A TESTING MODEL FOR THE MECHATRONIC SYSTEM

Liviu Mihai SIMA[1], Miron ZAPCIU[2]

Mechatronics is a technology that has developed and continues to develop more and more. The mechatronic system considered is the result of the integration of mechanical, electrical / electronic parts, as well as information processing. In order to carry out a systematic design of a mechatronic system, an interdisciplinary discipline of integrated design is required. To organize the design of activities in different disciplines and help designers to achieve multidisciplinary design integration, a design methodology is on a multidisciplinary interface model proposed. According to engineering systems, an extended "V"-model used as a macro-level process in the proposed design methodology. This starts with identifying system-wide requirements and ends with a user-validated system. The hierarchical design model adopted as a micro process level. The article is discussing about the ways to test a mechatronic system.

Keywords: mechatronic system, testing / techniques type, integrated design methodology, multidisciplinary interface model.

1. Introduction

The mechatronics is a multidisciplinary branch of engineering that focuses on the engineering of electrical, electronic and mechanical systems. The term "mechatronics" defines a framework for technical and practical considerations. [1]

The term mechatronics originated at the Yaskawa Corporation from the combination of mechanics and electronics in 1969. After the 1970s, the meaning of mechatronic used to include software and computation. [2]

With the development of technology, other disciplines (e.g. optics, hydraulics and pneumatics) are involved in the development of mechatronic systems. As mechatronic systems encompass a wide range of disciplines, the design of mechatronic systems requires a multidisciplinary integrated design. [3]

The design methodology can help the engineers from different disciplines to enable their collaboration for increasingly complex tasks. Until now, was not an effective design methodology provided, which can fully support the engineers to achieve such multidisciplinary integration. [4]

[1]PhD student, University POLITEHNICA of Bucharest, Faculty of Industrial and Robotics Engineering, Department of Robots and Production Systems, Bucharest, Romania, e-mail: liviusima23@gmail.com

[2]Prof., PhD coordinator, University POLITEHNICA of Bucharest, Corresponding member of Academy of Romanian Scientists, Scientific Director of Research Institute SC INCERTRANS SA, Bucharest, Romania, e-mail: miron.zapciu@upb.ro

The technology advances and the fields of engineering have managed to diversify. The intention of mechatronics is to produce a design solution that unifies each of them. Initially, the field of mechatronics was just to be a combination of mechanics and electronics, hence its name. [5]

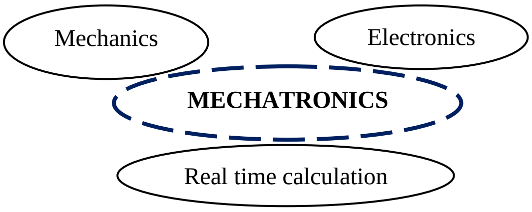

Fig. 1. The mechatronic discipline

The development of mechatronic systems opens the possibility to other innovative solutions and synergistic effects that are not possible only with mechanics or electronics. This technical progress has a very strong influence on a multitude of products in the fields of mechanical, electrical and electronic engineering. The design of conventional electromechanical components is increasingly changed. As examples, could be machines, vehicles or precision mechanical devices. Numerous processes and technical products in the field of mechanical and electrical engineering show an increasing integration of mechanics with digital electronics and information processing. This integration is done between components (hardware) and information-based functions (software), resulting in integrated systems called mechatronic systems. Their development involves finding an optimal balance between the basic mechanical structure, the implementation of electrical / electronic components and automatic information processing. [6]

The main concept of mechatronics is to work smartly and efficiently to achieve remarkable results in a short time. Mechatronics described in several ways: the combination of mechanics and working with precision engineering, sensor technology, actuator technology, computer science, and control theory and sensor technology. [7]

2. Ways to test a mechatronic system

The system several test models designed and / or developed. These models are mainly in the software industry used. [8]

Testing is necessary because some mistakes could appear. Some of those are not so important, but some of them are expensive or dangerous. Everything that happens needs checking, because things can go wrong, and the people are doing mistakes. [9]

Software development models are different processes or methodologies selected for project development, depending on the objectives of the project. Many development life cycle models were been developed to achieve the goals desired. The models specify the different stages of the process and the order in which they performed are. [10]

The selection of a model has a very big impact on the type of test performed. Is defining what, where and when from what planned is, influence the regression of the tests and largely determine which testing techniques needs to be used.

There are various software development models or methodologies. The ones take into consideration in article: Waterfall model, agile model, incremental model, RAD model, and iterative model, spiral model, prototype model, the "V"-shaped model and its variations.

The product can be during the production process from several points of view tested. Depending the product planned for testing in the manufacturing process, it can follow a certain type of model. The focus is on the Waterfall and "V"-models, which are being usually the most models used in testing. These are the two models exemplified. [11]

2.1. The Waterfall model

This model is a breakdown of project activities into linear sequential phases, in which each phase depends on previous results and corresponds to a specialization of tasks. The approach is typical for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows largely in one direction ("down" like a cascade) throughout the phases of design, initiation, analysis, design, construction, testing, implementation and maintenance.

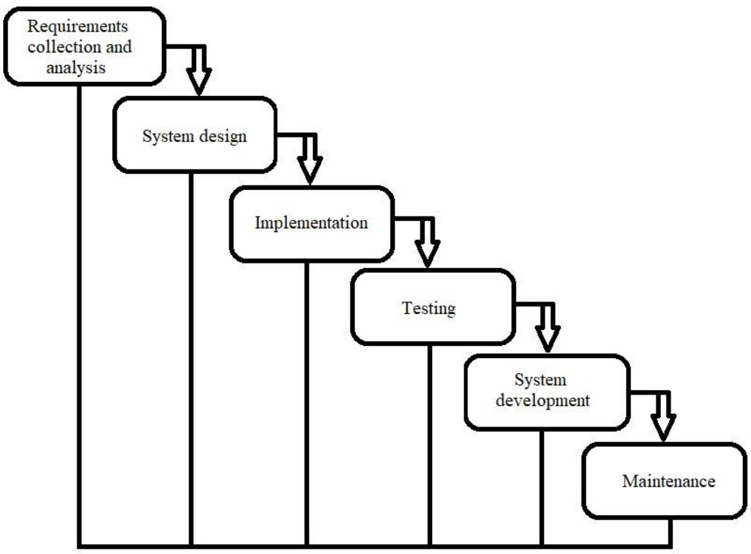

Fig. 2. "Waterfall" product testing model [8] [9]

The Waterfall development model has its origins in the manufacturing and construction industries. When this model was for software development first adopted, there was no recognized alternatives to knowledge-based creative work.

The Waterfall model refers to the fact that it should move to another phase only when the previous phase is reviewed and verified. In the case of the Waterfall model, very little customer interaction is involved during product development. Once the product is ready, it can be to end users shown. Once the product is developed and a malfunction occurs, the cost of solving such problems is very high. This situation appears because everything is from the document updated to the logic of the design. In practice, the Waterfall model has been by other models replaced.

2.2. The "V"-model

This model is a simple variant of the traditional Waterfall model with system or software development. As illustrated in Figure 3, the "V"-model is on the cascading model based by emphasizing verification and validation. The "V"-model takes the bottom of the Waterfall model and bends it upwards in a "V" shape so that the activities on the right check or validate the work products of the activity on the left. Specifically, the left side of the "V"-model represents the analysis activities that break down users' needs into small, easy-to-manage pieces. The right side of the "V" shows the corresponding synthesis activities that agree (and test) these parts into a system that meets the needs of users.

The "V"-model means verification and validation of the testing. Like the Waterfall model, the "V"-shaped life cycle is a sequential way of executing processes. Each phase needs to be before the start completed of the next phase. Product testing is in parallel with a corresponding development phase planned from the "V"-model.

Requirements such as BRS and SRS start the life cycle`s model just like the Waterfall model. In this model, is a system test plan created before the start of development phase. The test plan focuses on fulfilling the functionality specified in the requirements collection.

The HLD phase focuses on the architecture and design of the system. This phase provides an overview of the solution, platform, system, product and service or process. In this phase, is an integration plan created. Is need to test the capacity of collaboration from the software systems.

The LLD phase occurs when the actual software components designed are. This phase defines the real logic for each component of the system. The class diagram with all methods and the relationship between classes falls under the LLD. Component tests are at this stage also created. The implementation of the code takes place in the implementation phase. Once the coding complete is, the execution path continues on the right side of the "V". The previously developed test plans are now used. The encoding is located at the bottom of the "V"-shape model. The elements are from the module converted to code. The unit is by the developers tested on the code written by them. Especially in software development, the "V"-model is a development process that can be as an extension of the Waterfall model considered. Instead of moving them down linearly, the process steps bent upwards after the coding phase, to form the typical format of the "V"-model.

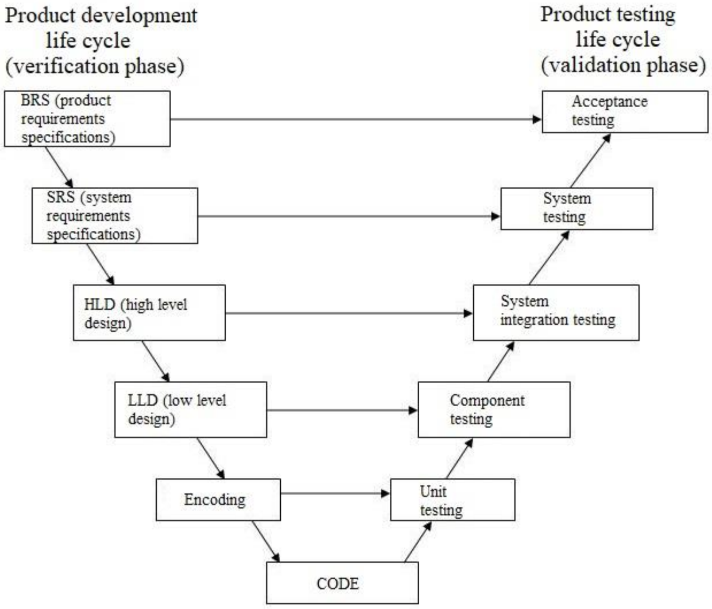

Fig. 3. Product testing of the "V" model [8] [9]

The "V"-model demonstrates the relationships between each phase of the development life cycle and the associated test phase. The horizontal and vertical axes represent the time or complexity of time (from left to right) and the level of abstraction.

This model is a graphical representation of a life cycle of the system development. Usage is to produce rigorous development lifecycle models and project management models. The "V"-model summarizes the main steps together taken with the appropriate results from the computerized system validation framework or the development of the project life cycle. The model describes the activities performed and the results produced during product development. The left side of the "V" model represents the breakdown of requirements and the creation of system specifications. The right side of the "V" represents the integration of the parts and their validation. However, the first requirements need validated according to the higher-level requirements or needs of the user. The verification can be done partially on the left side of the "V"-model. It is an incorrect approach to say that the validation only takes place on the right side of the "V"-model. The simplest way is to say that the verification is always in accordance with the requirements (technical terms) and validation versus specifications or user needs. The thing that the system is being build and verification if the system built correctly can express validation.

3. The proposed testing models

The mechatronic system is the resulting integration considered of the electrical / electronic system, mechanical parts and information processing. Therefore, in order to allow a systematic process of designing mechatronic systems with a high level of integration, the so-called multidisciplinary integrated design needed. However, neither the school nor the industry has yet provided an effective solution that can fully support the entire design process to achieve such an integrated multidisciplinary design. In order to organize design activities from different disciplines and to help designers achieve multidisciplinary integrated design, a design methodology based on a multidisciplinary interface model followed. [12]

In accordance with systems engineering practices, an extended "V"-model used as a macro-level process in the proposed design methodology. It starts with identifying system-wide requirements and ends with a user-validated system. The hierarchical design model adopted as a micro-level process. It supports specific design phases in which individual designers can structure design subtasks and can proceed and react in unforeseen situations. [13]

In order to ensure coherence and traceability between the two levels, the multidisciplinary interface is the model proposed. This design methodology is by studying demonstrated the process of designing a mechatronic system.

To test the state of a mechatronic system, any of the models presented above used. Of course, depending on the test model chosen, there are certain advantages or disadvantages. [14]

The presented design methodology adopts the extended "V"-model as a macro-level process and hierarchically the design model as a micro-level process. Multidisciplinary, the interface model helps designers ensure coherence between the two levels. [15]

3.1. The macro-level processes

The macro-level design process adopted an extended "V"-model process to present the overall flow for the mechatronic system. The left branch of the "V"-model extension represents the system design sub-process and described in qualitative models. After analyzing all the requirements for the whole system, sub functions and subsystems defined.

During the system, design sub-process, the design phases identified in specification, functional model and architectural phases. As represented in figure 4.

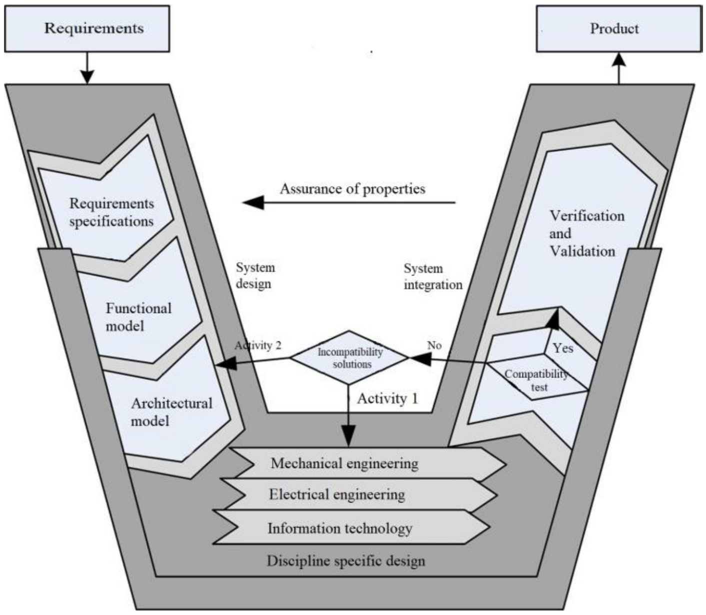

Fig. 4. The test model of a mechatronic system [16]

The right side of the "V"-model represents the sub-process integration system. In the extended "V"-model, this sub-process can be divided into two phases. These are the compatibility test and the verification and validation phase.

The compatibility objective of the testing phase is to guarantee and ensure correctly multidisciplinary integrated subsystems between different design teams. If the subsystems prove to be incompatible with each other, then the iterative process needs to be continued. The compatibility of the test in the early stages of system integration significantly reduce the number of iterations in the later phase. As a result, the overall development costs and time to market can decrease accordingly. Verification and validation is at this stage used to test the integrated performance of the system and verify that the system works and meets all the requirements previously proposed according to the designed plan. If the system needs to be improved, the previous design and phases repeats.

3.2. The micro-level processes

The hierarchical structure will be in the three phases of the system design sub-process applied. The design is parameter hierarchy proposed to help designers define key parameters in specific sub-process disciplines.

Regarding requirements, the specification derived from all mechatronic system and it can provide initial information about what customers are requesting. The requirements may apply to the general system, or to each subsystem (or component) or the interconnection between two subsystems. Therefore, the requirements detailed by breaking them down into additional ones, thus creating a hierarchy of them.

3.3. The functional model

A functional model refers to the modeling and specification phase of functional solutions. The functional model plays an important role during the system design process because built as a bridge between customers and the mechatronic system.

On one hand, the functions and sub functions proposed in the functional models are to meet customer requirements used. On the other hand, the architecture of the system is on the functional model established.

In the design phase, needs to be the functional models of mechatronic systems created. They need to meet the specifications and requirements to provide us with the basis for the functional structural derivation.

The hierarchical structure can also be for functional modeling used. The assumption is that a complex mechatronic system comprises a number of elementary functions then should be the functional structure of cooperation among these considered.

A single elementary function is characterized by the primarily use of a clearly defined effect (e.g. physical, chemical or biological) that can be considered as indivisible in the set of functions. Therefore, an architectural model consists of grouping the sub functions of the functional model in order to implement the proposed function.

3.4. The architectural model

After the process of hierarchical decomposition of the function, the designers should find the subsystems, which have implemented the proposed sub functions. In other words, these subsystems should have decomposed so that there is coherence between the functional model and the architectural model. Later, a complete architecture is by breaking down the subsystems into the phase of architectural model built. As a result, the decomposition process needs to be applied recursively. Once are the requirements placed in groups, they can be considered accordingly in specific sub functions. For example, the fault voltage is a sub-requirement of the safety requirements considered to avoid the state of breakdown (sub function). Therefore, the functional model built hierarchically according to the structure of the requirements specifications.

If all interfaces from the mechatronic system have been to be compatibility shown, which means that all components have been with each other integrated. Moreover, it needs the design process checked and validated. The verification and validation phase used to test the performance of the integrated system and it verifies whether the system performs the proposed function and meets all the previously proposed requirements. If the system needs to be improved, the initial operating phase repeats.

3.5. Achieving coherence between macro and micro design processes

Previously, were discussed the data of "V"-model and the hierarchical design model used as design processes at macro and micro-level.

The multidisciplinary interface is the model understood to use and support the ease of implementation of the system architecture during design at the model phase and to test the compatibility between subsystems at the compatibility testing stage. Achieving coherence between the two process models obtained from any direction, from the macro or micro-level.

3.6. From macro-level process to micro-level process: the hierarchical structure

The proposed extended "V"-model shows a general flow for the design process of mechatronic systems. It generally divided into three processes: system design, discipline specific design and system integration.

During the two design processes (system-specific design and discipline), each stage can be detailed by a hierarchical structure as a micro-level process. In the system design, hierarchical structure used to establish the requirements specifications, the functional model and the architectural model.

Moreover, it proposed to prioritize the design parameters with the help of designers to define key parameters from the design specific to the discipline. In general, the coherence between the macro-level process and the micro-level process ensured by the hierarchical structure during the two design sub-processes.

3.7. From micro-level process to macro-level process: the multidisciplinary interface model

This model provides a common representation for interfaces, which can be used both in the architectural phase-model of the system design as well as in the compatibility testing the system process at compatibility stage. In the architectural model phase, the three attributes of the multidisciplinary interface model can define an interface from different levels of decomposition or different disciplines with the same terminology.

The coherence determined between the model of the architectural design and the discipline-specific design sub-process system is by this common representation of the interface made.

In the compatibility test phase are tested the proposed-based interface of the multidisciplinary model. It is also tested the compatibility between subsystems (or components). This is helping designers to decide whether the design process can move to the next stage (verification and validation phase). Otherwise, it should return to the initial design phase, which refers to the discipline-specific process or the architectural model phase of the process system design.

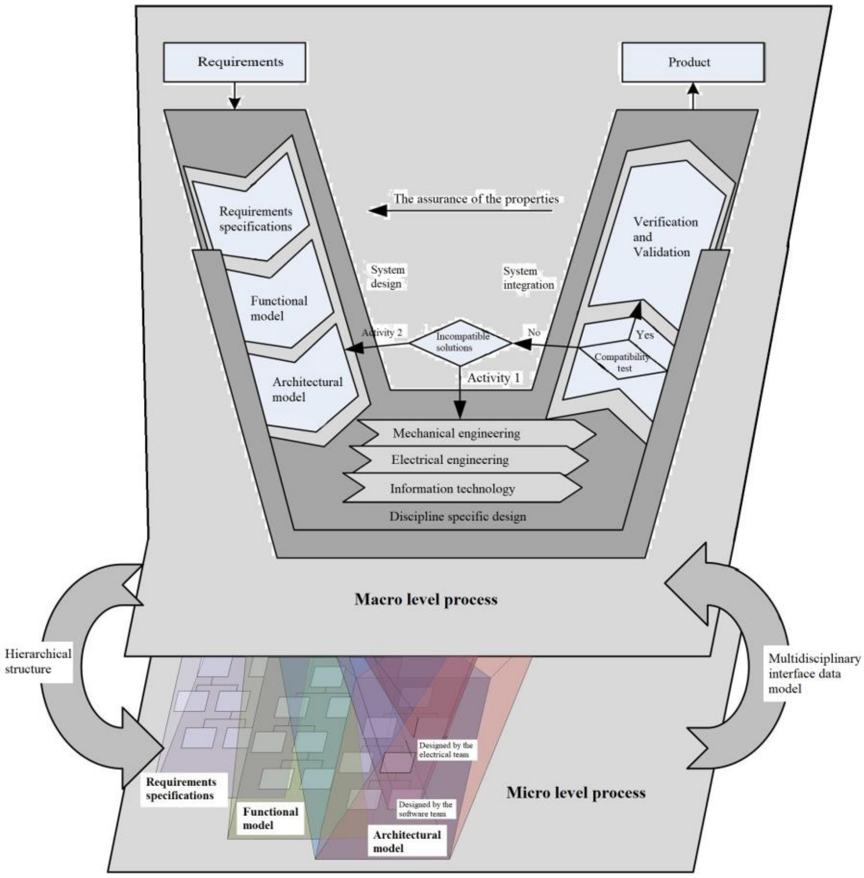

Fig. 5. Macro-micro link for the proposed model

Figure 5 shows the consistency between the two levels of processes. This consistency secure the hierarchical structure during the design of the system sub-processes and the design of the discipline-specific design process. During this time, it was designed an interface model during the system integration process.

The aim is to demonstrate the use of the proposed design methodology in a mechatronic design process. This type of approach can have applications for many mechatronic systems, thus facilitating the production process.

Conclusions

The requirements of verification and validation are a critical part of systems and software engineering. The importance of verification and validation (especially testing) is a major reason why the traditional cascade development cycle (Waterfall model) has undergone a minor change to create the "V"-model that links early development activities to their subsequent testing activities.

The "V"-shaped model should be for small to medium-sized projects used, where the requirements are clearly established; is utilized when ample technical resources are available, with the necessary technical expertise. To choose the "V"-shaped approach, a high level of customer trust is required. As no prototypes exists from before, there is a very high risk to meet customer expectations.

Another problem with "V"-model is that the distinction between unit testing, system integration and testing is not as clear as the model implies. For example, a number of test cases can sometimes be viewed as both unit and integration tests, thus avoiding the redundant development of associated test inputs, test outputs, test data, and test scripts.

However, the "V"-model is a useful way of thinking about development, as long as everyone involved (especially management) remembers that it is just a simplification of abstraction and does not intend to be a complete and precise model of a modern system or software development.

In the testing model, the macro-level design process is developed and based on the "V"-model, of which is considered as one of the more typical systems engineering approaches.

Therefore, the three main design processes (system design, discipline-specific and system design, and integration) of the macro-level process and organized sequentially in line with practical systems engineering.

However, designers from various disciplines perform concurrent design activities in each sub-process of the entire design process to achieve better design. The organization of design activities within the same process also referred as micro-level design process. This is the recommended way could be followed, in order to have during the fabrication process the mechatronic system tested from all point of view.

In the proposed model, presented as a model-test adapted to the mechatronic system. This model is on its interfaces created. The execution is systematically, as the integration of the product is. This approach based on the current models is, as the "V"-model could be adapted inside a mechatronic system.

REFERENCES

[1]. *Carryer JE, Ohline RM, Kenny TW* (2010) Introduction to Mechatronic Design, Prentice Hall, New Jersey, USA, ISBN 978-0-1314-3356-4, Ch. 2, 8-10, 13;

[2]. *Alciatore DG, Histand MB* (2012) Introduction to Mechatronics and Measurement Systems, Fourth Edition, McGraw-Hill, New York, USA, ISBN 978-0-0733-8023-0, Ch. 1-3, 6, 8, 9, 10;

[3]. *Abramovici M, Bellalouna F* (2007) Integration and Complexity Management within the Mechatronics Product Development, Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses, Proceedings of the 14^{th} CIRP Conference on Life Cycle Engineering, Waseda University, Tokyo, Japan, 11-13 June 2007, ISBN 978-1-84628-935-4, pp. 113-118;

[4]. *Ballas RG, Pfeiffer G, Werthschützky R* (2009*)* Elektromechanische Systeme der Mikrotechnik und Mechatronik, *2*. Auflage, Springer, Berlin/Heidelberg, Germany, ISBN 978-3-540-89320-2, Ch. 1, 3, 4;

[5]. *Fotso AB, Wasgint R, Achim R* (2012) State of the art for mechatronic design concepts, 8th IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA), Suzhou, China, 8-10 July, ISBN 978-1-4673-2347-5, pp. 232-240;

[6]. *Sellgren U, Törngren M, Malvius D, Biehl M* (2009) PLM for Mechatronics integration, Proceedings of the International Product Life Management Conference, Bath, UK, 6-8 July, ISBN 0-907776-50-7, pp. 106-116;

[7]. *Komoto H, Tomiyama T* (2011) Multi-disciplinary system decomposition of complex mechatronics systems, CIRP Annals, Vol. 60, Iss. 1, Elsevier, USA, ISSN 0007-8506, pp. 191-194;

[8]. *Spillner A., Linz T, Schaefer H* (2014) Software Testing Foundations: A Study Guide for the Certified Tester Exam, 4th Edition, Rocky Nook, California, USA, ISBN 978-1-937538-42-2, Ch. 3-6;

[9]. *Graham D, Black R, van Veenendaal E* (2019) Foundations of Software Testing: ISTQB Certification, Cengage Learning, Stamford, USA, ISBN 978-1-4737-6479-8, Ch. 2, 5;

[10]. *Alvarez Cabrera AA, Foeken MJ, Tekin OA et al* (2010) Towards automation of control software: a review of challenges in mechatronic design, Mechatronics, Vol. 20, Iss. 8, Elsevier, Amsterdam, Holland, ISSN 0957-4158, pp. 876-886;

[11]. *Gausemeier J, Dumitrescu R, Kahl S, Nordsiek D* (2011) Integrative development of product and production system for mechatronic products, Robotics and Computer-Integrated Manufacturing, Vol. 27, Iss. 4, Elsevier Ltd., UK, ISSN 0376-5845, pp. 772-778;

[12]. *Bishop, RH* (2005) Mechatronics An Introduction, CRC Press, Boca Raton, USA, ISBN 978-0-8493-6358-0, Ch. 1;

[13]. *Zheng C, Le Duigou J, Bricogne M, Hehenberger P, Eynard B* (2015) Multidisciplinary Interface Modelling: A Case Study on the Design of 3D Measurement System, Proceedings of 12th IFIP International Conference, October 19-21, Doha, Qatar, ISBN 978-3-319-33111-9, pp. 856-866;

[14]. *Bishop Gausemeier J, Frank U, Donoth J, Kahl S* (2009) Specification technique for the description of self-optimizing mechatronic systems, Research in Engineering Design, Vol. 20, Springer Verlag, Germany, ISSN 1435-6066, pp. 201-223;

[15]. *Gausemeier J, Moehringer S* (2003) New Guideline VDI 2206 - A flexible procedure model for the design of mechatronic systems, Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm Sweden, 19-21 August 2003, ISBN 1-904670-00-8, pp. 1-10;

[16]. *Zheng C, Hehenberger P, Le Duigou J, Bricogne M, Eynard B* (2017) Multidisciplinary design methodology for mechatronic systems based on interface model, Research in Engineering Design, Vol. 28, Springer, Berlin/Heidelberg, Germany, ISSN 1435-6066, pp. 333-356.