

Requirements Documentation of a Controlled Complex Motion System

Alessandro De Carli, Stefano Andreozzi, Andrea Fiaschetti
Department of computer and control systems “Antonio Ruberti”
“Sapienza” - University of Rome
Via Ariosto, 25 - 00185 Roma
alessandro.decarli@uniroma1.it

Abstract-The implementation of a controlled complex motion system requires integration of technologically heterogeneous subsystems and collaboration of experts, skilled in different technologies. To attain the desired requirements it is suitable to apply a design procedure organized according to a systematic approach, divided in different steps and representing standalone activities. The fundamental steps are the elicitation, decomposition and formalization of requirements. The systematic procedure proposed in this paper can be developed and supported by the UML (Unified Modelling Language) software tool. Different diagrams characterize both the most significant aspects of the hardware and software structure and the functional facilities. The procedure has been applied to a very simple controlled motion system, just to outlight the advantages and to encourage its reproduction for tutorial purpose.

I. INTRODUCTION

Engineering of controlled complex motion systems represents a challenge to different applicative sectors since a controlled motion system consists of different elements controlled in diverse ways. The elements are organised according to hierarchical levels and are strictly interconnected by interactions, not always well defined and well specified. Different technologies are applied to implement a controlled complex motion system.

The realisation of controlled complex motion systems presents problems related to:

- definition of desired functionality and specifications;
- design procedure based on not always well defined empirical approaches;
- integration of heterogeneous subsystems;
- elements assembled in subsystems characterised by a specific functionality;
- assembled system having performances dissimilar from the foreseen and desired one, although each subsystem has been correctly designed and realised;
- interactions increasing the complexity and consequently the quantity and the severity of problems;
- design procedures carried out without taking into account interactions due to the integration of subsystems independently designed and implemented [1].

In the design of controlled complex motion systems, an essential aspect is related to identification and separation of dif-

ferent problems in order to make possible a more successful evaluation of alternative design solutions. In such a way the selection of controlled complex motion system functionalities becomes easier, so as the coordination of system elements to realise subsystems integration. It is then advisable to split interactions of elements from their internal behaviour, especially when a parallel operation of some subsystems is foreseen [2].

To overcome these problems it is suitable to divide the design procedure in different successive steps each one standardised, self-contained and weakly interacting. To verify the validity of the produced work, a procedure that relates the different design steps is applied and sequentially carried out. This procedure is tailored to the structure design of a controlled complex motion system, the elements for its implementation, and the software for the controlled operation. Fig. 1 illustrates the block diagram of this procedure and evidences the steps that should be carried out firstly to develop the implementation and than to perform the partial and the final validation [3].

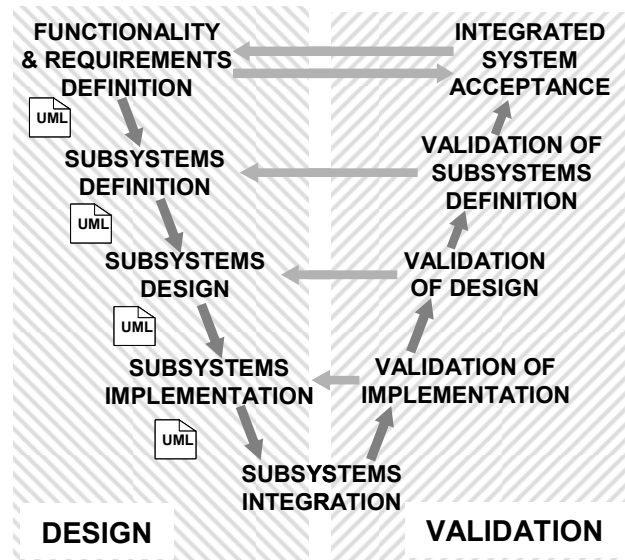


Fig. 1. Design procedure and its validation.

The procedure can be represented by blocks organised according to the “V” structure. On the left side, the blocks represent the sequence of the fundamental design steps that ends with the implementation. On the right side, starting from the

bottom the sequence of the partial validation steps is illustrated and related to each design block.

To achieve the result of dividing the design procedure in a sequence of steps, it is necessary to apply a systematic approach which suggests to separate the design of elements, composing a controlled complex motion system, in subsystems. In such a way it is possible to utilise some subsystems in the design of the same motion system or in similar ones. It will be also easier to recognize lacks in the design procedure just when this latter is in the developed state. Evaluation and improvements of the performance specifications can be carried out more easily during the design procedure rather than during the validation tests and on the implemented system. In fact, eventual lacks are foreseen and not discovered during the motion system operation.

The design procedure can be carried out following two different approaches. In the traditional one, bottom-up, the evaluation and the improvement of the performance specification is carried out at the end of each design process. In the innovative one, evaluation and improvements are carried out just from the beginning of the design steps by applying a virtual reality simulation supported by some suitable and already available software.

In the new approach to the design process, two professional figures should cooperate. The first one is the domain engineer, skilled in the specific sector in which the motion system should operate, the second one is the knowledge engineer skilled in software support for approaching the design according to the innovative trend.

This paper illustrates the innovative approach and the application to a controlled complex motion system that has been chosen very simple in order to make easier the understanding of the proposed procedure and to facilitate its acquirement by reproducing the example.

II. REQUIREMENTS

A controlled complex motion system generally interacts with its surroundings and should be realised so as to attain the desired performance specifications. The set of user requirements concerns the definition of the performances that the end-user desires from the complex motion system. The set of system requirements contains the definition of functionalities and desired performances that a system should achieve. The peculiarities of external surroundings form the knowledge domain. Its properties together with proper functionality allow to attain the user requirements [4].

To apply the proposed innovative approach, it is suitable to start from the user requirements in order to identify and structure the system requirements. The first ones are in general formulated by an end-user by using a natural language and formulated in terms of the desired targets. The system requirements are defined from the controlled complex motion system designers and regard an object oriented description organised in a hierarchical way.

The requirements elicitation gives the following advantages in the design process:

- evidence of desired functionalities;
- history of requirement changes during the design process;
- validation tests and their quantitative evaluation organised and carried out during the design process;
- documentation of the main aspects of the controlled complex motion system by using a standard and not strictly technical approach;
- involvement in design and implementation of people differently skilled by using an appropriate documentation.

In such a way the requirement documentation constitutes a clear and easily understandable support to formulate contractual requests.

In working out requirements it is advisable to distinguish between functional requirements and non-functional requirements [5]. The first ones describe the controlled complex motion system behaviour during its operation and characterise the relation between:

- input variables, i.e. variables that can be modified by the end-user;
- internal variables, i.e. variables characterising operating conditions;
- output variables, i.e. variables indicating the attainment of the target behaviour by the input variables application.

The non-functional requirements describe: performance aspects, feasibility, robustness, failure tolerance, security, usability, controlled complex motion system behaviour during preview operating conditions, cost, testability, maintainability, reusability, documentation. All these aspects influence in a relevant way cost and quality of the design product.

Requirements are firstly formulated in a structured text organised in a database; subsequently repetitive interactions between the end-user and the design staff allow to obtain that requirements should be:

- global: i.e. concerning the whole system and being always valid;
- correct: i.e. respecting constraints due to physic feasibility and actual normative;
- complete: i.e. formulating sentences and words having a univocal and absolute meaning;
- clear: i.e. presenting no ambiguities;
- consistent: i.e. avoiding significant conflicts;
- modifiable: i.e. offering the possibility of updating;
- verifiable; i.e. applying objective criteria and specific metrics for validation;
- traceable: i.e. easily identifiable;
- feasible: i.e. respecting temporal and economic constrains;
- minimal: i.e. presenting no redundant and unnecessary aspects.

The above mentioned requirements are represented by means of an object oriented language, based on a suitable set of diagrams and well defined formal rules. Such diagrams represent the object oriented model of requirements. In order to make the discovering of errors in requirements easier, it is ad-

visible to carry out an object oriented representation in such a way that:

- the interaction between objects should be very low;
- the existing interaction should very clearly indicated,
- each object should be characterised only by the essential interactions.

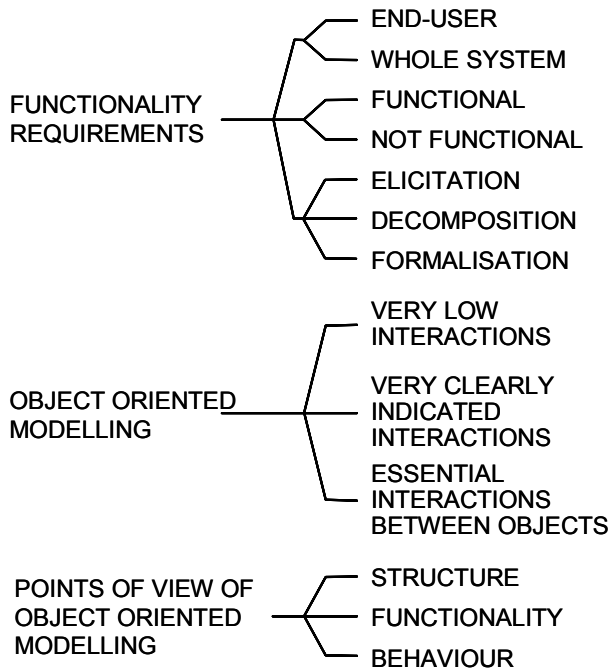


Fig. 2. Requirements characterization and their documentation.

In the object oriented description, the point of view related to different aspects of the requirement knowledge should be carried out so as to represent the structure, the functionality and the behaviour [6].

Each point of view relates one aspect of the controlled complex motion system with its desired performance. The aspects are: definition of its behaviour and its elements, interactions with external agents, intrinsic constraints and the necessary external resources, as shown in Fig. 2 [7].

The formalisation of controlled complex motion system requirements makes possible the establishment of a specific function library that can be utilised in the development of new projects. In object oriented modelling, a low level of abstraction allows a direct and more useful reutilisation in development of new projects. Although a significant reduction of cost and time to market could be attained, modifications of controlled complex motion system requirements could determine deep changes in his implementation. A high level of abstraction allows a larger homogeneity of different aspects in the formalisation of the controlled complex motion system qualifications but it is possible to represent only a limited part of efforts to attaining the final result [8].

III. ELICITATION AND DECOMPOSITION

In the object oriented modelling of a controlled complex motion system, the proposed approach foresees three steps: elicitation, decomposition, and formalisation. Suitable software can be applied for developing each step.

In the deduction of controlled complex motion system requirements, risk of errors is high. If these errors are not promptly corrected, the attainment of the desired final result should be compromised. Moreover it is very important to consider that, at the beginning of the design process, inappropriate project choices could increase cost of implementation and management of a controlled complex motion system. For these reasons requirements should be elicited by an iterative procedure dedicated to reducing uncertainties.

The elicitation of controlled complex motion system requirements starts from the document presented by end-users. In general, this document illustrates the project and the desired performance in a textual form, i.e. a technical note. In order to allow an easier and deeper understanding of this document, a suitably selected list of symbols and words should be used, with each element having a meaning not only clearly comprehensible and shared between end-users and designers, but also by the domain experts and system engineers. At the beginning of the design process the definition of quality indices is fundamental.

Inside this document, the homogeneous requirements and their definitions are defined and included in a structured and well organised database. It is advisable to apply object oriented software to structure and manage the requirements of a controlled complex motion system.

Requirements are considered objects and organised according to tree-shaped hierarchical modules. Headings are utilised as knot objects while requirements become leaf objects. For each requirement it is possible to define attributes at module levels and at object levels.

The set of information that can be memorised as an attribute is specified by its definition. Combining these facilities with the possibility to query the database in terms of attribute contents, an end-user can define the different points of view utilised to view a precise attribute or to export information versus other software tools. Each point of view can be stored inside a module for a subsequent utilisation. At the end, the unidirectional and bidirectional dependences between correlated elements are defined. This latter facility is particularly efficient when in the controlled complex motion systems hierarchical dependences are active between two or more elements.

The peculiarities, that make this approach particularly attractive, are:

- production of reports containing structured requirements that can be carried out in textual and in graphical forms. The formalism helps both the design organised in working group and the validation of design proposal elaborated by different proponents. This procedure is generally indicated as standardisation of requirements;

- all the modifications are memorised in details by means of the database.

These aspects are very important due to the rapid evolution in technology and standardisation of controlled complex motion systems. Modifications are frequently requested in developing new projects and in updating the already existing ones.

IV. FORMALISATION

The visual language used to formalise requirements is Unified Modelling Language, generally indicated as UML. It is not proprietary and it is based on a limited set of well defined graphical symbols that allow the object oriented representation.

UML diagrams can be grouped into three different but complementary sets that give the following points of view

- functional points of view to illustrate the utilisation modalities;
- static points of view to illustrate the structure and the implementation of each element;
- dynamic points of view to illustrate the operating modalities during the time.

The functional point of view includes the use case diagram and the activity diagram. These diagrams allow the description of the operations of a system. In particular, the use case diagram describes functionalities needed by the system, whereas the activity diagram describes the sequence of events and interventions, the logical connections and the parallelism.

In the static point of view, class diagram and object diagram are utilised to represent the system structure without describing its behaviour. The first one is utilised as the basis to realise an object oriented structure and to define the related functionalities. The second one is utilised to represent information as objects dedicated to the realisation of each class.

In the dynamic point of view, sequence diagram, collaboration diagram, a state diagram are utilised to describe the interactions between the system components and their time evolution. In particular, state diagram informs on the modalities according to effects of input events and on internal effects of these inputs.

The sequence diagram is used to describe the time evolution of each element involved in a particular operation described by the use case diagrams. Moreover it is used to give a detailed description of the messages exchange inside the system. All changes of information on origin, destination, temporal constraints, and interdependences of messages are indicated.

UML diagrams can carry out all the design process. Firstly it is suitable to consider the whole system and operate with higher level of abstraction. By starting from a structured report, three kinds of information should be worked out:

- subjects: i.e. entities involved in the system evolution due to external and internal actions devoted to obtain the desired functionality and performance;
- functions: i.e. actions and procedures effected by the system, without considering a detailed description;

- interactions: i.e. between components of a system and their functionality.

This procedure is analogous to the elaboration of a function prototype by using an imperative programming language.

To draw a sequence diagram interacting objects are firstly identified, i.e. objects interchanging messages during the system or subsystem operation, and then a temporal axis is assigned in order to specify the exchanged messages. Starting from the sequence diagram, the design works out the functional subdivision by utilising the activity diagrams obtained by identifying the elementary activities that contribute to attain the desired functionality.

Finally state diagrams are drawn. They combine the functional points of view given by the activity diagram to the logical sequence of messages obtained by the sequence diagrams.

The elements forming a state diagram are:

- states: i.e. actual conditions or existing modes in which a system can be;
- conditions: i.e. situations that should be verified to permit the transition into a new state;
- events: i.e. specific actions that trigger a condition;
- transitions: i.e. the events that allow to activate a new state.

V. EXAMPLE

In order to appeal the use of the above proposed approach, the procedure has been applied to a tutorial example that can be easily reproduced and improved by people interested to acquire the innovative way to the design and the documentation processes. Although very simple, the example is strictly related to some significant industrial applications.

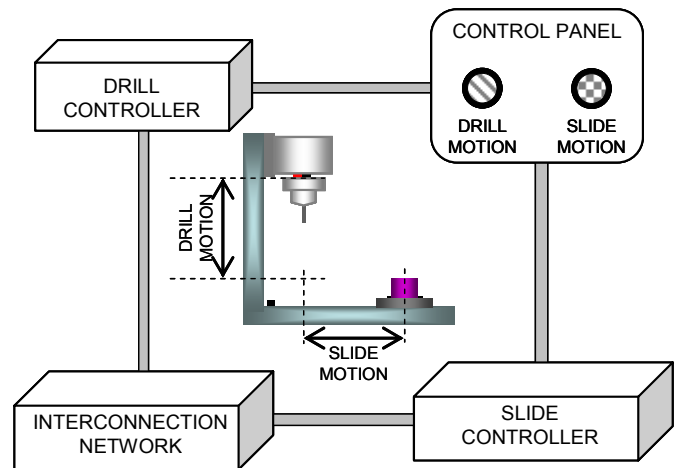


Fig. 3. The motion system of the example.

Fig. 3 shows the proposed motion system. It consists of a column drill and a slide in which the piece that should be drilled is placed. The motion of the drill and the motion of the slide are both prefixed. The drill and the slide are connected to dedicated controllers by a communication network. A control panel fixes the start of the drilling process operating cycle by the direct action of the system operator or by a suitably pro-

grammed PLC. Position sensors determine the top and the low position of the drill and the left and right position of the slide.

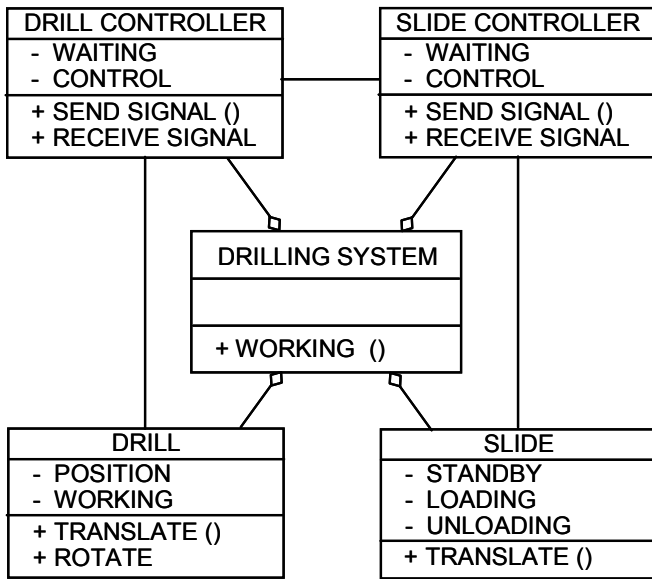


Fig. 4. The Object Diagram

The first step in the design procedure consists in defining the functionality and in selecting the structure and the related components. Fig. 4 documents the main components by the UML object diagram.

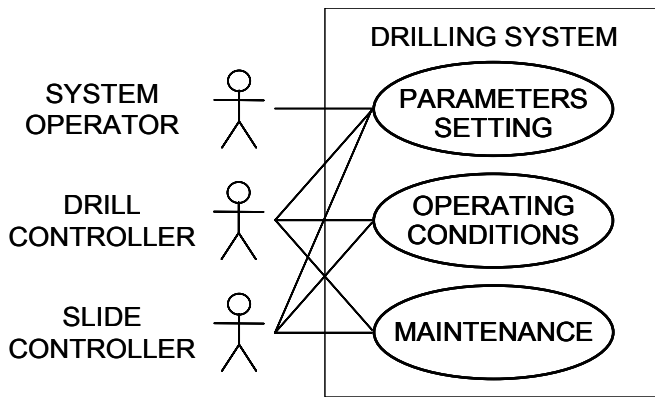


Fig. 5. The Use Case Diagram

In order to work out the most important operating conditions, in order to design the programs that should be implemented in a PLC, reference will be made to:

- the modalities for setting the parameters characterising the operating cycle;
- the aspects characterising the nominal operating conditions;
- the planning of the maintenance.

The use diagram in Fig. 5 indicates the related documentation.

Assuming that that the working cycle is activated by a system operator action whereas the drill and the slide are con-

trolled by the dedicated controllers, the drilling process consists of the following steps:

- the system operator activates the working cycle by the control panel;
- the control panel activates the slide controller to move the slide to the right end position;
- the piece to be drilled is put on the slide;
- the slide moves to the left end to prepare the piece for the drilling process;
- the slide controller activates the drill controller;
- the drill controller activates the drill vertical motion to reach the piece surface;
- the drill controller activates the drill rotary motion to realise the hole in the piece;
- the drilling process is effected and completed;
- the drill control activates the motion of the drill up to the standby position;
- the worked piece is discharged;
- the slide controller signals to the control panel the end of the drilling process.

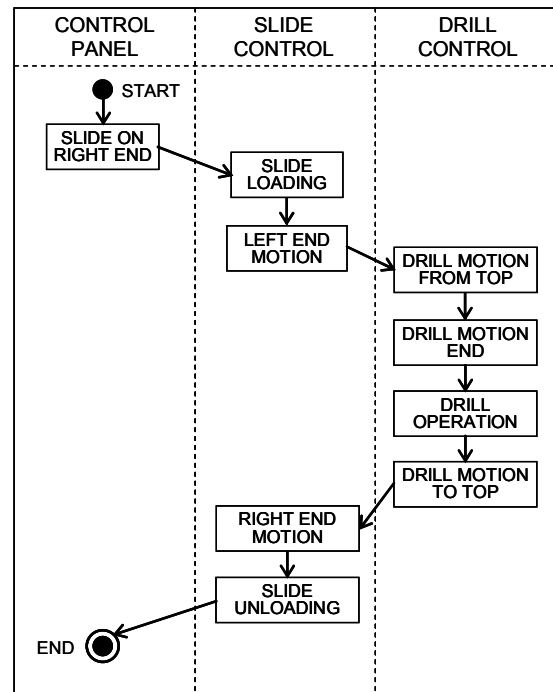


Fig. 6. The State Diagram

The actions planned to realise the working cycle can be very clearly documented by the UML state diagram shown in Fig. 6 and by the sequence diagram of Fig. 7.

All the control actions can be concentrated in a PLC controller. The design procedure of the related program can be carried out in a more effective way by using the UML diagrams. The advantage consists in a first easy verification of the validity of the control action sequence and in the documentation of the design project in such a way that can be understood by different skilled people, involved in by implementation of the controlled system. The final verification

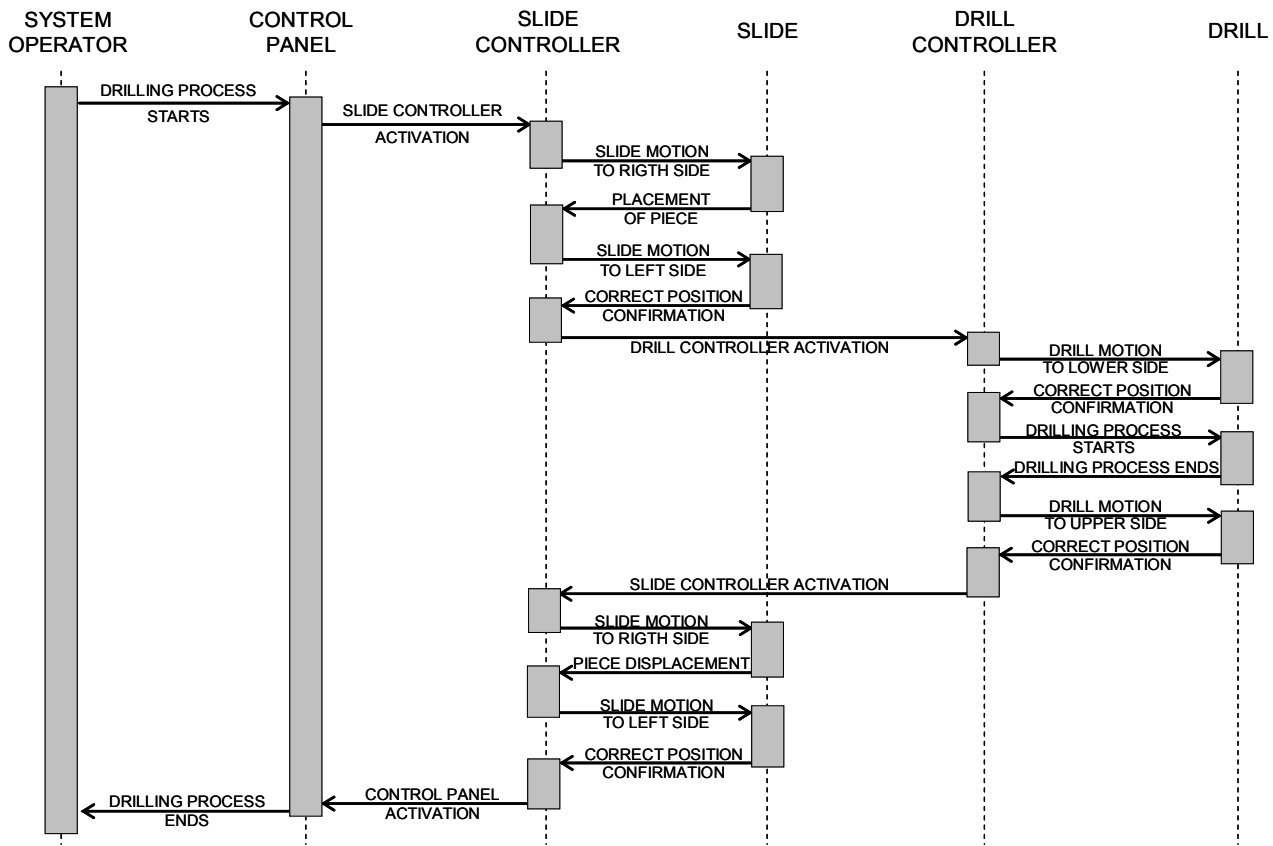


Fig. 7. The Sequence Diagram

of the program validity can be effected by using the State-flow toolbox of Matlab for simulating the UML sequence diagram.

VI. CONCLUSIONS

The proposed procedure is aimed by the necessity of reducing the cost of the controlled system and the reduction of the time to be marked. In fact, the global cost of a controlled complex systems is highly influenced by the choices effectuated in the first steps of the design process. The attainment of the desired functionality, efficiency and requirements depends by the cooperation of designer staff in which members have different experience and background.

The systematic organisation of the design process according to a widely accepted systematic procedure allows to make transparent and easily transferable the results for the evaluation. The UML diagrams facilitate the documentation and the transparency.

Since UML software is free, the initial cost for applying the innovative procedure is very low. Once an acceptable competence is reached by the designer, more sophisticated software are available on the market [9].

For people involved in the design using home developed procedure, practically inapprehensible for different skilled people, the application of the innovative procedure for developing the design process and for the documentation of the results does not require significant efforts. The proposed exam-

ple has been selected intentionally very simple to make easier this passage, but has been used in a significant application [10].

REFERENCES

- [1] R. Isermann, "Mechatronic systems-innovative products with embedded control," *Control Engineering Practice*, vol. 16, pp. 14-29, 2008.
- [2] M. Sgroi, L. Lavagno, and A. Sangiovanni-Vincentelli, "Formal models for embedded system design," *IEEE Design & Test of Computers*, vol. 17, issue 2, pp. 14-27, 2000.
- [3] B.W. Boehm, "Guidelines for verifying and validating software requirements and design specifications," in *Proc. of the European Conference on Applied Information Technology (IFIP '79)*, pp. 711-719, 1979.
- [4] J. Hammond, R. Rawlings, and A. Hall, "Will it work?," in *Proc. of the 5th IEEE International Symposium on Requirements Engineering*, pp. 102-109, 2001.
- [5] S. R. Faulk, "Software requirements: a tutorial," in *Software Requirements Engineering 2nd Edition*, R. Thayer M. Dorfman Eds, IEEE Computer Society Press, 1997.
- [6] R. Kawabata, and K. Itoh, "Knowledge representation and knowledge management for various phases in systems analysis and design," *The ATLAS Module Series on Transdisciplinary Education & Research, TAM-2*, pp. 1-21, 2005.
- [7] B. Nuseibeh, and S. Easterbrook, "Requirements engineering: a roadmap," in *Proc. of IEEE Conference on The Future of Software Engineering*, Limerick, Ireland, pp. 35-46, 2000.
- [8] K. Keutzer, A. R. Newton, J. M. Rabaey, A. Sangiovanni-Vincentelli, "System-level design: orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, issue 12, pp. 1523-1543, 2000.
- [9] D. Harel, and A. Naamad, "The STATEMATE semantics of statecharts," *ACM Transactions on Software Engineering and Methodology*, vol. 5, no. 4, pp. 293-333, 1996.
- [10] A. De Carli, S. Andreozzi, and A. Fiaschetti, "Documentazione dei requisiti di un sistema controllato", *Automazione e Strumentazione*, Anno LV, no. 11, pp. 80-86, Dicembre 2007.