



HAL
open science

Tackling optimization and system driven engineering in coupling physical constraints with MBSE: the case of a Mobile autonomous line of products

Lorraine Brisacier-Porchon, Omar Hammami

► To cite this version:

Lorraine Brisacier-Porchon, Omar Hammami. Tackling optimization and system driven engineering in coupling physical constraints with MBSE: the case of a Mobile autonomous line of products. CSER 2023 - Conference on SYstems Engineering Research, Mar 2023, Hoboken (New Jersey), United States. ⟨hal-04041367⟩

HAL Id: hal-04041367

<https://hal.science/hal-04041367v1>

Submitted on 22 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

2023 Conference on Systems Engineering Research

Tackling optimization and system driven engineering in coupling physical constraints with MBSE: the case of a Mobile autonomous line of products

Lorraine BRISACIER-PORCHON*, Omar HAMMAMI

IP Paris, ENSTA Paris 826, Boulevard des Maréchaux, 91120 Palaiseau France

^bSecond affiliation, Address, City and Postcode, Country

Abstract

The main goal of product line engineering is to build complex system architecture at the best quality, cost, resource $\langle Q,R,T \rangle$ ratio. The return on investment in terms of $\langle Q,R,T \rangle$ assessment is however not trivial, as systems to build rise in complexity. Moreover, the perspective of system of systems engineering that set up both historic and new systems in capabilities, and the introduction of more and more autonomous systems in architectures makes the anticipation of return on investment impossible to achieve without computer assistance. The necessary tools to assess $\langle Q,R,T \rangle$ as precisely as possible also involve a wide exploration of possibilities in an ever-changing context. The absence of mechanical or multi physical aspects in SysML-based tools, either in its version 1.3 or 2.0, makes it inefficient in representing or simulating robotic systems or system of system engineering. This article explains the benefits of tackling a classic Multi-Objective Knapsack Problem (MOKP) to the UGV product line items selection using a seamless system architecting toolchain. The association of MBSE (Model-based System Engineering), OR (Operation Research) and MBD (Model Based design) that generated various designs is presented. Our results in system engineering in UGV presents a Pareto Front of trade-offs that can count as numerous possibilities that sole MBSE or separate MBD simulation could not have represented as the best in the sense of $\langle Q,R,T \rangle$. The simultaneous variations in both hardware and mechanical design show was entirely automated using standard tools with no redesign. This shows that seamless automations should pave the future of system engineering tools. With Operation research and Systems engineering tackling methods, our model upscale to real systems will shape System-Driven Engineering that will require new skills in system validation and verification and simulation analytics.

© 2023 The Authors.

Keywords: System Driven Engineering, Operation research ;MBSE, CAD, MCMDKP

1. Introduction

Product line context of system of system architecture design requires more and more complex tools and methods. The goals complexity, expressed as capabilities, are getting more complex [1] Defence systems are no

* Corresponding author. Tel.: +33 69 885 9365; fax: +0-000-000-0000 .

E-mail address: Lorraine.brisacier@ensta-paris.fr

exception to this trend. The number of disciplines and their interactions as architecture trade-offs that are required to build an unmanned ground vehicle are a sample of complexity itself. If the systems act together as system of systems and this capability is the quality objective of the systems on the ground, a costs and development time are constraints to achieve the capabilities [2]. This article proposes a method to select a product line of robots that would fit system of system defence scenario.

Selecting a toolchain to build an architecture is indeed an optimization problem in itself, as the precision and the efficiency of the method and tool can never be proven with respect to the system category involved. Worse, the state of the art in SysML v1.3 [3] and the perspectives offered by SysML v2.0 [4] only offers a bridge to mechanical constraints representations via parametric representations. As MBSE is mainly based on SysML v1.3 in literature [3], the MBSE approaches are not equipped with sufficient material to cover Systems Engineering process. Most requirements will be in that sense left aside from start to be dealt with later in “prototyping” validation. Yet those requirements, concerning for instance mechanical, electrical, dynamic automation or mandatory re-use of old parts or systems leads to costs and delays as much as other decisions regarding software, that could be covered by MBSE. Last, the toolchains in question lack of precision[5] because of redundancy, ambiguities or by human interoperability.

While most common system engineering publications rely on MBSE and SysML 1.3 principles to provide pragmatic models [6] and to share high-level information, the inclusion of physical constraints are seldom expressed in system models. Furthermore, the processes exposed for MBPLE using variants at high level with Cameo Modeler [7] or ARCADIA/Capella [8] do not include automated connections that would take mechanical constraints impacts on system overall decisions.

As a result, the consequences of design decisions taken on mechanical aspects of a system can seldom be included in decisions regarding software. The configuration management tools for MBSE and MBD are also often disconnected. System design and MBSE methods and tools have proven to be rather unprecise too [5], which means that the simulation of MBSE models has to be adapted to the systems by the system engineers rather than tool vendors. Hence the effort to use a tool in a company requires to maintain specific skills in order to keep model simulation active. Those skills, specific to both system knowledge and tool knowledge, are by definition rare and take time to train [9]. Furthermore, the training given by advisors from tool development company are seldom adapted to the needs of engineers. The systems engineering skills toolset [10] can be drawn as requirements and needs of system engineers. The costs and delays in product line engineering are influenced by those factors.

In this article, we claim to sketch a seamless engineering process that creates a set-based vision to take the best architecting decisions in constrained time and resource using a mature mathematical field: optimization [11].

On the other hand, operations research offers means to use objectives and constraints in mathematical equations. It may offer what is lacking in automation to build a toolchain with computer assisted connections between multiple disciplines required to build systems of systems. The Product line core concepts influence the costs and delays of variant expression, those choices have to be the most valuable compromises between objectives while respecting the constraints. This article shows a method to operate the parallel between product line item selection and optimization problems.

Some publications have started to tackle optimization problems to system of systems architecting such as [12] or [13] and make-or-buy trade study or safety assessment [14] and global system engineering models in aeronautics [15]. The growing interest for strong mathematical foundations for system engineering follows the trend of INCOSE vision 2035 [16]. Following that trend, the present article claims the construction of a Knapsack problem [17] model that suits the complexity of defining the best <QRT> architectures for small robotic system of system deployment in a defensive context. We have selected a genetic algorithm [18] to create a Pareto Front of variants that will be the set-based vision of our product line.

2. State of the art

a. System engineering product line definitions

The ISO 15288 in its 2015 version 2015 [19] mentions product lines as architecture and design activities result. The product line contains a core architecture and variations. The objectives of a company are to ensure a minimal cost and delay strategy to achieve the best product as defined in capabilities. Therefore, organizing core architectures and low-cost variations between products is an option to rationalize the investments. In a theoretical world, harmonizing production and conception means in a holy grail for analytic thinkers. But the outcome and output of industrial investment does not hold long in an ever-changing context. The product itself has an influence on all contributing systems, and the methods to reach the product line core concepts cannot be assessed in an acceptable timeframe without computer assistance, as shown in [20]. The limits and constraints in real life are imposed by contractual

engagement, the inheritance of past projects with reuse of historical investments, which makes a set-based vision at first necessary to make the most resilient decisions.

According to [7], the ARCADIA/Capella association offers variability and product line features associated with pure variant tools and a specific knowledge of systems to be built, with a 80% knowledge of core concepts. However, the lack of 3D representations in the tool makes it useless in considering the mechanical constraints on architecture. The absence of scheduling aspects of system engineering is also a problem. The return on investment (ROI) of MBSE on software tools [21] computation puts in perspective the major flaw of systems engineering representations : its efficiency. If the analogy with optimization is possible, then taking decisions over a short-term period does not necessarily serve the system scale objectives. The constraints brought by other domains are never represented in this ROI, and therefore cannot resist to major context changes such as the depletion of resources.

b. Architecture Frameworks: a necessary and sufficient set of views

To the best of our knowledge, seamless engineering flows tailored to a specific line of product is not a common publication trend. The seamless flow seems only to be interesting if the development of method is worth more than what is invested in automations [22]. The best quality automations are not always the best choices regarding costs and delays in engineering. The quality metrics for a product line are founded on income generated [23]. The down scaling from numerous vehicles to a small number of vehicles is not trivial, as the investment in process becomes less efficient, because the scale of consequences has to be shortened.

The representations to guide product lines choices could be framed in architecture frameworks, such as NATO Architecture Framework (NAF) [24], but it lacks of validation principles [2] which means that there will be difficulties to design the model itself and to introduce automated variations in it. Furthermore, the system to be built does not necessarily require all representations. For example, Figure 2 presents a subset of views from the NAF 3.1 constellation in Figure 1 that was necessary and sufficient to building [2]. But to the best of our knowledge, the necessary set of views to design is not standardized with respect to the category of product. Without a database of matching subset of views, the engineers are left to make choices in representations that will take their time without knowing whether the set of views is going to be useful to the project.

To the best of our knowledge, physical constraints are not considered in sysML v1.3, SysML v2.0 or NAF 3.1 or NAF 4.0. The parametric studies are not automated with 3D models. Therefore, no strategic decisions on core concepts can be based on those system models, because only a part of the overall compromises are brought to light.

Figure 1 : Constellation of interconnected NAF views

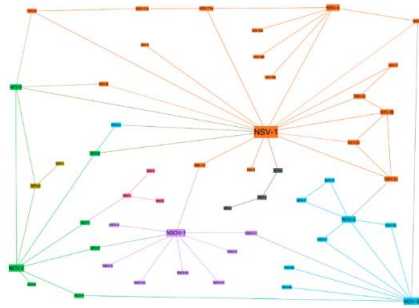
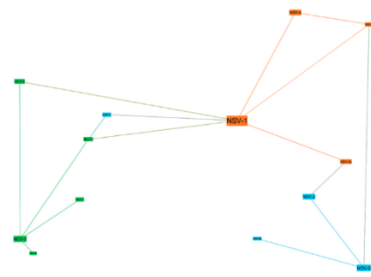


Figure 2: Constellation of selected interconnected NAF views



In this article, we will present only MBSE standard views that are included in Magic grid, because the subset presented in Figure 2 is included in Magic Grid. The extension to more pragmatic models using NAF views is ensured if the connection to the tool is set. The choice of using Magic Grid [25] is justified because of the costs of licences: if the subset of views necessary to solve a problem is present in a smaller cheaper framework, it should be used.

c. Operation Research: problems of Knapsack and solver tools

[23] and [24] exposes the current state of the art in operation research methods and tools. From those, there is an overview in particular of the knapsack problem, which illustrates a mathematical problem that anchors in real life situations [26] [27]. The Knapsack at glance in this article is the Multi-dimensional Multi-objective Knapsack depicted in equations (1)-(4), which is, again from [27], one of the hottest topic in operations research. To solve those proven strongly NP-Hard problems [28], an exact solution can be found if and only if all solutions are found. But in large

numbered instances, the exhaustive number of solutions would take thousands of years to be computed, which makes approximative solutions more efficient in a limited time. In the field of algorithms propositions, detailed in [29], a collection of non-dominated solutions with respect to objectives is defined as a Pareto Front. If our goal is to produce a set-based view of products, we will only highlight those in this article.

Before introducing the mathematical model candidate to system architecture, here is a list of variables used and their pragmatic definition.

- p_{ij} are the profits expected from the resolution of individual problems. The profits are expressed in terms of $\langle Q, R, T \rangle$. p_{ij} is a vector as we are dealing with true multi-objective problem.
- There are $j \in 1, \dots, n$ items types in each knapsack, i.e. in each product line variant
- There are $i \in 1, \dots, m$ constraints to be applied to each knapsack type, all expressed in terms of $\langle Q, R, T \rangle$
- c_i is a constraint
- w_j is the weight of an item x_{ij} ,
- $l \geq 2$ is the number of objectives to solve. The objectives are prioritized from 1 to t and will be resolved in a Pareto Optimality
- x_{ij} are the instances of item type j, 0 if the instance is activated and 1 if not.
- X_j are the list of item types in the concept of a knapsack.

$$\max (\sum_{j=1}^n p_{j1} X_j, \dots, \sum_{j=1}^n p_{jl} X_j) \quad (1)$$

$$s. t. \sum_{j=1}^n w_j x_{ij} \leq c_i \quad (i = 1, \dots, m) \quad (2)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad (j = 1, \dots, n) \quad (3)$$

$$x_{ij} \in \{0,1\} \text{ and } X_{\{j\}} \in \{0,1\} \quad (i = 1, \dots, m, j = 1, \dots, n), \#(4)$$

The Knapsack problem was first introduced in 1978, and numerous solutions are available to use in systems engineering. In this article, we use the NSGA-II algorithm [30] to solve our instance of the problem. The selection criteria of an algorithm from the state of the art [29] was the resolution time on large instances of problems.

If the number of constraints i is higher than the number of variables j , then the system problem will be considered hyperstatic, which could be strong grounding for constraints re negotiation in a system context. The equations can be questioned also in a changing context, with the priorities of the objectives, and modifications of constraint expressions.

d. System Category: small sized robotic defensive system of systems

The inclusion of robotic systems in battlefields in the future will imply to model the robots. Our works have generated some objectives related to the capabilities identified for the robots, and some constraints from the software, mechanical, electrical power. The mathematical foundations from equations (1) to (4) ensure that the addition of objectives or constraints from our example will only be adding computation time, with little costs. As long as the

As state of the art about small size robots is consequent, we managed to select one that appears in many publications to put to test in a "mule" mission: the robot will have the mission to follow humans, at walking speed, around the campus for 3h. Some examples of our testing scenario have been deployed in [31].

The system category of UGV will be detailed below : some of its multiphysical aspects are modeled using equations (5) to (15). The Goal of our model is to enhance the $\langle Q, R, T \rangle$ of a future robot of the same category. For synthetic redaction purpose, we focused on introducing variations in the wheel design, and built a model to link it to its system scale performance, including the impacts on the MBSE model.



Figure 3: View of the UGV system at glance

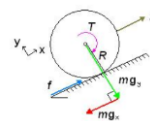


Figure 4: Wheel torque decomposition

From previous experiments with the robot, we had strong incentive to adapt the robot to the expected capabilities that inspired the V&V in [32]. The conclusions led us to the idea of adapting the wheels to fit better a surveillance mission in the campus. The worst-case scenario for our available engine would be an acceleration up a maximum slope θ of 30° .

Equations (5) and (6) express the force vectors decomposition illustrated in Figure 4:

$$mg_x = mg \times \sin(\theta) \quad (5)$$

$$mg_y = mg \times \cos(\theta) \quad (6)$$

The friction between the wheel and surface can be modelled as f . The torque T is defined in equation related to f (7):

$$T = f \times R \quad (7)$$

Figure 5 illustrates the Torque and friction f in the worst-case scenario, where the acceleration a is not null. Equation (8) expresses the acceleration in the x axis, balancing forces on x axis.

$$\sum F_x = M \times a = f - m \times g_x \quad (8)$$

$$M \times a = T/R - M \times g \times \sin(\theta) \quad (9)$$

$$T = R \times M \times (a + g \times \sin(\theta)) \quad (10)$$

$$T = \frac{(a + g \times \sin(\theta)) \times M \times R}{N} \quad (11)$$

The Torque value in (10) is the expression we modelled for one wheel. In (11), we add N as the number of wheels, which divides the torque required for the system:

$$P = T \times \omega \quad (12)$$

The next equations (12) to (15) add the electrical power that will define the category of the system. The equations below will help sizing the capacity c of the battery:

$$P = I \times V \quad (13)$$

Where ω is the angular velocity specified by the builder of the robot and P is the power required from the engine.

$$I = \frac{T \times \omega}{V} \quad (14)$$

V is the supply voltage of the system, and I is the maximal admissible current from vendor specifications.

$$c = I \times t \quad (15)$$

This capacity is obtained per motor. The example Husky has two motors, so this required capacity has to be doubled.

e. Quality metrics on our design flow for engineering

An architecture can be evaluated in terms of appropriate $\langle Q,R,T \rangle$ assessment, with Q referring to the rate of capability realization up to its limits, R to the resources found to make the best Quality in terms of costs, and T the time scheduled to realize the concepts as defined in [19]. The evaluation of a method and tool to realize an architecture and evaluate its performances has to be evaluated as well. Our previous works lead us to build [5] a framework to evaluate methods and tools. We use it to criticize the TUMA flow. Five criteria have been selected to make an evaluation, that can be applied to evaluate any advantages and drawbacks of a method and tool in relation with a system.

1. precision, which is related to the correlation between method and model. The more automated and simulated the workflow, the more precise the flow gets.
2. ease of use, which is related to the adoption of the toolchain. The presence of help, community or anything that will make the toolchain affordable and quick to apply.
3. efficiency, which takes the coverage of the method and tool compared to what is needed to achieve the capability of engineering the system, related to the notions of system-driven engineering (SDE).
4. interoperability, which considers the continuity of the engineering flow with other tools and methods.
5. re-usability, which considers if the tool and method can be reused, or if the artefacts left by the method and tool can be reused in other programs at minimal costs.

All criteria are guides to place the toolchain in a more general mapping and will serve the enterprise architecture model.

3. Experiment – proposition of the TUMA flow in engineering a line of product of the UGV category

In this article, we claim the initiative of tackling Knapsack problem to solve a robot product line architecture. The core features will be decided on what are the constraints and objectives of a mobile robot. The design for the robot should be issued from its category in terms of capabilities, modelled using objectives, and constraints. The design method will be to pick the most different options in a space of possibilities. The design space as defined in [33] is the space of possibilities offered in solving the model below in a constrained time.

The engineering flow around the variables, domains and constraints are interconnected. The views proposed are vehicles to the information that flows through the engineering process. The problem is decomposed in two automatically linked workflows, using ModelCenter tool in version 15.3[34], Catia v5[35] and Cameo Modeler v 19[36].

a. The TUMA flow overall principles

We introduce in this article a seamless engineering flow called TUMA that produces comparable architectures in terms of $\langle Q, R, T \rangle$ objectives. The first steps 1 to 4 of this flow are composed of electronic benchmarking and simulation: the contribution of software and hardware of systems are defined first in the architecture. Then a second simulation is operated from the architecture sketches selected in the first part of the flow: this happens through step 5. The simulations in step 5 happen using model-based design (MBD) models such as 3D model generation or physical evaluation. Note that both Step 4 and step 5 contain a knapsack model, so equations (1)-(4) are implemented in both simulations with different objectives.

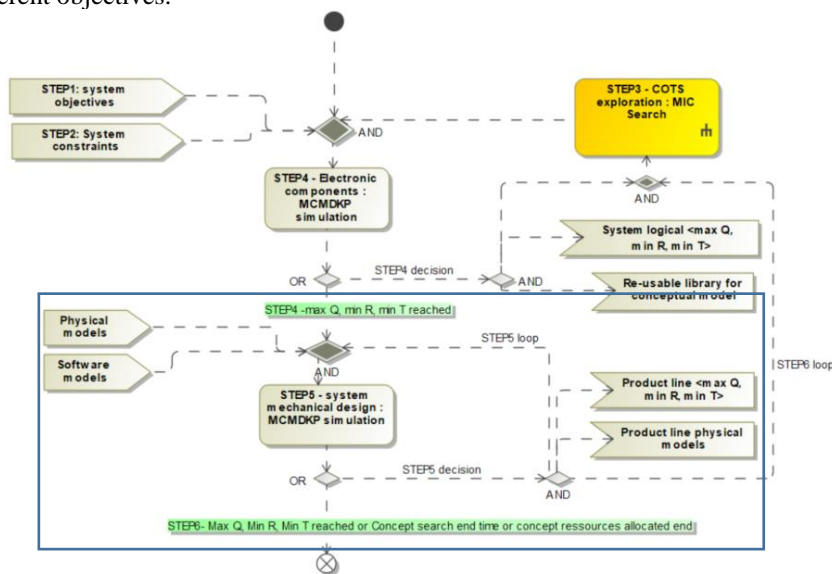


Figure 5: TUMA flow, an instance of SDE for UGV category engineering

In this article, we focus on step 5 presented in Figure 5. The data sets that were input in our method was previously selected from a Pareto-optimal architecture that were the best $\langle Q, R, T \rangle$ compromises on MBSE considerations. The steps 1 to 4 are related to the diversity of options in trade-off analytics that are exposed for example using Model-identity Card (MIC) based search [37], in the same spirit as [38]. In that way, all X_j from equation (1) are already set-up, and only variability in x_{ij} from equation (4) have to be explored in step 5.

b. Variables and their domain

The system product line configurations are variables. The common core of the product line will be a consequence of the identification of all variants of the product line.

The systems UGV main features and variations are inspired from the category and the context in which the product line shall be designed: some $\langle Q, R, T \rangle$ competitive systems that would replace the Husky on Figure 3, and that would together achieve different purposes. The robot category – an electrically powered architecture and size with the UGV engine- is not questioned in our works. The presented works can be extended to other features in another project context.

Hence the variables of the system are defined in:

1. X_1 : Batteries. Domain of x_{i1} : {Battery1,...,Battery10}
2. X_2 : Wheel radius. Domain of x_{i2} : [100,165] mm
3. X_3 : Wheel thickness. Domain of x_{i3} : [10,20] cm
4. X_4 : Tyre types. Domain of x_{i4} : {Type 1, type 2, type3, Type4}
5. X_5 : Chassis type. Domain of x_{i5} : {Light, heavy}
6. X_6 : Processor boards. Domain of x_{i6} : {Processor1,...,Processor10}
7. X_7 : Camera types. Domain of x_{i7} : {Camera1,...,Camera10}
8. X_8 : LIDAR types. Domain of x_{i8} : {LIDAR1,...,LIDAR10}
9. X_9 : GPS types. Domain of x_{i9} : {GPS1,...,GPS10}

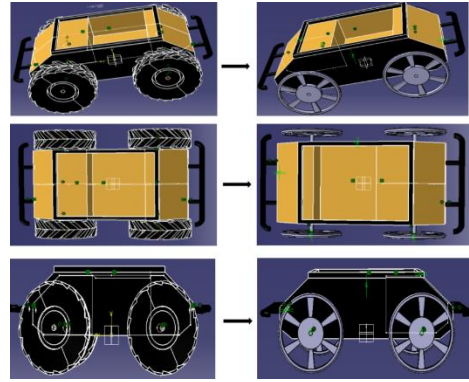


Figure 6: wheel models in thickness x_{i3} and radius x_{i2} for our UGV product line variations : from original (left) to thinnest wheels (right)

Figure 6 represents possible architectures that were generated by our toolchain. The views of model variations were generated and evaluated at the same time. In that way, there is no ambiguities between assessment in an architecture and its concepts model. The engineering flow presented here from MBSE to MBD is based on standard tools, and required no adaptation.

c. Constraints

The limits imposed by SysML usage are related to the lack of physical expressions. The inclusions of 3D views associated automatically with a concept will delete the ambiguities that reside in redundancy caused by manual input in either one model or the other. Decisions in one domain, in our case 3D or MBSE, may have an impact on each other. The proposed method will both reduce unseen ambiguities, because it proposes automated representations of decisions in correlated views, and reduce obvious ambiguities with a mathematically founded way to take decisions over them in the sense of constraints of all domains considered, and not only in one. The constraints of our systems take the form of mechanical equations (5)-(15), and the impact of variability in the mechanical design over MBSE model is exposed.

The considered constraints were related to some selected system requirements, adding of deleting constraints would only modify the results but the method will remain. The requirement considered were:

<p>«requirement» Payload</p> <p>Id = "REQ-081" Text = "The robot shall transport loads of weight equal to or less than 75 kg."</p>	<p>«requirement» RobotWeight</p> <p>Id = "REQ-027" Text = "The robot weight shall be equal to or less than 50 kg."</p>	<p>«requirement» Speed</p> <p>Id = "3" Text = "The robot speed shall be greater than 1 m/s."</p>
--	--	--

The environment in which those requirements is considered slope theta of maximum 30°, dry environment, and concrete soil. The overall performance will be validated in fine if the system reaches 3h of speed over 1m/s (walking speed) under those conditions. The conditions of testing and simulation are not discussed in this paper.

The UGV is complex, and as in a Knapsack problem definition, the problem of line of products will be considered as NP-Hard. The complexity of multidimensional representations and performance assessment promises to help in all fields (including safety, security, testability). The models created from the optimisation model will represent the core concepts and variabilities. The representations of the system in Figure 7 illustrates an architecture for UGV system category. In that way, constraints of MBSE related to the hardware and software architecture can be considered in optimization. The results will produce a set of architectures from which the best compromises can be highlighted to build prototypes for the product lines.

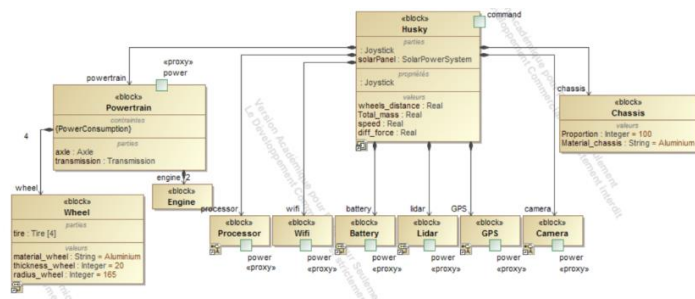


Figure 7: Husky Block Definition Diagram (BDD) in White Box / NSV-1 view

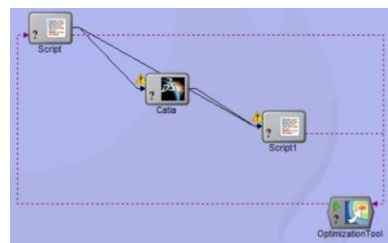


Figure 8: ModelCenter v15.3 model of UGV

The model can be divided in 4 nodes, in which the constraints and objectives are expressed.

Figure 8 expresses our engineering problem for UGV in a classical optimization flow in ModelCenter 15.3. In this model,

- the node Script is the expressions of domains and variables that are at stake, matching the domains of X_j .
- The Catia node has the constraints expressed in equations (5)-(15), in input the details x_{ij} that are required and outputs the evaluation of Husky diff_force, speed, total_mass and wheels_distance.

The node Script1 takes in input both CATIA outputs and more constraints illustrated in

- Figure 7 about the system electrical performance to reach the objective of 3h in the described conditions and is linked to the MBSE and requirement model. The requirements that are validated here are depicted in part 2.
- in node Optimization, there are $l = 3$ prioritized objectives in terms of speed ($k=1$), force ($k=2$) for payload ($k=3$) and mass in the scenario presented in section 2.

4. Results

The simulation scenario first started with wheel shapes evaluations, in order to reduce the weight of the system and to gain speed. The goal was to reach our capability of uninterrupted 3h of walking speed with usage of 5 to 6 electronical devices including a LIDAR that draws power while in motion, and to be able to do so in the worst-case environment depicted in section 2.

We used NSGA-II algorithm in order to have a rapid and satisfying convergence to an approximative result as reviewed in [18], because architecting time is itself time constrained. The presented engineering flow has the advantage to use tools off-the-shelf: all mathematical algorithms are already implemented in ModelCenter, and if the algorithm were not satisfying, the state of the art is available in the tool.

The input data of this step, all w_j of TUMA, was imported from system component datasheet exploration. The exploration of component datasheet can be criticized, as described in our previous works [31]. Still, the data imported was selected from available information about components, either declared or tested.

The results are checked back

The design space that was under test was small compared to the complexity for a product line of UGVs. We let the computer generate 48 configurations variants overnight (24h) with 1000 items x_{ij} population. The most time-consuming evaluation was the creation of a variation in Catia node (approximately 15s for each). Then the creation of the MBSE evaluation view (approximately 10s). If we add to that time the unquantifiable uncertainty of our test computer (5s) for each evaluation, all the possibilities could have taken $48000 \times (15 + 10 + 5)s = 1,44.10^6 seconds$, be 400 hours of computation, which we did not have.

These results are based on 2 function objectives: mass minimization and speed maximization in the case depicted in section 2b. The solutions found are illustrated in

Figure 9. We can identify the feasible, infeasible configurations and a Pareto Front of non-dominated solutions.

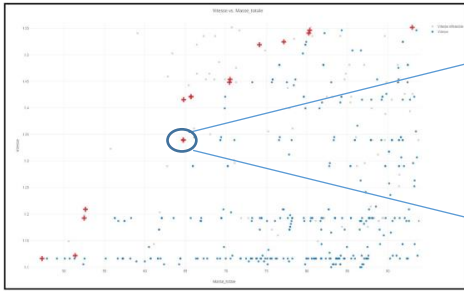


Figure 9: set based architectures for speed vs mass

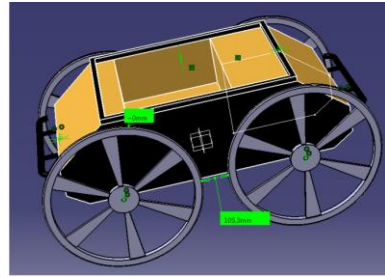


Figure 10: Design #8499 from Pareto Front

The final assessment of the system engineering will be pronounced with the verification of client requirements. Validation with requirements and margins. The product line generation will be selected from a set-based view of Pareto-optimal solutions, which is a success in our bi-objective evaluation. The proof of concept here is a stepping stone to dig deeper in this analysis, because it has created numerous possibilities for the product line decisions in less than 24h, while using CAD and MBSE without automation, two or three possibilities would have come forth without proof that they are non-dominated or consistent with each other.

5. Outlook

The TUMA flow is an instance of System-Driven Engineering principles illustrated in Figure 5. Our further works will be to explore the possibilities offered by System-driven engineering, of which the TUMA Flow is an instance. The tailored subset of views and the mathematical models for UGVs in this article can be questioned: we could add an articulation with scheduling problems and we could work on definition of the adequate set of views with respect to the system category, but we claim that using TUMA flow was precise and efficient in the sense of the COFRA depicted in section 2c by design.

The multiplication of instances of SDE principles will open a new set of engineering skills. The architecture and design of product lines can be in that way seen as a sequence of decisions on contradictory objectives. The best designs will therefore be the Pareto-optimal compromises. The representativity of simulation results using all required aspects of engineering decisions has to be clear. These results should be an incentive for tool vendors to force the native links and interoperability between MBSE, CAD and simulation. The lack of consistency between those tools make them hard to access, and reduces both their efficiency and their precision as a toolchain, in the case that manual input or output will most certainly cause ambiguities. The time and costs associated with tools ambiguities are always higher than those observed coming from simple textual ambiguities, because the problem of making multi physical architecting decisions is not solved.

In our case, the previous design has imposed constraints and requirements that led to hypothesis in constraints and objectives. The process in this article is valid for re-engineering of systems. The extension to creating entirely new architectures can be included in a learning loop designed in a return arrow in Figure 5 from STEP5 back to STEP3, but will require new skills. The addition of objectives and constraints will also add time to the simulation process. The combination of algorithm manipulation skills and analytics to read the results views in all architecture frameworks, systems engineering and MBD is still too rare to make our propositions popular. The association of systems engineering and operations research knowledge might be vital to ensure system engineering vision for 2035[16].

6. Conclusions

System architecture multi-disciplinary decisions should never be taken discipline by discipline, but rather as a line of product sequence of actions. The aspects of scheduling related to this engineering problems are not highlighted here, but their inclusion in a mathematical operation research problem does not challenge the analogy. The concept of tackling optimization model to System engineering for product lines architecting activities holds through our example, and should be tested on other products to achieve a database of appropriate mathematical models with respect to system category.

Future work in full agreement with one reviewer suggestion includes “optimize the optimization” by re-injecting real data from the developed and fielded system (eg: reliability data) and compare it to the MBSE model. There is a huge avenue of research and opportunities for the SE community. From a mathematical point of view, this paper opens directly accessible decades of results on the large portfolio of knapsack problem and its numerous variants. A forthcoming paper will tap in MCMDKP version of knapsack for large scale system of system optimization problems.

Acknowledgments

We would like to thank the reviewers for their valuable comments which helped improve this paper and allowed us to convey the right messages.

References

- [1] P. T. Grogan, "Perception of complexity in engineering design," *Syst Eng*, vol. 24, no. 4, pp. 221–233, Jul. 2021, doi: 10.1002/sys.21574.
- [2] L. Brisacier-Porchon and O. Hammami, "Coupling Architecture Framework and Operation Research: the Case of NAF with Cost and Delay Driven Multi-Criteria Scheduling," presented at the SoSE 2022, Rochester, NY, USA, Jun. 2022.
- [3] S. Wolny, A. Mazak, C. Carpella, V. Geist, and M. Wimmer, "Thirteen years of SysML: a systematic mapping study," *Softw Syst Model*, vol. 19, no. 1, pp. 111–169, Jan. 2020, doi: 10.1007/s10270-019-00735-y.
- [4] M. Bajaj, S. Friedenthal, and E. Seidewitz, "Systems Modeling Language (SysML v2) Support for Digital Engineering," *INSIGHT*, vol. 25, no. 1, pp. 19–24, Mar. 2022, doi: 10.1002/inst.12367.
- [5] L. Brisacier-Porchon, O. Hammami, and R. Boutemy, "Modeling A UAV in Practice: A Comparison between Rhapsody and Capella," presented at the IEEE ISSE, Vienna, Austria, Sep. 2021.
- [6] A. B. Rauzy and C. Haskins, "Foundations for model-based systems engineering and model-based safety assessment," *Syst Eng*, vol. 22, no. 2, pp. 146–155, Mar. 2019, doi: 10.1002/sys.21469.
- [7] M. Forlingieri and T. Weillkiens, "Two Variant Modeling Methods for MBPLE at Airbus," p. 17.
- [8] J. Navas, S. Bonnet, J. Voirin, and H. G. Chalé Gongora, "A value-driven, integrated approach to Model-Based Product Line Engineering," *INCOSE International Symposium*, vol. 31, no. 1, pp. 95–110, Jul. 2021, doi: 10.1002/j.2334-5837.2021.00828.x.
- [9] A. Harding, "Systems Skills ... From Here to Diversity," *INSIGHT*, vol. 25, no. 3, pp. 51–58, Sep. 2022, doi: 10.1002/inst.12399.
- [10] N. Hutchison and T. McDermott, "Is Systems Engineering Effectiveness the Heart of Today's Employability Skills?," *INSIGHT*, vol. 25, no. 3, pp. 11–16, Sep. 2022, doi: 10.1002/inst.12393.
- [11] E. M. Bednarczuk, J. Miroforidis, and P. Pyzel, "A multi-criteria approach to approximate solution of multiple-choice knapsack problem," *Comput Optim Appl*, vol. 70, no. 3, pp. 889–910, Jul. 2018, doi: 10.1007/s10589-018-9988-z.
- [12] D. Coolen and C. Dagli, "A System-of-Systems Meta-Architecture Optimization to Recommend a Solution for Personalized Home Fitness," p. 13, 2022.
- [13] J. Axelsson, J. Fröberg, and P. Eriksson, "Architecting systems-of-systems and their constituents: A case study applying Industry 4.0 in the construction domain," *Syst Eng*, vol. 22, no. 6, pp. 455–470, Nov. 2019, doi: 10.1002/sys.21516.
- [14] A. B. Rauzy and C. Haskins, "Foundations for model-based systems engineering and model-based safety assessment," *Syst Eng*, vol. 22, no. 2, pp. 146–155, Mar. 2019, doi: 10.1002/sys.21469.
- [15] J. Bussemaker and L. Boggero, "From System Architecting to System Design and Optimization: A Link Between MBSE and MDAO," presented at the 32nd annual INCOSE international Symposium, 2022, p. 17.
- [16] INCOSE, "Systems Engineering Vision 2035," *Systems Engineering Vision 2035*, 2022. <https://www.incose.org/about-systems-engineering/se-vision-2035>
- [17] A. Biglar, "Some applications of Knapsack problem," 2018, doi: 10.13140/RG.2.2.15115.39209.
- [18] J. G. Hobbie, A. H. Gandomi, and I. Rahimi, "A Comparison of Constraint Handling Techniques on NSGA-II," *Arch Computat Methods Eng*, Jan. 2021, doi: 10.1007/s11831-020-09525-y.
- [19] AFNOR, *ISO/CEI/IEEE 15288:2015 Systems and software engineering — System life cycle processes*, AFNOR. 2015. [Online]. Available: www.afnor.org
- [20] B. E. Ozkan and S. Bulkan, "Multi-objective approach to forecast design refresh time due to COTS obsolescence," *Systems Engineering*, p. sys.21613, Feb. 2022, doi: 10.1002/sys.21613.
- [21] J. B. Duffy, R. Combs, J. Feng, and J. P. Richardson, "Return on Investment in Model-Based Systems Engineering Software Tools," p. 15, 2021.
- [22] O. Hammami, "SYNSYS-ME: Seamless System Engineering to mechanical flow through multiobjective optimization and requirements analysis," in *2014 IEEE International Systems Conference Proceedings*, Ottawa, ON, Canada, Mar. 2014, pp. 411–416. doi: 10.1109/SysCon.2014.6819290.
- [23] *Systems Product Line Engineering Handbook*, Cépaduès. 2016.
- [24] NATO, "Nato Architecture Framework (NAF) - 3.1 - NATO All View," Oct. 2014. [https://training-course-material.com/training/Nato_Architecture_Framework_\(NAF\)_-_3.1_-_NATO_All_View](https://training-course-material.com/training/Nato_Architecture_Framework_(NAF)_-_3.1_-_NATO_All_View)
- [25] A. Aleksandraviciene and A. Morkevicius, *Magic Grid Book of Knowledge*, Vitae Litera. 2018.
- [26] V. Cacchiani, M. Iori, A. Locatelli, and S. Martello, "Knapsack problems — An overview of recent advances. Part I: Single knapsack problems," *Computers & Operations Research*, vol. 143, p. 105692, Jul. 2022, doi: 10.1016/j.cor.2021.105692.
- [27] V. Cacchiani, M. Iori, A. Locatelli, and S. Martello, "Knapsack problems — An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems II," *Computers & Operations Research*, vol. 143, p. 105693, Jul. 2022, doi: 10.1016/j.cor.2021.105693.
- [28] R. Luo, S. Ji, and B. Zhu, "A Pareto evolutionary algorithm based on incremental learning for a kind of multi-objective multidimensional knapsack problem," *Computers & Industrial Engineering*, vol. 135, pp. 537–559, Sep. 2019, doi: 10.1016/j.cie.2019.06.027.
- [29] M. Abdel-Basset, R. Mohamed, O. M. Elkomy, and M. Abouhawwash, "Recent metaheuristic algorithms with genetic operators for high-dimensional knapsack instances: A comparative study," *Computers & Industrial Engineering*, vol. 166, p. 107974, Apr. 2022, doi: 10.1016/j.cie.2022.107974.
- [30] D. Saqui *et al.*, "NSGA2-based method for band selection for supervised segmentation in hyperspectral imaging," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Bari, Italy, Oct. 2019, pp. 3580–3585. doi: 10.1109/SMC.2019.8913846.

- [31] L. Brisacier-Porchon and O. Hammami, "Datasheet: where is requirements engineering gone? A case in Benchmarking Mobile Robotics," presented at the CSER, Trondheim, Norway, Mar. 2022.
- [32] L. Brisacier-Porchon and O. Hammami, "Inconsistent and Incomplete Datasheet: The case for systematic use of requirement engineering," presented at the 32nd Annual INCOSE International Symposium, Detroit, Michigan, Jun. 2022, p. 17.
- [33] A. K. Raz, C. R. Kenley, and D. A. DeLaurentis, "System architecting and design space characterization," *Syst Eng*, vol. 21, no. 3, pp. 227–242, May 2018, doi: 10.1002/sys.21439.
- [34] Phoenix Inc, "ModelCenter official Website." <https://www.phoenix-int.com/product/modelcenter-integrate/>
- [35] "Catia v5 website," 2022. <https://www.3ds.com/products-services/catia/products/v5/portfolio/>
- [36] "Cameo Systems Modeler on Dassault Systems website," 2021. <https://www.nomagic.com/products/cameo-systems-modeler>
- [37] G. Sirin, C. J. J. Paredis, B. Yannou, E. Coatanea, and E. Landel, "A Model Identity Card to Support Simulation Model Development Process in a Collaborative Multidisciplinary Design Environment," *IEEE Systems Journal*, vol. 9, no. 4, pp. 1151–1162, Dec. 2015, doi: 10.1109/JSYST.2014.2371541.
- [38] J. H. Bussemaker, P. D. Ciampa, and B. Nagel, "System Architecture Design Space Modeling and Optimization Elements," p. 14.