
- stopCmd: an input sourced at the main controller. If true, the elevator must stop.
- ready: an output port directed to the main controller aimed to notify if it can accept movement commands. It also means the elevator is stopped due to environmental reason (not because commanded so).

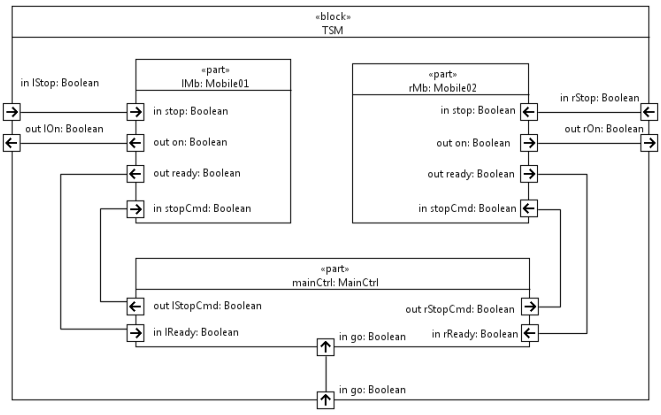


Fig. 8. Centralized Control System and Connection of its Components

As we have mentioned earlier, using the Activity Patterns we have to define safety SW contracts for each of the SW components of the system (elevatorsCtrl, mainCtrol) at Component Level and also the design contracts at System Level.

In this case, each elevator has been assigned a different contract: one must stop due to an external reason and notify this circumstance. The other must follow this reaction indirectly by means of the main controller. It is worth mentioning:

- The assumed minimal change rate imposed on the external stop signal and the stopCmd sourced at the main controller.
- The reaction time since stop is activated until this circumstance is notified through the ready port.
- The reaction time till an elevator switches off its on signal since the reception of a command though stopCmd port.

Each contract has two parts: the **Assumptions** part, where we define which are the assumptions of the component that we are considering and then the **Guarantee** part. Assuming that the conditions of the first part are fulfilled we guarantee that the conditions defined in the second part will happen. In this use case, the considered safety SW contracts are temporal contracts as we are considering only a temporal safety requirement (reaction before 50 ms).

In this case, in order to avoid the race-conditions, we assume that the *stop* and *stopCmd* inputs change rate is bigger than 50 ms and 10 ms respectively and with this assumption we guarantee the reaction time limits for the *ready* and *on* outputs in different situations.

The same pattern is used for the mainController contracts definition. The main controller, upon being notified that one of the elevator has stopped (or is not ready) must notify the other one to stop. In this case, the overall controller must command the right elevator to stop upon having been notified of such circumstance in the left one within a given interval.

And finally, the system contract will be the one of the non-functional requirement plus the addition of some assumptions

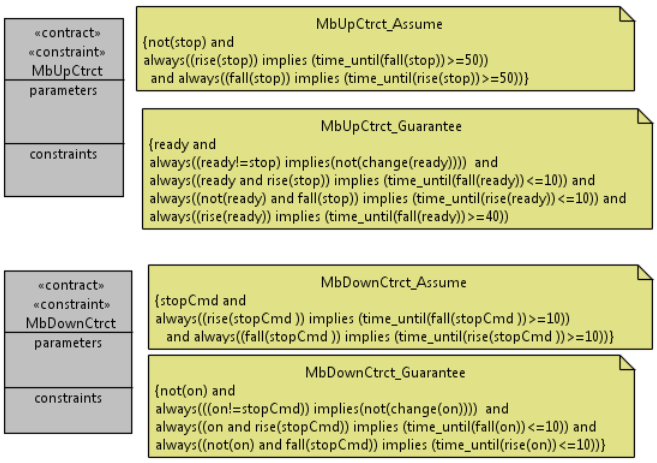


Fig. 9. Elevator's Contracts

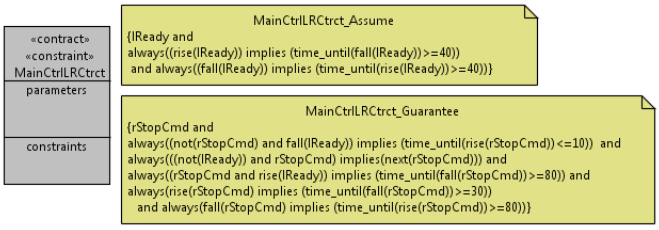


Fig. 10. Main controller's contract

about its environment.

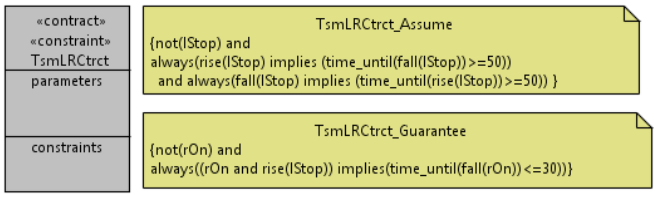


Fig. 11. System's Contract

### B. Distributed Case

In the distributed case each software component is deployed in a exclusively dedicated processor (regarding the control system) and the various input/output ports need the mediation of a field bus. Figure 12 shows the system composition.

As previously mentioned, the requirement verification is accomplished in an asymmetrical way showing that the right elevator follows the stopping of the left one. By symmetry it can be argued in the opposite sense.

It is worth mentioning that the reused components of the centralized system along with its contracts have been maintained unchanged. Also, this is the case regarding the system's contract. Thus only bus' contracts will be shown.

Bus contracts basically guarantee the retransmission of signals. Its duty has been specified by 2 contracts, one for each direction of those signals:

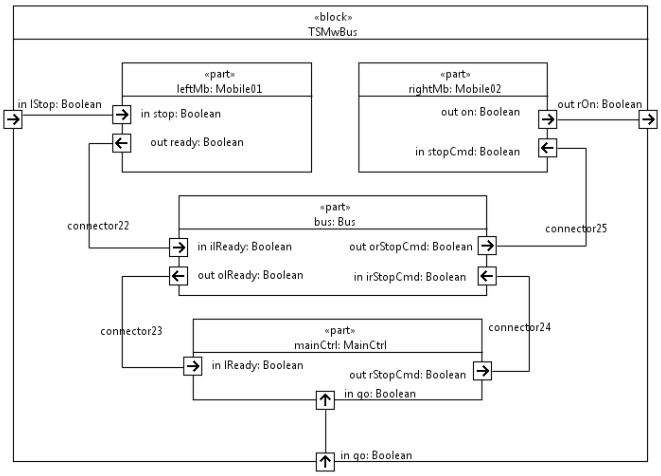


Fig. 12. Distributed Control System and Connections of its Components

- from the left elevator toward the main controller.
- from the main controller toward the right elevator.

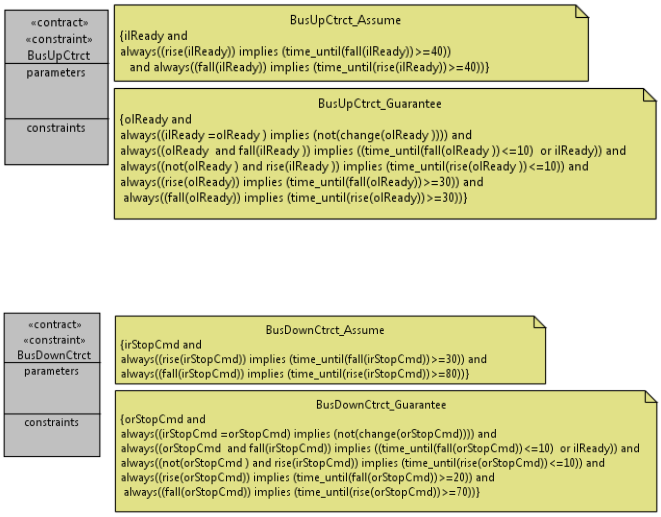


Fig. 13. Bus Contracts in the Distributed Control System

### C. Demonstrator Implementation

In this section some implementation details will be provided. The selected processors are LM3S8962 and LM3S2100, both from Stellaris (now Texas Instruments). They both are Cortex-M3 micro-controller. The used development boards run at 8MHz and the peripheral that have been used are CAN for connecting to the field bus, digital inputs for sensors, and PWMs for actuating on motors.

As communication software, on top of Stellaris BSP (Basic support package) a small middleware has been implemented following the publish/subscribe paradigm. The reason for such decision are that on the one hand, it is broadly used in industrial environments and is well suited for CAN bus, and on the other hand, it permits us to decouple software components from distribution decisions.

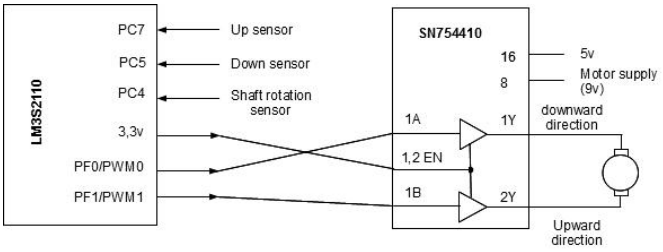


Fig. 14. Micro-controller I/O from/toward an elevator

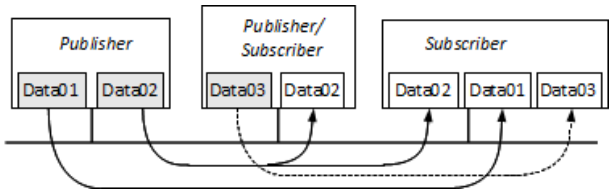


Fig. 15. Implemented publish/subscribe communication paradigm

Finally, software has been implemented in C language and GNU compilers for ARM. No operating system has been used but the system has been implemented as a bare-bone application with a cyclic executive. This scheme was enough for schedulabilty analysis.

In the Figure 16 we can see the mock-up of the developed elevator system:

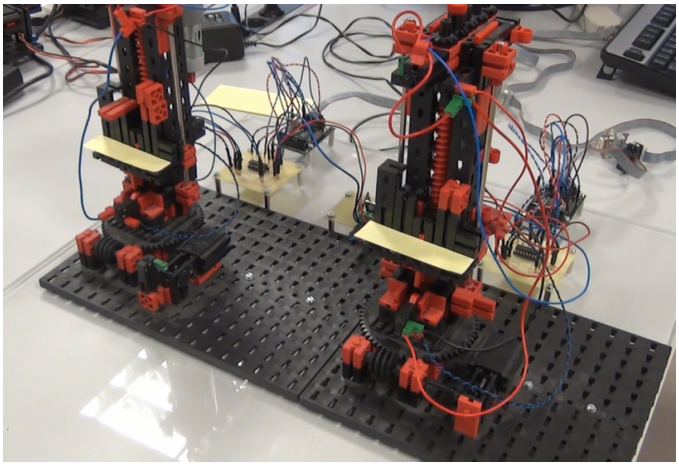

Fig. 16. Elevator System Control Mock-Up

## VI. Evaluation

Two groups of students (4-5 students in each group) have been working in the experiment. The same problem have been presented and similar results have been obtained. In the first iteration, they have designed and developed the elevator system as a Centralized System. The main objectives in the first iteration were very linked with the related subjects:

- Reliability: be able to do a Hazard Analysis and take the knowledge about reliability related concepts as SIL.
- System Design/Modeling: be able to design systems using SysML and also the Contract Based Modeling
- Real Time Systems: be able to implement/develop the system using Real Time concepts and communications buses like CAN

These three subjects, are not only based on the results of this use case, they have also another learning objectives. This practical exercise/experiment has an 20% impact in the final mark of each of the subjects.

In the first iteration these objectives were reached and this year the second iteration with the same students has been done. For the second iteration, the system have had an evolution and some architectural requirements have been changed. As a result, the students have had to implement a Distributed elevator control System. In this second iteration, the learning technical objectives were:

- Check and see the benefits of the reusability when designing and developing an updated system.
- Regulations and Standards: be able to interpret at least the IEC 61508 [4] functional safety standard and if possible other ones (CENELEC 50126 [1], CENELEC 50128 [2], CENELEC 50129 [3], ISO 26262 [7]).

The first objective has been reached easily because the students have designed and checked the second version of the system much faster than in the first iteration.They have seen that there was not need to change the components and the reuse of them and their reintegration was possible with little effort. At component level, the same SW components have been reused and a new one has been implemented (bus component). At system level, the system level contracts have been reverified and revalidated in order to see that the new integration of the components reaches the safety requirements of the system.

In order to reach the second objective, an extra exercise has been defined. One of the groups has done a study about what will be affected if the SIL level is changed because of the new requirements that are considered in a new system. For that, some environmental conditions has been changed (sensor's failure probability, location of the system...) and the new requirements involve a new SIL level for the system. For the other group, the application domain has been changed. They have studied about reusing the elevator system control in the control of an automatic roof of a car. For that, they have considered the ISO 26262 [7]. Finally, the results of each of the studies have been presented in the class.

## VII. Conclusions

Once the experiment has finished, these have been the conclusions that the students have written about their works:

- Reusability: The benefits are related to the time needed for doing the update (faster: first iteration about 40 hours work, second iteration less than 20 hours work) and also the new system will be safer (proven in use components reuse). The main reason for that it is because in the second iteration it is not necessary to implement/change and verify all the components. Doing the composition analysis is enough.

- Standards and Regulations: A real use of the standards has been done and the structure and main concepts of the functional safety standards have been learned.

In order to do the evaluation of the Educational Use Case, this inputs have been considered:

- Students Results, Exams Results
- Students Satisfaction Survey

Regarding to the students results, they have been very satisfactory. The difficulty level of the exams they have done was similar to the previous years' and the results have been a bit better (5%). Also the way that they have worked have allowed to do a better learning of the functional safety standards because they have "*use*" them.

If we focus on the students satisfaction survey, in this case, the results have been good. The practical way they have worked and the way to learn the new concepts and functional safety standards has had a very positive response. The satisfaction of the students regarding to the courses has been $7.2/10$ and in previous years was $6.9/10$.

The qualitative and quantitative results of the evaluation of the courses have been analyzed. The results are good, nevertheless it is not possible to conclude that the E&T Use Case that have been used in the courses to be the reason of the good results.

Moreover, a specific questionnaire about the use case and the experience of using it in the learning process has been filled by the teachers in charge of the courses involved in the E&T Use Case. And the results are very positive; the four teachers involved think that the E&T Use Case is very useful for transferring knowledge related to the courses in a practical way and make learning process easier.

As a final conclusion, the obtained results have been very positive and the main objectives have been reached:

- The reusability of SW components in Safety Critical Embedded Systems benefits has been shown and transfered to the students, future engineers.
- Functional Safety Standards have been studied in a practical and efficient way.

Using the questionnaires and surveys mentioned above, very valuable information could be obtained but the results could be biased or be subjective as the results are based on the opinion of involved students and teachers. In order to obtain evidences of the effectiveness of the use case, an empirical experiment should be carried out in the medium and long term, but this will be outside the scope and time schedule of SafeCer. We will have to restrict to first evidence towards end of the project.

So, the idea of the University of Mondragon is to continue using the Educational Use Case in the Embedded Systems Master because it is a very good way to acquire the competences and knowledge in this area (standards and regulations, safety critical systems' development methods, etc.) and also to be able to minimize the efforts in new developments of this kind of systems taking into account the reusability.

## ACKNOWLEDGMENT

The research leading to these results has received funding from the ARTEMIS JU under grant agreement 295373 (nSafeCer project) and from National funding (Ministerio de industria, energa y turismo). The project has been developed by the embedded system group supported by the Department of Education, Universities and Research of the Basque Government.

## REFERENCES

[1] AENOR, UNE-EN 50126: Aplicaciones Ferroviarias: Especficacin y demostracin de la fiabilidad, la disponibilidad, la mantenibilidad y la seguridad (RAMS), 2005.

[2] AENOR, UNE-EN 50128: Aplicaciones Ferroviarias. Sistemas de comunicacin, sealizacin y procesamiento Software para sistemas de control y proteccin de ferrocarril, 2002.

[3] AENOR, UNE-EN 50129: Aplicaciones Ferroviarias. Sistemas de comunicacin, sealizacin y procesamiento. Sistemas electrnicos relacionados con la seguridad pra la sealizacin, 2005.

[4] IEC, "61508:2010, Functional safety of electrical/electronic/programmable electronic safety related systems," 2010.

[5] M. Illarramendi, L. Etxeberria and X. Elkorobarrutia, "Reuse in safety critical systems: Educational use case," in Workshop Session on Teaching, Education, and Training for Dependable Embedded and Cyber physical Systems [ERCIM/ARTEMIS/EUROMICRO], Santander, 2013, pp. 402-407.

[6] M. Illarramendi, L. Etxeberria and X. Elkorobarrutia, "Reuse in Safety Critical sSystems: Educational Use Case First Experience," in Workshop Session on Teaching, Education, and Training for Dependable Embedded and Cyber physical Systems [ERCIM/ARTEMIS/EUROMICRO], Verona, 2014.

[7] ISO, "ISO26262 Ed.1: 2012, Road vehicles- Functional Safety," 2012.

[8] SafeCer, "A Generic Process Model for Integrated Certification and Development of Component-based Systems," 2014.

[9] SafeCer Project, "CTF platform prototype: Software description overview," Tech. Rep. D3.1.3, 2012.

[10] SafeCer Project, "Survey of available tools," SafeCer, Tech. Rep. D3.1.1, 2011. 2011.

[11] AIT, "WEFACT-Workflow Engine for Analysis, Certification and Test," AIT, 2012.

[12] CHESS Artemis Project, "Component based modeling environment with dependability and real time analysis support" , http://www.chess-project.org/page/download,2012.

[13] INTECS, "CHESS2OCRA:transformation of a CHESS UML model (.uml) in a OCRA input file (.oss), 2014.

[14] FBK, "NumSMV3: Functional verification, dependability and safety analysis tool", https://es-static.fbk.eu/tools/nusmv3/

[15] FBK, "OCRA: Package of NumSMV3 model checker that supports Othello Contracts" ,https://es-static.fbk.eu/tools/ocra/