Estonian Journal of Engineering, 2008, **14**, 4, 303–316 doi: 10.3176/eng.2008.4.02

Conceptual design framework supported by dimensional analysis and System Modelling Language

François Christophe[a], Raivo Sell[b] and Eric Coatanéa[a]

[a] Department of Engineering Design and Manufacturing, Helsinki University of Technology, P.O. Box 4100, FIN-02015 HUT, Finland; francois.christophe@tkk.fi
[b] Department of Mechatronics, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia; raivo@staff.ttu.ee

Received 17 September 2008

Abstract. Early design is widely accepted inside the engineering design community as a crucial design stage. This is due to the fact that decisions taken at this stage constrain heavily the final performance of products. This article presents a design framework for the early design stage of mechatronic products. This framework provides a scientifically coherent methodology for refinement, analysis, modelling, comparison and evaluation of design solutions at early stage of the design process. A System Modelling Language (SysML) is proposed as a powerful modelling language properly adapted to mechatronic requirements. In addition, the article proposes to combine SysML with dimensional analysis and qualitative physics in order to provide a design tool able to carry out also early simulations, comparisons and evaluations.

Key words: system modelling, conceptual design, dimensional analysis, evaluation, unified design methodology.

1. INTRODUCTION

Early design stage is a fundamental phase of the design process. It has been shown [1,2] that 75% of the final cost of a product or service is constrained during the initial design phases due to decisions taken at this stage of the design process. The same analysis can be made for technical performance of machines or devices. Consequently, it is important to possess in the early design stage efficient modelling, comparison and evaluation tools. These tools should assist designers and other involved persons during the analysis and modelling stages. At the moment, research in engineering design has provided a significant number of practical design tools, but most of them are focusing on the later design stages (like embodiment and detail design). Existing tools for modelling, evaluation and comparison are characterized by the lack of commonly accepted fundamental scientific basis and by poor repeatability of there results. These drawbacks have been pointed out in [3].

This paper is an attempt to provide a coherent design methodology combining analysis, evaluation and comparison of design concepts. The scope of this paper is limited to mechatronic products but hopefully our approach is much broader and encompasses other design areas such as service and process design.

The paper is organized in the following manner. The second section is presenting basics of the System Modelling Language (SysML) expanded by the application-specific profile [4]. This language is an evolution of the Unified Modelling Language 2 (UML) and we aim at using it as a powerful tool for modelling mechatronic design problems.

The third section presents a methodology, based on dimensional analysis and multi-agent optimization, used for behaviour simulation of machines and also for comparing and evaluating different solutions. The mathematical apparatus, provided by dimensional analysis, can be fruitfully combined with the SysML modelling approach and provides a coherent framework for early design of mechatronic systems.

The fourth section considers integration of dimensional analysis into the modelling system.

The last section summarizes and considers problems for future research.

2. CONCEPT MODELLING WITH SysML TOOLBOX

According to the International Council on Systems Engineering [5], Systems Engineering is an interdisciplinary approach for the realization of successful systems. The whole design process focuses on defining customer needs and requires functionality in early stages of the development cycle, documenting requirements followed by design synthesis and system validation, considering the complete problem of operations, performance, testing, manufacturing, cost, schedule, training, support and disposal [5]. This definition points out the importance of early design and integrated activity very clearly setting high demands for modelling concepts and tools. Complex system design embraces several domains, which have their own tools and techniques, used for several years already.

In the software design world UML is the *de facto* standard for object-oriented software design. After UML 1.1 and UML 1.5, the most recent official version is now UML 2.1. The essence of software modelling (as of all modelling) is abstraction: the removal of fickle and distracting details of implementation technologies as well as the use of concepts that allow more direct expression of phenomena in the problem domain [6]. One of the recent trends is the increase of the role of software in everyday products. According to this, there is an increasing need for close communication between software design and conventional hardware design.

There have been several attempts to apply UML for non-software design in recent years. The important outcome is OMG SysML specification, finalized in 2007, which is initially derived from UML for System Engineers Request for Proposal (UML RFP) [7] in 2003. However there are several state-of-the-art works, based on the UML:

- UML Profile for Schedulability, Performance, and Time Specification [8];
- UML 2.0 Profile for Embedded System Design [9];
- UML Testing Profile [10];
- UML Profile for SoC (Systems on Chip) [11];
- UML 2 to Solve Systems Engineering Problems [12];
- UML for Hybrid Systems [13].

For mechatronic system design in general the SysML specification is a great tool for modelling and representation of the systems in early design and later. SysML reuses a subset of UML 2 diagrams and augments them with some new diagrams and modelling methods appropriate for systems modelling. SysML is designed to complement UML 2; thus systems engineers, who are specifying a system with SysML, can collaborate efficiently with software engineers, who are defining a system with UML 2 [14]. Four pillars of SysML are shown in Fig. 1.

In mechatronics and system engineering very wide range of applications can be considered. Different products and domains have their own specifics and therefore it is necessary to customize general system modelling tools to meet the specifics of the particular application domain. At the same time the connections and compatibility have to be preserved. UML and SysML have the profiling mechanism to extend or restrict the initial language constructs, ensuring the required compatibility at the same time. Further we explain the SysML toolkit, which consists of a SysML profile for mobile platform development in conceptual stage as an application example.

The toolkit is defined as a SysML profile and external simulation package. The profile itself consists of template libraries, diagram extensions and model libraries. Standard model libraries are *Principle*, *Terrain* and *ContactType*.

The model library *Principle* is a collection of standard mechatronics subsystems, elements and working principles. This library is most similar to the existing design software library, where standard parts are defined and collected into packets. The Mobile Platform Toolkit (MPT) *Principle* library consists of the working principles and subsystems formulated in SysML and extended profile. This means that similar subsystems can be found in different libraries, although the abstraction level is different. The subsystem is defined in formal language rather than as a physical component. The boundaries between the physical domains are not precisely defined and can be determined later at the detailed design stage. The model can be developed by linking the subsystems and working principles from the library with loosely coupled relations whereas certain key parameters are defined. These parameters are in most cases derived from the requirement model and are related with many other parameters of the system. For example, simple mathematical model, linking different parameters is

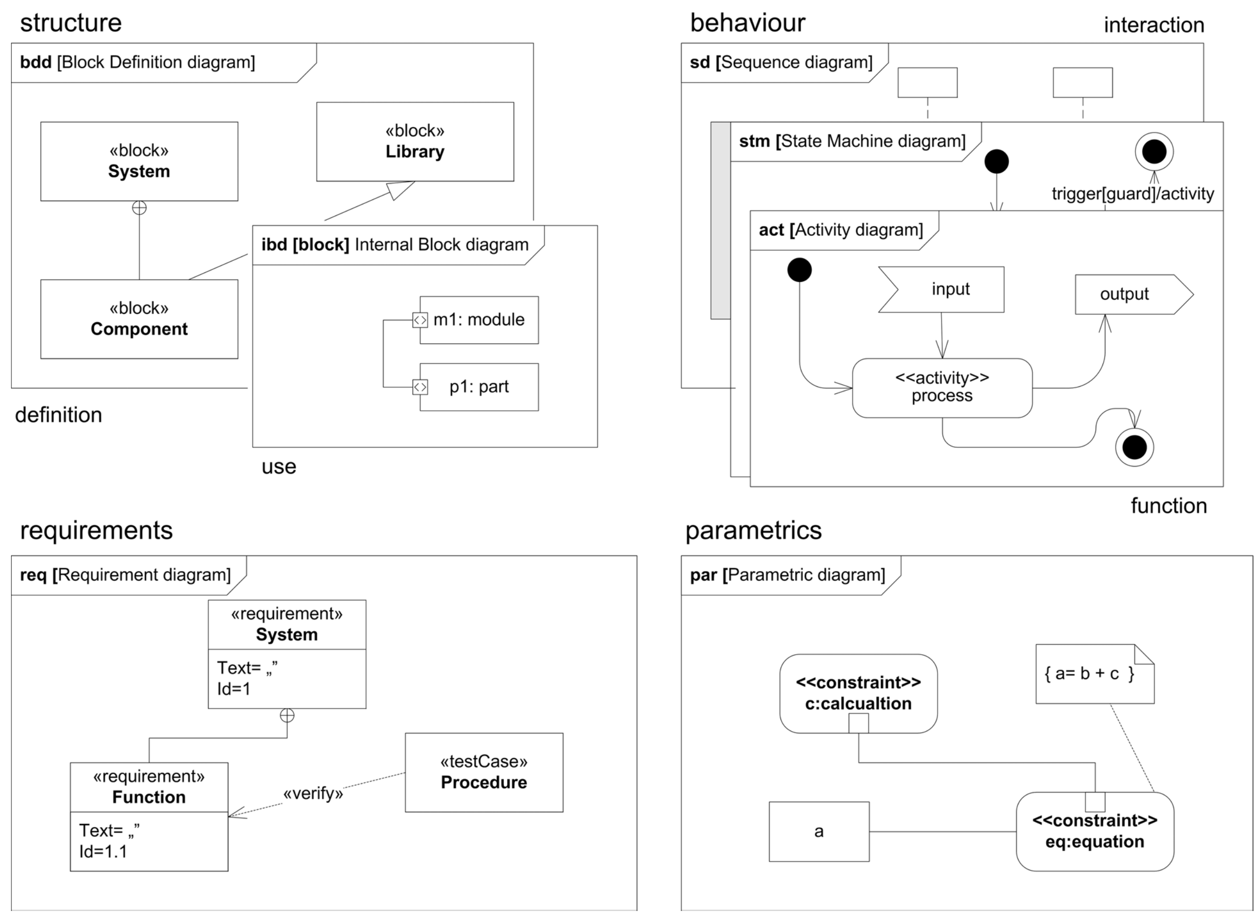


Fig. 1. SysML pillars [15].

defined by *Parametric* diagram and key parameters are defined with extended stereotypes. The general structure of the toolkit is shown in Fig. 2. This figure presents the toolkit structure of a mobile robot platform.

Terrain and *ContactType* libraries are holding the parameters of different terrain and vehicle-soil contact. The reason for establishing the *Terrain* and *ContactType* libraries was the mobile platform performance analysis and simulation need. Depending on the required terrain capabilities, the mobile robot must deal with obstacles, surface characteristics, slopes, etc. Terrain properties are important in robot design since smart and optimal design can save energy, improve the performance, optimize the budget etc. These parameterized models can be linked to the design element or design candidate and used in initial simulations.

The conceptual modelling exploits several SysML-defined diagrams with extended toolkit objects. Toolkit specifies the modelling steps and appropriate diagrams according to the application. In Fig. 3, the system main services are modelled in Use Case diagram where MPT-specific stereotypes are used. For the structure and behaviour similar diagrams are constructed. The toolkit specification has been further studied in [[4]].

The simulation is usually used at the later design stage where the system model is relatively precisely defined. To get the maximum benefit, the proposed design framework includes the simulation into the conceptual design stage. The model (structure and behaviour) consists of special block elements stereotyped as *simu*. An example is shown in Fig. 4, where *simu* block is the control algorithm of a robot, controlling the leg and wheel motors according to terrain changes. The *ControlFPGA* block is a link to the simulation model. Simulating the control algorithm, the engineering team gets the feedback of critical component parameters required to fulfill the initial requirements or simulating different algorithm candidates determining the system feedback.

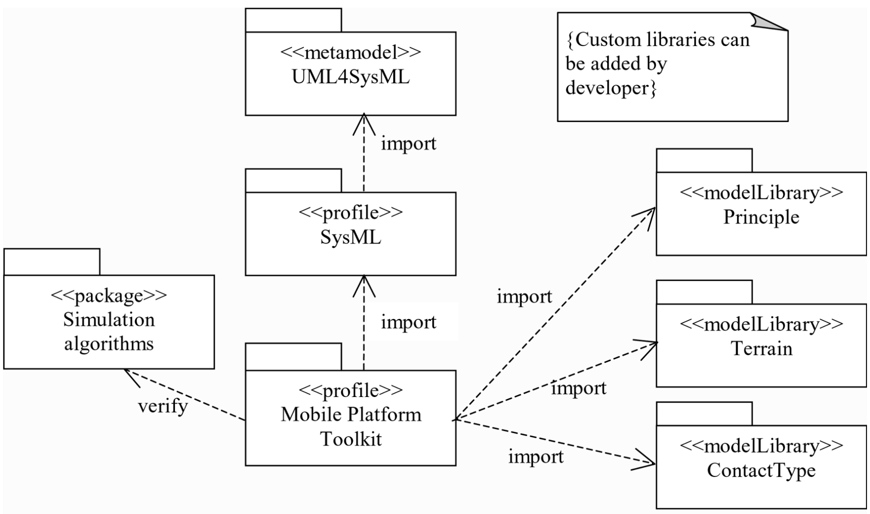


Fig. 2. Structure of the Mobile Platform Toolkit.

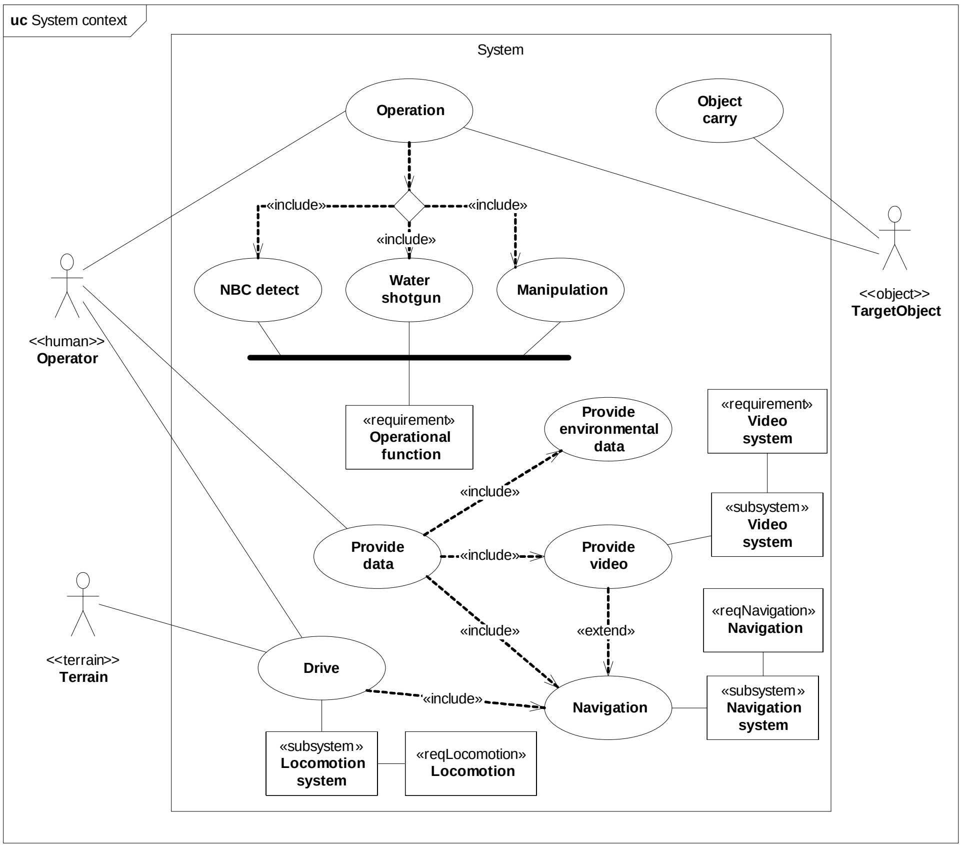


Fig. 3. System services.

An important aspect of early design is to develop several solution candidates. Traditionally it has been done manually by creating models with later analysis of their features. Recently many non-traditional techniques and methods for solving engineering problems have been developed. One of the reasons is definitely the increase of the computing power. That allows us to solve engineering tasks, which cannot be described with linear differential equations and are non-deterministic. The techniques, applicable for more advanced generation and evaluation of mechatronic systems, are the following:

– multi-agent systems;
– genetic algorithms/genetic programming;
– neural networks;
– fuzzy logic.

These methods have been successfully applied in several cases for solving specific problems of optimization, machine learning, adaptive control, path planning, etc. For example, fuzzy logic is widely used in controller systems and neural networks by parameter prediction. However, in many cases the theory is

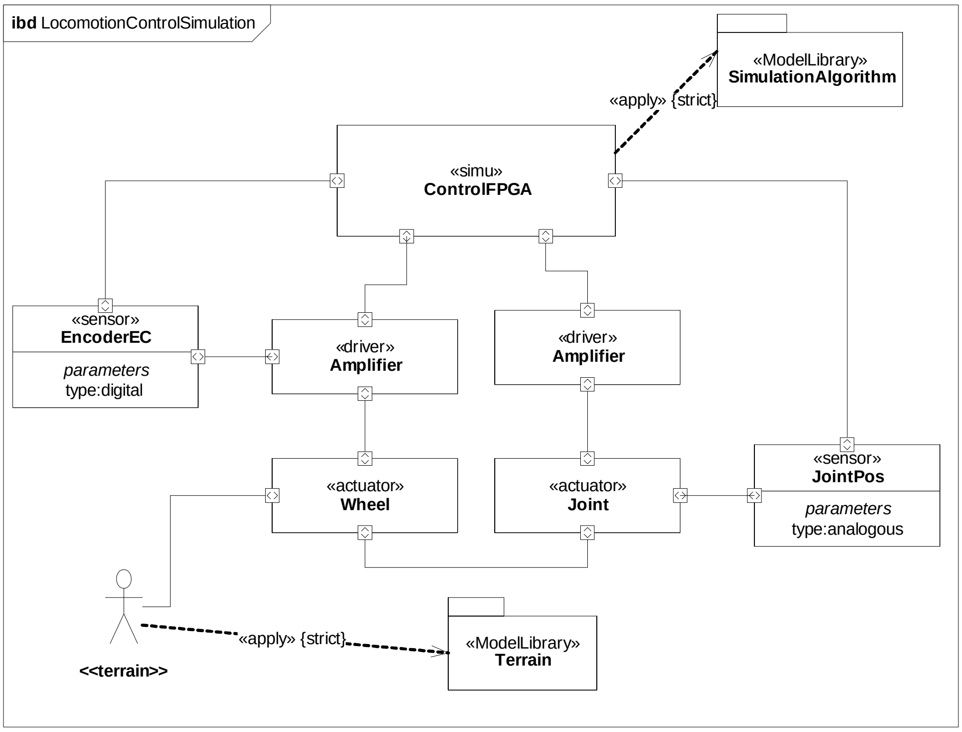


Fig. 4. Simulation block in the diagram of the system structure.

applied only in computer environment, calculating or simulating a certain problem. Genetic algorithms are often used for finding global optimum in case of great state space. The advantages of artificial intelligence methods over the traditional ones are the ability to search over the entire solution space. They are applicable to a wide range of problems including non-continuous functions and functions, involving different types of variables. There has been a limited number of attempts of exploiting above techniques for the generation of design solutions. Some papers [16–18] have shown the possibility to apply the multi-agent system, genetic programming and bond graph combination to automate the generation of the initial system concept. SysML modelling toolkit can be combined with a theoretical approach for early evaluation and comparison. This theoretical framework can provide a useful complement to the modelling approach in order to qualitatively simulate, compare and evaluate solutions.

3. SIMULATION AND COMPARISON OF CONCEPTS WITH DIMENSIONAL ANALYSIS

3.1. Dimensional analysis and behaviour simulation

3.1.1. Basis of dimensional analysis

Dimensional analysis (DA) is a field of qualitative physics, which considers units and magnitudes. DA is often used in order to verify the dimensional homogeneity of physical equations but its scope is much broader. Similarity between scales is a major area of applications [19]. The fields of application are numerous (electromagnetic theory, aerodynamics, aeronautics etc.). DA mostly relies on the Vashy–Buckingham theorem, which states that the study of a physical problem, expressed with *n*-dimensional quantities, can be reduced by a factor k when expressed in a dimensionless form. Dimensionless numbers such as Reynolds and Froude numbers follow from the DA method. Bashkar and Nigam have provided a *machinery* to allow the use of DA in the analysis of a mechanism [20]. This *machinery* provides powerful tool for the behavioural simulation of a mechanism. Furthermore, it has been proved in [21] that under certain conditions, there exists a formal link between the topological structure of a design and the metric space provided by DA. Thus DA can be used in conceptual design for simulation and comparison purposes.

3.1.2. Computation of dimensionless numbers

The Vashy–Buckingham theorem does not provide any specific guidance related to the choice of the variables used for the reduction of the problem. In order to enable systematic computation of dimensionless numbers, we consider the input and output variables of a concept as performance variables. Then the choice of repeating variables should be done within the concept's internal variables and according to the unique number of the system's governing dimensions.

This systematic computation can be done according to Butterfield's paradigm [22]. This paradigm is used in order to select the minimum set of repeated variables, which ensures the non-singularity of the metrization procedure. This procedure provides one dimensionless group for each concept. The practical computation of dimensionless numbers is described in [21].

3.1.3. Simulation of the behaviour of a concept

The simulation of the behaviour of a concept of solution is the immediate result of the dimensionless group computation. In fact, the dimensionless numbers computed for one concept allow us to qualitatively show the evolution of each variable according to the variation of the other variables [20].

As an example, we can consider an electrical battery and simulate its charging phase. In this example we consider the following variables: U – potential of the battery, I – its charging intensity, E – the energy stored, Ω – its internal resistance, ρ_V – the volume density of the battery and ρ_M – its mass density.

For that device, the variables of interest are U and I, the other ones being internal variables. DA gives us two dimensionless numbers:

$$\pi_U = UE^{-1}\Omega^{-1/2}\rho_V^{1/2}\rho_M^{-1/4}, \tag{1}$$

$$\pi_I = IE^{1/3}\Omega^{1/2}\rho_V^{-5/6}\rho_M^{-1/4}. \tag{2}$$

From this dimensionless group, we can simulate the behaviour of a certain type of battery during the charging phase, considering Ω, ρ_V and ρ_M as known. Indeed if the battery is charging, U should increase. An increase of U implies an increase of the amount of energy stored E. From π_I, we can deduce that an increase of E will lead to a decrease of the intensity of charge I. This example efficiently reflects the normal behaviour of a battery being charged. The simulation procedure can be generalized to any kind of complex mechanisms and can explain qualitatively their physical behaviour [[20,21]]. This is a part of the theoretical background, based on the principle of similarity, which allows early simulations of complex mechanisms. The similarity principle can also be used for the comparison of the concepts of solutions. This is the goal of the following section.

3.2. Principle of similarity and comparison of concepts of solutions

3.2.1. Similarity principle

In order to be comparable, two concepts of the solution should share the same function and provide the same type of output variables. This means in practice that the dimensionless numbers, involving output variables, should be equal regardless of the internal variables of the concepts. This is the similarity principle [[23,19]].

If we consider two concepts π_1 and π_2, sharing the same type of variables, the similarity principle can be expressed as

$$\pi_1 = A^{\alpha}B^{\beta}C^{\chi}\dots X^{\zeta}. \tag{3}$$

If the scales of the parameters vary from one machine to another, we have

$$\pi_1' = \left(\frac{A}{m}\right)^{\alpha}\left(\frac{B}{n}\right)^{\beta}\left(\frac{C}{o}\right)^{\chi}\dots\left(\frac{X}{p}\right)^{\zeta}, \tag{4}$$

where m, n, o and p are the scale ratios.

In order to meet the similarity conditions π_1 and π_1', we need to fulfill the following condition, which follows from Eqs. (3) and (4):

$$A^{\alpha}B^{\beta}C^{\chi}\dots X^{\zeta} = \left(\frac{A}{m}\right)^{\alpha}\left(\frac{B}{n}\right)^{\beta}\left(\frac{C}{o}\right)^{\chi}\dots\left(\frac{X}{p}\right)^{\zeta}. \tag{5}$$

This means that the similarity condition is

$$m^{\alpha} n^{\beta} o^{\chi} \ldots \rho^{\zeta} = 1. \tag{6}$$

This simple case can be generalized for concepts in the case when π_1 is expressed using different types of variables.

3.2.2. Method of comparison

In order to compare different concepts, we define an ideal concept (i.e., a usual approach used in multi-objective optimization) according to ideal target values of the performance variables. The procedure of comparison can be done between the ideal concept and real concepts, respecting the principle of similarity. The aim is to define for a real concept the real values of the performance variables both *approaching the ideal values* and *meeting the similarity principle*. This approach leads to a combinatorial optimization procedure. The complexity of this problem grows exponentially with the amount of performance variables.

3.2.3. Agent-based optimization

Multi-agent systems may be efficient in multi-objective optimization problems. Our aim is to use them to tackle the complexity of the optimization procedure described above. The agent-based method that we propose in this paper can be considered as a set of concepts, for which we try to find optimal values for the variables according to performance constraints. This method allows us to avoid any kind of weighting approach commonly used in design, which is a source of subjectivity in the selection and evaluation of concepts.

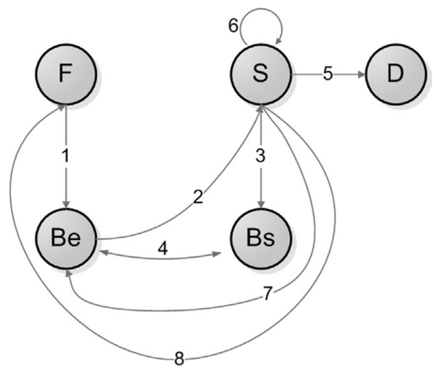
Indeed, each attribute of performance is in the first step supposed to have the same importance. In the second step importance of the attributes can be differentiated. The multi-agent optimization procedure is a powerful method to explore the design space.

4. INTEGRATION OF DIMENSIONAL ANALYSIS IN SYSTEM MODELLING

4.1. Semantic unification

In order to combine methods described above, we have to unify the terminology. This unification is in accordance with the Function-Behaviour-Structure framework [24]. Figure 5 shows conceptual design as a set of eight processes, which allows us to position clearly different SysML model diagrams and the use of DA in the evaluation process. This overview of conceptual design is focused on three classes of variables used to describe different aspects of a design object:

- function variables (F) describe the purpose of the object;
- behaviour variables (B) describe the attributes, derived from the structure of the object or the attributes, expected to be derived from it;
- structure variables (S) describe the components of the objects and their relationships.



Legend	Associated diagram or method
1 Formulation	*Requirements* and system context by *Use Case*
2 Synthesis	Architecture, *Block Definition* and *Internal Block* diagrams of models
3 Analysis	Butterfield's paradigm and *Parametric* diagram
4 Evaluation	Qualitative simulation and comparison with reference, *Parametric* with external simulation algorithm
5 Documentation	Modules for automatic documentation (XMI…), SysML model
6, 7, 8 Reformulations	Verification: *Activity* diagram as *TestCase*

Fig. 5. Conceptual design, the corresponding SysML diagrams and DA module.

In Fig. 5. Be represents the expected behaviour of the object. This expected behaviour is derived from the object's function. The expected behaviour variables are represented by SysML *Requirements*.

On the other hand, Bs represents the "actual" behaviour of a concept of the solution, e.g. the behaviour derived from its structure. The actual behaviour is represented by SysML *Activity* diagram.

4.2. Insight provided by dimensional analysis

Figure 5 highlights the recursive aspect of the conceptual design process. This aspect is due to the dynamic property of design. In fact, the creation of a new product modifies the global design environment so that the requirements and needs are reformulated. Thus the application of adaptive tools such as multi-agent systems during the conceptual phase of design is justified.

A critical issue is the necessity of exhaustive functional requirements. Indeed, well-defined and structured requirements permit to avoid useless iterations of the design process and help providing a highly performing object. To our opinion defining the key criteria and key performances of the object should be the main focus by design. SysML *Requirement* diagram offers a suitable structure for this purpose because it allows hierarchical structuring of the requirements, according to the importance given to a function. The *Requirement* diagram allows us to compute dimensionless groups, representing the expected behaviour of the object. As shown in Fig. 6, *Parametric* diagram is used to verify the good

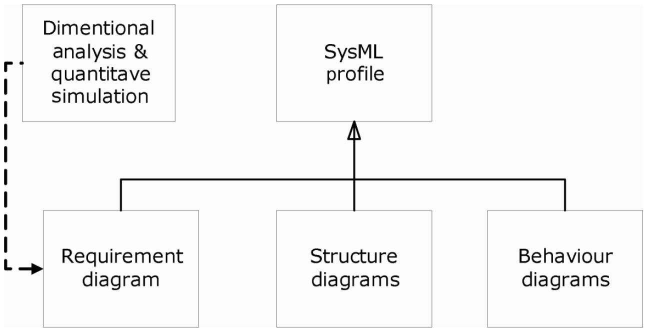


Fig. 6. Integration of dimensional analysis and SysML for the model evaluation purpose.

behaviour of a detailed model while dimensional analysis enables evaluation of early models and to compare their derived behaviour with the expected one. Additionally, during the early stage of design, *Parametric* diagrams might not be possible to describe due to the inherent lack of information at that stage. On the opposite, qualitative simulation will give designers a first glance on the shape of parametric equations [[20]].

5. CONCLUSIONS

This paper has presented the initial development of a synthetic approach to refining, creating and evaluating solutions during the early design process. The synthetic approach was dedicated to mechatronic systems. The method relies on SysML and a toolkit used for modelling and refining the design problem, dimensional analysis and qualitative physics used for comparing, evaluating and simulating the solutions. The method is scientifically coherent and based on proved scientific concepts. The approach is aimed at guiding the designers from the validation of the needs to the comparison and evaluation of mechatronic solutions. We have described the general approach and not considered details of application. The SysML has been developed specifically for systems modelling. In the same vein, dimensional analysis, principles of qualitative physics and extensive use of the concept of similarity is novel in the sense that it has never been used in a systematic manner for design purposes. This paper should be viewed as an initial attempt to provide a complete early design framework for mechatronic systems.

Semi-formal languages, such as SysML, have their limitations due to their imprecision, particularly in the behavioural description of the model. Our future interests will concern continuity questions between different states of a machine. We assume that using DA could allow finding of the threshold values of variables describing the transition between two states. This is a demanding issue as DA is meant for qualitative consideration and thus, passage from qualitative to quantitative consideration is at the limit of DAs application.

In future an integrated software environment is to be developed to improve the usability of the presented methodology.

ACKNOWLEDGEMENTS

Raivo Sell has been supported by the Estonian Ministry of Science and Education (grant No. 0142506s03) and by Estonian Science Foundation (grant No. 7542). The work of Eric Coatanéa has been performed within the research project COMODE. The COMODE project has received research funding from the EIF Marie Curie Action, which is part of the European Community's Sixth Framework Program.

REFERENCES

1. Lotter, B. *Manufacturing Assembly Handbook*. Butterworths, Boston, 1986.
2. Hsu, W. and Woon, I. M. Y. Current research in the conceptual design of mechanical products. *Computer-Aided Design*, 1998, **30**, 377–389.
3. Blessing, J. Consolidation of design research: the issue of design theory. In *International Conference on Design*. Dubrovnik, 2006.
4. Sell, R. *Model Based Mechatronic Systems Modeling Methodology in Conceptual Design Stage*. TUT Press, Tallinn, 2007.
5. *Systems Engineering Handbook*. INCOSE-TP-2003-016-02, Version 2a, Technical Board of International Council on Systems Engineering (INCOSE), 2004; http://www.incose.org
6. Booch, G., Rumbaugh, J. and Jacobson, I. *The Unified Modeling Language User Guide*. Addison Wesley, MA, 1999.
7. *UML for Systems Engineering RFP*. OMG document ad/03-03-41, 2003; http://syseng.omg.org/
8. *UML Profile for Schedulability, Performance, and Time Specification*. OMG document ptc/2003-03-2, 2003; http://syseng.omg.org/
9. Kukkala, P., Riihimäki, J., Hännikäinen, M., Hämäläinen, T. D. and Kronlöf, K. UML 2.0 profile for embedded system design. In *Proc. Design, Automation and Test in Europe Conference*. Munich, 2005, 710–715.
10. *UML 2.0 Testing Profile Specification, version 1.0*. OMG document formal/05-07-07, 2005; http://syseng.omg.org/
11. Rajan, S. P., Hasegawa, T., Shoji, M., Zhu, Q. and Nakata, T. UML profile for SoC RFC. *DAC 2005 Workshop, UML-SoC 2005 UML for SoC Design Conference*. Anaheim, 2005.
12. Gurd, A. Using UMLTM 2.0 to solve systems engineering problems. Telelogic. 2003; http://whitepapers.zdnet.co.uk
13. Berkenkötter, K., Bisanz, S., Hannemann, U. and Peleska, J. HybridUML profile for UML 2.0. *Int. J. Software Tools Technol. Transfer*, 2006, **8**, 1–36.
14. *System Modeling Language (SysML) Specification. Version 1.0 Draft*. OMG document ad/2006-03-01, 2006; http://www.sysml.org
15. Sell, R. and Tamre, M. Integration of V-model and SysML for advanced mechatronics system design. In *Proc. Research and Education on Mechatronics Conference REM05*. Annecy, 2005, 276–280.
16. Rzevski, G. On conceptual design of intelligent mechatronic system. *Mechatronics*, 2003, **13**, 1029–1044.
17. Granda, J. J. The role of bond graph modeling and simulation in mechatronics systems. An integrated software tool: CAMP-G, MATLAB–SIMULINK. *Mechatronics*, 2002, **12**, 1271–1295.

18. Seo, K., Fan, Z., Hu, J., Goodman, E. D. and Rosenberg, R. C. Toward a unified and automated design methodology for multi-domain dynamic systems using bond graphs and genetic programming. *Mechatronics*, 2003, **13**, 851–885.
19. Sonin, A. A. *The Physical Basis of Dimensional Analysis*, 2nd ed. Department of Mechanical Engineering, MIT, Cambridge, MA, 2001.
20. Bhashkar, R. and Nigam, A. Qualitative physics using dimensional analysis. *Artificial Intelligence*, 1990, **45**, 73–111.
21. Coatanéa, E. *Conceptual Design of Life Cycle Design: A Modeling and Evaluation Method Based on Analogies and Dimensionless Numbers*. Helsinki University of Technology, Espoo, 2005.
22. Butterfield, R. Dimensional analysis revisited. *J. Mech. Eng. Sci.*, 2001, **215**, 1365–1375.
23. Matz, W. *Le principe de similitude en génie chimique*. Dunod, Paris, 1959.
24. Gero, J. S. and Kannengiesser, U. The situated function-behaviour-structure framework. *Design Study*, 2004, **25**, 1–25.

Kontseptuaalse projekteerimise raamistik, kasutades dimensionaalset analüüsi ja süsteemi modelleerimise keelt

François Christophe, Raivo Sell ja Eric Coatanéa

On käsitletud projekteerimise varajase staadiumi metoodikaid ja välja arendatud uudne lähenemine projekteerimise kontseptuaalse faasi modelleerimiseks ning optimaalse lahenduse poolautomaatseks genereerimiseks. Lahenduste simuleerimiseks ja võrdlemiseks on kasutatud dimensionaalset analüüsi ning käitumise simuleerimist. Süsteemi modelleerimiseks on kasutatud uudset süsteemi modelleerimise keelt SysML. On käsitletud nende kahe lähenemise integreerimise võimalusi, probleeme ja positiivset efekti. Rakendusnäitena on vaadeldud mobiilset robotit.