# A Nonparametric Evaluation of SysML-based Mechatronic Conceptual Design

Mohammad Chami [a] Haitham Bou Ammar [b] Holger Voos [c] Karl Tuyls [b]
Gerhard Weiss [b]

[a] *Institute of Applied Research, Uni. of Appl. Sciences Ravensburg-Weingarten, Germany*
[b] *Department of Knowledge Engineering, Maastricht University, The Netherlands*
[c] *Research Unit in Engineering Science, University of Luxembourg, Luxembourg*

**Abstract**

Mechatronic technologies are used in a wide range of industries, from aerospace to automotive, manufacturing and even to personal devices, such as cd/dvd players. Although their multidisciplinary nature provides great functionalities, it is still one of the substantial challenges which frequently impede their design process. Apart from this problem, an early system design evaluation while adhering to adaptable design requirements is still missing. In this paper we propose a SysML-based method for an Intelligent Conceptual Design Evaluation of mechatronic systems, abbreviated as SysDICE. Particularly, we contribute by, firstly, making use of SysML as a common modeling language for the engineering team involved in the design process and secondly, by adopting a widely used, in artificial intelligence, pattern recognition tool, namely non-parametric regression, to support a multi-alternative design mechanism, with the aim of attaining the best combination of components' alternatives that suits a set of prioritized numerical requirements. To evaluate our framework, we have conducted two design experiments: (1) a two-wheel differential drive robot, and (2) a quad-rotor unmanned aerial vehicle. Results prove how our framework can assist system engineers and support the design process.

## 1 Introduction

The system design phase of mechatronics typically exhibits a multidisciplinary nature by aggregating various engineering disciplines (i.e., mechanical, electrical, software and control), project and business management fields. From a system engineering perspective [1], system engineers must have a broad knowledge on the system from the end-user domain till the system's technical engineering domains. This nature imposes a substantial challenge that deals with integrating the involved human factors with their methodologies, modeling languages and software tools for the aim of attaining an efficient system design.

In theory the course of system design from idea creation to product disposal has been successfully proposed (e.g. [2, 3]). However, the industrial development techniques are still mono-disciplinary [4]. Particularly, the integration phase of the different disciplines' outcomes yet arrive at later stages, which makes the procedure expensive, cost and time in-efficient. Moreover, in reality, a document-based manner has been followed to hold the disciplines' interdependencies (i.e., how, when and in what way any discipline influences another). This frequently leads to weak synchronization between the interdependencies of entities and can result in inefficiencies that often appear during the integration and testing [1]. Therefore, an early integrated evaluation of the system, as a whole, is strongly demanded.

So far, little attention has been given to the collaborative work for evaluating designs in a sequel of making the procedure adaptable, efficient, and intelligent. In this paper we target the previously mentioned problems and contribute by: (1) capturing the interdisciplinary information across system engineers and designers using SysML to generate a system design model, which is (2) mathematically formulated for the aim of the satisfaction of a set of prioritized numerical requirements by (3) adopting non-parametric regression. In this way the system design procedure is more adaptable and coherent.

# 2 Related Work

Multidisciplinary approaches in mechatronic design have been frequently discussed in research. For instance, in [5] a high-level system model is presented and in [6] a constraint modeling-based approach is described. Although, these approaches contribute greatly, unfortunately, they are ungeneralizable, where previously unconsidered disciplines can be hardly integrated later. To solve this generalization issue, different approaches have been deployed. SysML [7] is one of these general-purpose approaches. For instance in [8], SysML was used to specify the central view-model of the mechatronics system. In [9], the system-level modeling with SysML was adopted to support mechatronic design. While in [10], SysML profiles were particularly applied to support the multi-view modeling approach.

From a requirements engineering point of view, various methods dealing with requirements analysis and traceability have been proposed. However, the mapping between requirements and system design model entities (i.e., components, properties) still rarely exists and, even if it did, it requires high synthesis and modification effort. Although SysML supports in modeling this mapping, its execution is still an open topic.

These previous works as well as others have contributed to the maturity of SysML. However, SysML does not have a formal semantic, is solely useful for project specific intentions, and lacks support of generalized execution. Extending our previous work [11, 12], while focusing on the system engineering level, we generalize the previous approaches by providing a mathematical formulation of the technical and economical aspects to support SysML execution and thus interoperability among the different design disciplines.

Artificial Intelligence (AI) methods have been proposed to aid the mechatronic design process. For instance, in [13] the design activity optimization was solved using a heuristic-based hybrid search algorithm and in [14] a maximum likelihood estimation method for determining the unknown design parameters based on given information was proposed. The main problems in existing approaches are twofold: (1) high effort in capturing the interdisciplinary information to be used in AI, and (2) problem specific design modeling and optimization, due to the adoption of parametric techniques. In our work we generalize the previous approaches, where we reduce the effort in providing the knowledge needed for AI through the proposed SysML model, and use non-parametric regression techniques to provide a problem independent design framework.

# 3 Background Preliminaries

This section presents background material needed to understand the remainder of the paper.

## 3.1 Mechatronic System Design

In theory, the VDI 2206 guideline [3], is one of the popular exemplifications of the mechatronic design process. It supports the creation of an interdisciplinary principal solution during the V-model's system design phase. Traditionally, during the initial design stages, the requirements are captured, categorized and analyzed. Therefore, modeling, analysis and simulations are the main activities performed in any mechatronic design methodology to assess a set of demanded requirements. Apart from the methodological aspects, different engineering tools are being employed and can be categorized into three types: (1) domain-specific tools (e.g., circuit design tools, software engineering tools, mechanical CAD tools), (2) domain-coupling tools (e.g., MATLAB, Modelica), and (3) all-in-one tools (e.g., Mechatronic Concept Designer).

In order to describe the disciplines' interdependencies between the tools, a document-based approach has been followed, such as Excel sheets, MS word, and/or PowerPoint files. This issue has been the reason of many project failures due to the lack of traceability and enactment of these interdisciplinary entities. Thus, this approach was over-thrown by the *Model-Based Systems Engineering* (MBSE) methodology. Here, models are used to represent such interdependencies and are intended to facilitate the design activities thus resulting in better communication, system design integration and system reusability [1].

## 3.2 Systems Modeling Language (SysML)

SysML is a *"general-purpose graphical modeling language"* [7]. It is developed as a software engineering extension of a customized subset of the *Unified Modeling Language* (UML) with the goal of being applied for systems' engineering applications. SysML captures the multidisciplinary knowledge by providing various diagrams: block definition, internal block, parametric and package diagrams to present the *structure* of the system. It further delivers activity, sequence, state machine and use case diagrams to describe the *behavior* of the product. Finally, with its major contribution, it allows for modeling the *requirements* of a system with the aid of its requirements' diagrams. It also integrates the previous three aspects (i.e., structure, behavior, and requirements) through allocations across their corresponding elements. SysML further offers a profile mechanism, where a profile is formed from a set of stereotypes of its elements. These stereotypes extend the syntax of SysML allowing it to be more applicable in concrete applications.

### 3.3 Gaussian Processes

Gaussian Processes (GPs) are a form of non-parametric regression techniques. Following the notation of [15], given a data set $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{m}$ where $\mathbf{x} \in \mathbb{R}^d$ is the input vector, $y \in \mathbb{R}$ the output vector and $m$ is the number of available data points when a function is sampled according to a GP, we write, $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $m(\mathbf{x})$ is the mean function and $k(\mathbf{x}, \mathbf{x}')$ the covariance function, fully specifying a GP. Learning in a GP setting involves maximizing the marginal likelihood of Equation 1.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} - \frac{1}{2}\log|\mathbf{K} + \sigma_n^2 \mathbf{I}| - \frac{n}{2}\log 2\pi, \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^{m \times 1}$ is the vector of all collected outputs, $\mathbf{X} \in \mathbb{R}^{m \times d}$ is the matrix of the data set inputs, and $\mathbf{K} \in \mathbb{R}^{m \times m}$ is the covariance matrix with $|.|$ representing the determinant. GPs automatically avoid over-fitting due to the presence of the second term in Equation 1 (i.e., $\frac{1}{2}\log|\mathbf{K} + \sigma_n^2 \mathbf{I}|$). Due to space constraints we refer the interested reader to [15] for a thorough discussion of the topic.

# 4 The Need for a Unified Language and Adaptation

We adopt the V-model suggested by the VDI 2206 guideline [3], shown in Figure 1, as a macro-cycle consisting of requirements analysis, system design, domain-specific design and system integration phases that end with the product disposal. Despite the V-model's support for modeling and model analysis, the whole process is currently a theoretical construct without tool support. In addition, different gaps and short comings exist among the employed models as shown in Figure 1. These gaps affect the traceability and impede in updating the actuality of the different entities across the phases. Previous experience [12] has shown how the application of SysML in documenting such interdisciplinary relationships in a system model can glue these gaps.

These gaps could be traced back to problems in communication and integration among the different disciplines due to the lack of an efficient system model. In addition, another problem is the lack of adaptability and generalizability in the design process.

To solve these problems, we make use of SysML, as a common modeling language, to model three aspects of the system's design. Namely, we use the requirements (req), block definition (bdd), and parametric (par) diagrams to model the system's requirements, structure, and constraints respectively. In the sequel of making the design process adaptable to changing requirements and/or priorities as well as to support a multi-alternative design platform, we make use of GPs and optimization. In the following, the technical details will be further explained.

# 5 AI Support for Mechatronic System Design

In this section we will detail the proposed framework, to: (1) capture the interdisciplinary knowledge among the different involved domains, (2) provide the mathematical formulation of the requirements satisfaction problem, and (3) reflect upon the GP approximation used in our platform.

### 5.1 SysDICE Overall Framework

Figure 2 presents a high level scheme of the proposed framework. We categorize the human factors involved into (1) Discipline and (2) System engineers. For the first group, a discipline-specific information can be represented in SysML while assuring that the SysML detail level is restricted to only the amount of information needed for achieving a cross-discipline mapping. For the second category, system engineers, can model system requirements, the abstract conceptual solution and manage the system model using SysML. They are able to evaluate the system design model through MATLAB which is running in the background to provide a solver for SysML.

Furthermore, Figure 2 indicates three types of activities (i.e., requirements, structure, and constraints modeling) essential in any system design phase. Each of these activities results in a (set) of SysML diagrams. These diagrams provide a multidisciplinary model split into three fundamental levels: (1) the system's requirements with their desired numerical values and weighted priorities (e.g., total weight of 2 Kg with 70% priority), (2) the hierarchy of the components together with their respective parameters (i.e., components can be interdisciplinary, mechatronics, such as a motor with motor board controller or discipline-specific such as chassis as mechanical, electronic board as electrical or pure software code), and (3) the interrelationships between disciplines through the constraints with their corresponding input and output properties (e.g., power consumption, operational time, total price).

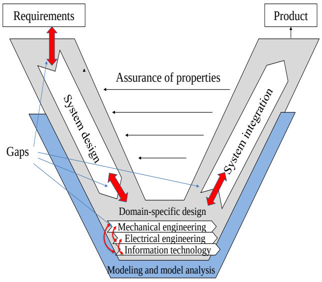


Figure 1: The V-Model as a macro-cycle describes the generic procedure for designing mechatronic systems [3].

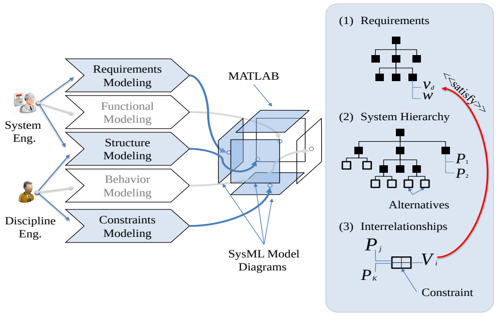


Figure 2: The SysDICE framework: Five modeling activities result in a set of SysML diagrams (The functional and behavior modeling are future work).

In this manner a system model holds all the necessary interdisciplinary relations, constraint information, different component alternatives as well as requirements values and priorities in one unified model. This unified SysML model is then converted into MATLAB for evaluating different configurations of requirements' values and priorities. This evaluation is conducted with the goal of attaining the best component alternative to suit the customers' objectives. For that, a theoretical system design model is presented in Section 5.2 and a mathematical optimization problem is formulated and solved as described in Section 5.3.

## 5.2 System Design Model

During early design stages a set of requirements spanned over the various domains is provided. In our framework, each of these requirements is modeled using the ≪*requirement*≫ block within the req diagram. To be fully able to specify a numerical design requirement, we extend the existing SysML requirement block by stereotyping it to include its value, $v_d$ and its corresponding priority, $w$. We further consider the hierarchy of the requirements using the *containment* relationship for the traceability.

In industry, after the design requirements have been settled, system engineers commence to analyze the type of system satisfying such requirements. At this stage, the system evolves from a black box to detailed subsystems reaching the component levels. Following a similar trend, our framework then decomposes the system into its constituent subsystems and their corresponding components. This is achieved through the SysML ≪*block*≫ element and the *composition* association within the bdd diagram. Each component of the system has various alternatives which are modeled with a stereotyped ≪*block*≫ in order to represent their uniqueness in a possible conceptual design solution. They are specified by their corresponding properties such as the weight, the price, the power consumption and so forth. The relations between these properties are modeled using the ≪*constraintProperty*≫ within the par diagram.

Additionally, the system design model is generated in an iterative and evolutionary manner with each of the three activities. At the stage where the model is fully specified from the requirements down to the properties level, the goal then is to find the optimal alternative combination that best suits the prioritized, and possibly conflicting requirements. Therefore, the stereotyped requirements with corresponding values (i.e., $\mathbf{v}_d$ and $\mathbf{w}$) as well as all other blocks with their respective properties are transformed to MATLAB. The constraint properties with their MATLAB-based equations are transformed into MATLAB functions. In the next section we provide the mathematical formalization of the weighted requirement satisfaction problem.

## 5.3 Mathematical Formulation

Given a set of $k$ requirements, we define $\mathbf{v}_d = [v_d^{(1)}, \ldots, v_d^{(k)}]^T \in \mathbb{R}^{k\times 1}$ to represent the different desired values of each of the requirements, and $\mathbf{W}_{k,k} = diag(\mathbf{w})$ to be the diagonal matrix representing the priorities of each of these requirements. We further define $\mathbf{v} = [v_1, \ldots, v_k]$, to represent the output of the constraint equations which relate a set of priorities as its inputs.

We assume that these values are noisy[1], with a gaussian noise, and that the requirements are weighted in each of the $k$ directions according to their priorities. Therefore, the likelihood for a desired value to occur is

[1]We assume that the combination and or values of the properties are not exact and rather noisy.

defined by,

$$p(v_d^{(i)}|v^{(i)};\sigma^2, w^{(i)}) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} w_{i,i}(v_d^{(i)} - v^{(i)})^2\right) \tag{2}$$

Maximizing the natural logarithm of Equation 2, leads to the following minimization problem,

$$\min_{\mathbf{v}} \frac{1}{2}[\mathbf{v} - \mathbf{v}_d]^T \mathbf{W}[\mathbf{v} - \mathbf{v}_d] \tag{3}$$

Equation 3 represents the weighted requirement satisfaction problem. In other words, the solution of the minimization problem is seeking the optimal value, $\mathbf{v}^*$, so as to minimize the error with respect to the desired combination weighted by the priorities (i.e., $\mathbf{v}^* = \arg\min_{\mathbf{v}} \frac{1}{2}[\mathbf{v} - \mathbf{v}_d]^T \mathbf{W}[\mathbf{v} - \mathbf{v}_d]$).

To approximate the values of the corresponding combination of the properties, we resort to GPs. The reasons for our choice is threefold: (1) the constraint equations are complex and thus require non-parametric functional approximators, (2) the lack of available training data which imposes good generalization properties of the used approximators, and (3) the need for a problem independent framework.

The approximated functions are then substituted in Equation 3, to generate a new minimization problem defined by the following cost function,

$$\min_{P} J(P) = \frac{1}{2} \sum_{i=1}^{k} w_{i,i}(\mathcal{GP}_i(P) - v_d^{(k)})^2, \tag{4}$$

where $P = p_1 \bigotimes p_2 \cdots \bigotimes p_N$, with $N$ being the number of components, and $J$ representing the cost function.

To minimize Equation 4, we need to compute the derivatives with respect to the input. Here we approximate the derivate of a GP using first order approximation and then use conjugate gradient descent for the optimization. The output is $P^*$ that satisfies the set combination of the prioritized requirements (i.e., $\arg\min_P \sum_{i=1}^{k} w_{i,i}(\mathcal{GP}_i(P) - v_d^{(k)})^2$).

# 6 Experiments and Results

In this section we explain two different design experiments that were conducted.

## 6.1 Experiment One: Two Wheel Differential Drive Robot

The first experiment illustrates the design of a two wheel differential drive robot. The *e-puck*[2], top-right of Figure 3, is an example of such robots. Next we describe the application of our proposed framework in: (1) modeling the robot using SysML and (2) using the mathematical formulation and GPs to find the optimal combination of component alternatives to satisfy different requirements' configurations.

### 6.1.1 SysML Model Generation

During the initial stages of the robot's system design phase, system engineers identified robot's requirements as well as the possible conceptual solutions and discipline engineers detailed the solution concepts with their domain-specific information and the possible alternatives. Conclusively, a system model of the robot with SysML was achieved based on these information. SysML modeling was done using the open source tool TOPCASED-SysML [16].

Figure 3 shows the three types of SysML diagrams: req, bdd, par diagrams used to model the required information of the mobile robot. The top-left of Figure 3 shows a part of the main design requirements: the *TotalWeight*, the *TotalPrice*, the *MaximumTranslationalVelocity*, and the *OperationTime*. Each is stereotyped as *"REQ"* to allow for the addition of the requirements' properties (i.e., $v_d$ and $w$). Similarly all other requirements were modeled. Each *REQ* must be satisfied by a value of a design entity (i.e. component, property or even a system). Therefore, the ≪*satisfy*≫ association was used to represent which design entity satisfy which requirement.

The robot components are modeled using bdds. We model the components hierarchy, using the SysML ≪*composition*≫ relationship. Figure 3 details modeling these components. Each component of the system is described using its own block that holds certain properties typically needed by the engineer during the design phase. In our example the robot consisted of 7 different components, each having its own alternatives. These alternatives are modeled with blocks that are stereotyped as *"ALT"* so to indicate the multi-alternatives for each component during the transformation (e.g., Motor1Type1, Motor1Type2).

[2]e-puck: http://www.e-puck.org/

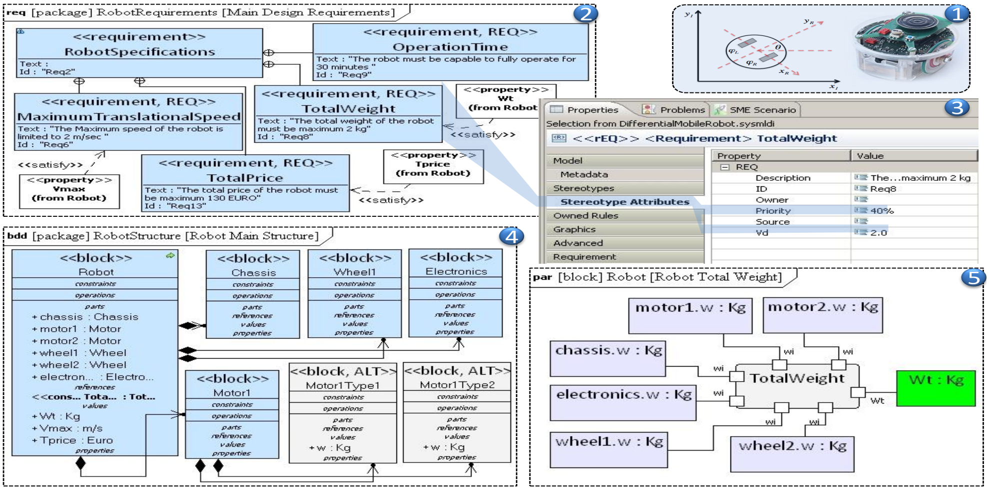


Figure 3: Robot SysML diagrams: (1) e.g. the *e-puck*, (2) requirements with (3) *TotalWeight* properties, (4) bdd for components' structure and alternatives and (5) par for the *TotalWeight* constraint property.

Various *par* diagrams were used to model the mathematical equations between the component properties. Each equation is represented with a ≪*constraintProperty*≫ with its own input and output properties. For instance, the constraint *"TotalWeight"* is used in the *par*, Figure 3, to relate all the components' weight properties (component.w) thus indicating the value of the actual total weight of the robot $W_t$. Here the *TotalWeight REQ* is satisfied by this property $W_t$ that indicates the actual value $v$. The kinematical, dynamical as well as other related equations, such as the total power consumption, the total price, and the operational time have been also modeled similarly with other par diagrams. At this stage a SysML model incorporating all the disciplines is generated after several iterations. Therefore, the necessary information for system engineers is ready and the communication burden is solved.

**Conclusion I:** SysML can serve in bridging the communication problem.

### 6.1.2 SysML Model Evaluation

To better evaluate the framework, we have conducted various experiments with different priorities and desired values of the requirements. The system was provided with different alternatives having different properties, such as, the mass, the price and so forth as described above. The algorithm was provided with different $\mathbf{v}_d$'s and $\mathbf{w}$'s. After the GPs were approximated, conjugate gradient descent was applied to find the optimal alternative suiting the requirements. The values corresponding to the properties of the determined alternative could be seen in the appendix[3]. Figure 4 shows the results of providing different values and priorities. The three axis of the graph represent the components, properties and the alternatives respectively. The different planes are the optimal alternatives resulting from different requirements's values and priorities. Each of these priorities and/or properties change represents a different design focus. For instance, in the first plane (1st alternative) the focus was more towards having a high velocity robot (i.e., 2 m/s) with high operational time (i.e., 1 hour), where both requirements were given a priority of 90%. The second plane (4th alternative) represents a moderate robot while the third

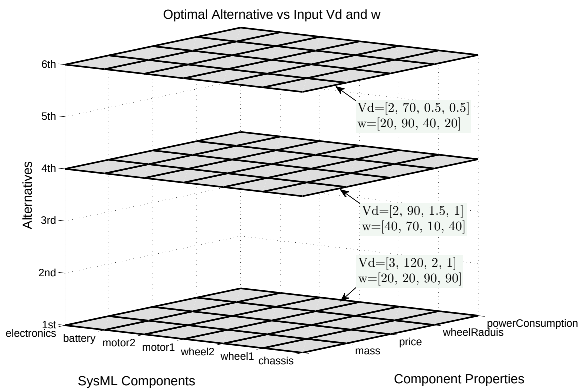


Figure 4: Results on three different design experiments. Each plane represents the optimal alternative for the corresponding requirement priorities and values.

[3]Appendix published online at: https://dl.dropbox.com/u/2689877/bnaic2012IntelligentDesignAppendix.pdf

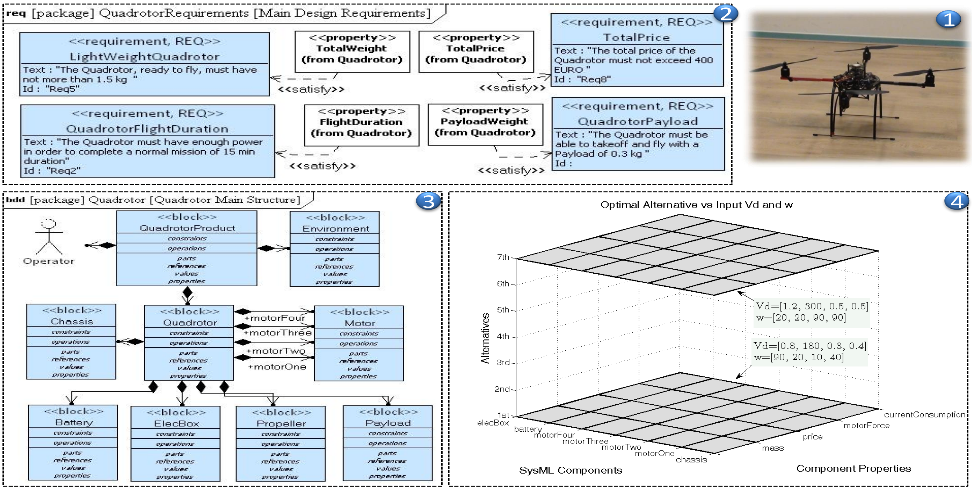


Figure 5: (1) the quad-rotor system present at the Swarm-Lab in the Department of Knowledge Engineering, Maastricht, The Netherlands, (2) a req diagram showing a part of the main design requirements satisfied by their respective properties, (3) a bdd presenting the quad-rotor components structure and (4) results of two different design focus requirements.

($6^{\text{th}}$ alternative) correspond towards having a cheap price robot of 70 € with a high priority (i.e., 90%). It becomes obvious from Figure 4 that the platform was capable of capturing different optimal alternatives suiting different design focuses and requirements and thus being adaptable and generalizable.

**Conclusion II:** The proposed framework is capable of attaining the optimal combination to suit a set of prioritized requirements.

**Conclusion III:** The proposed framework is capable of attaining different optimal alternative solutions to different design focuses and thus being adaptable.

## 6.2 Experiment Two: Quad-rotor

To better asses the design and the evaluation process, we have conducted a second more complex design task. In this experiment a quad-rotor unmanned aerial vehicle, shown at the top-right of Figure 5 was designed. The quad-rotor is a system consisting of four rotors in a square connection. The dynamics of the system are represented by a 12-dimensional state-vector and the actions are different torques delivered by the motors. In this task we had more constraints taken into account as well as more alternatives.

### 6.2.1 SysML Model Generation

The SysML model generation phase was generated similarly to the one described in the previous experiment. Here the system had to satisfy four requirements. Namely, *LightWeightQuadrotor*, *TotalPrice*, *QuadrotorFlightDuration* and *QuadrotorPayload*, shown in Figure 5. Further, the components as well as the constraints were modeled using the corresponding diagrams similarly to the last experiment.

### 6.2.2 SysML Model Evaluation

We have also conducted various experiments with different priorities and desired values of the requirements. The system was provided with different alternatives having different properties. The algorithm was provided with different $\mathbf{v}_d$'s and $\mathbf{w}$'s. After the GPs were approximated, conjugate gradient descent was applied to find the optimal alternative suiting the requirements as described in Section 5.3. The bottom-right side of Figure 5 shows the results of providing different values and priorities for the requirements in two experiments (the properties' values corresponding to all possible alternatives could be seen in the appendix[3]).

Here also, the three axis of the graph represent the components, properties and the alternatives respectively. The different planes are the optimal alternatives resulting from different requirements's values and priorities. Each of these priorities and/or properties change represent a different design focus. It is clear from the values that the first plane ($1^{\text{st}}$ alternative) correspond to a low-weight quad-rotor (0.8 kg) with priority of 90% while the second plane ($7^{\text{th}}$ alternative) is a result of a quad-rotor with high flight duration (0.5 hour) and high playload (0.5 Kg) with both a 90% priority. Similar conclusions could be drawn from this experiment, where the proposed framework bridges the communication gap, can attain optimal alternative combination and is adaptable.

# 7 Conclusions and Future Work

In this paper we presented a SysML-based approach in order to support the design of mechatronic systems. By leveraging SysML, the platform was capable of incorporating the interdisciplinary interrelations that go with and complicate the design process. The framework was split into three fundamental levels that are typically used in the design process. It further makes use of Gaussian Processes in order to find a functional mapping at the system-design level. These are then used to solve for the best alternative that optimally suits a set of requirements. Experiments conducted on the design of two systems, show the accessibility and adaptability of the approach, whereby the framework was capable of bridging the system engineering level communication problems, attaining optimal alternatives to a set of requirements, and producing adaptable solutions to various design focuses.

In future work, we aim to extend the actual system model with the interfaces across components in order to restrict the space of alternative exploration to suit the requirements. On a higher level, other discipline-specific information, the functional and the behavior aspects, will be incorporated in the existing system model. Moreover, we are in a sequel of using transfer learning to adapt already learned behaviors in similar designs of similar systems.

# References


[1] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. No. ISBN: 978-0123743794, Elsevier, Morgan Kaufmann OMG Press, 2008.

[2] G. Pahl, W. Beitz, J. Fledhusen, and K.-H. Grote, *Engineering Design A Systematic Approach*. Springer, third edition ed., 2007.

[3] *VDI 2206 Design methodology for mechatronic systems*. Beuth Verlag GmbH, June 2004.

[4] F. P. Stappers, L. J. Somers, and M. A. Reniers, “Multidisciplinary Modeling - Current status and expectations in the Dutch TWINS consortium,” in *ICSSEA*, 2008.

[5] A. A. A. Cabrera, M. S. Erden, M. J. Foeken, and T. Tomiyama, “High Level Model Integration for Design of Mechatronic Systems,” IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, October 2008.

[6] K. Chen, J. Bankston, J. H. Panchal, and D. Schaefer, *A Framework for Integrated Design of Mechatronic Systems*, ch. 2, pp. 37–70. Springer, 2009.

[7] “Object Management Group (OMG) Systems Modeling Language (OMG SysML™), available at http://www.omgsysml.org .,” November 2008.

[8] K. Thramboulidis, “The 3+1 SysML View-Model in Model Integrated Mechatronics,” *Journal of Software Engineering and Applications (JSEA)*, vol. 3, no. 2, pp. 109–118, 2010.

[9] A. Qamar, J. Wikander, and C. During, “Designing Mechatronic Systems: A Model-Integration Approach,” in *Proceedings of the 18th International Conference on Engineering Design (ICED11)*, vol. 4, pp. 145–156, 2011.

[10] A. A. Shah, D. Schaefer, and C. J. Paredis, “Enabling Multi-View Modeling With SysML Profiles and Model Transformations,” in *International Conference on Product Lifecycle Management*, p. 10, Inderscience Enterprises Ltd, 2009.

[11] M. Chami, H. Seemller, and H. Voos, “A SysML-based Integration Framework for the Engineering of Mechatronic Systems,” IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, IEEE, 2010.

[12] R. Stetter, H. Seemller, M. Chami, and H. Voos, “Interdisciplinary System Model for Agent-Supported Mechatronic Design,” in *Proceedings of the 18th International Conference on Engineering Design (ICED11), Vol. 4*, 2011.

[13] O. Mouelhi, P. Couturier, and T. Redarce, “An Artificial Intelligence Approach for the Multicriteria Optimization in Mechatronic Products Design,” in *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation*, pp. 1731–1736, 2009.

[14] X. Xu, L. Fu, and S. Fang, “Research on Product Variant Design with Uncertainty Information,” in *Proceedings of the 7th World Congress on Intelligent Control and Automation*, (Chongqing, China), June 25 - 27 2008.

[15] C. E. Rasmussen and C. K. I. Williams, “Gaussian processes for machine learning,” MIT Press, 2006.

[16] “TOPCASED-SysML Modeling Framework Open Source Project, http://www.topcased.org ..”