



HAL
open science

Populating MBSE Models from MDAO Analysis

Ombeline Aiello, David Sanchez del Rio Kandel, Jean-Charles Chaudemar,
Olivier Poitou, Pierre De Saqui-Sannes

► **To cite this version:**

Ombeline Aiello, David Sanchez del Rio Kandel, Jean-Charles Chaudemar, Olivier Poitou, Pierre De Saqui-Sannes. Populating MBSE Models from MDAO Analysis. 2021 IEEE International Symposium on Systems Engineering (ISSE), Sep 2021, Vienne (en ligne), Austria. pp.1-8, 10.1109/ISSE51541.2021.9582519 . hal-03562206

HAL Id: hal-03562206

<https://hal.science/hal-03562206v1>

Submitted on 8 Feb 2022

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.

Populating MBSE Models from MDAO Analysis

Ombeline Aiello*, David Sanchez Del Rio Kandel *[†], Jean-Charles Chaudemar *,
Olivier Poitou[‡] and Pierre de Saqui-Sannes*

*ISAE-SUPAERO, Université de Toulouse, France

ombeline.aiello, jean-charles.chaudemar, pierre.de-saqui-sannes@isae-supaero.fr

[†]EPFL, Lausanne, Switzerland

david.sanchez@alumni.epfl.ch

[‡] ONERA, Toulouse, France

olivier.poitou@onera.fr

Abstract—Over the past decade, Systems Engineering has switched from document-centric approaches to model-based ones. In this context, Model Based System Engineering (MBSE) and Multidisciplinary Design Analysis and Optimization (MDAO) have emerged as two complementary disciplines. How to combine MBSE and MDAO approaches for the benefits of systems engineers is still an open issue. This paper discusses a case study of coupling MBSE and MDAO. The MBSE part relies on SysML and timed automata, two modeling languages that are supported by the TTool and UPPAAL-SMC tools, respectively. The MDAO part is developed in the context of Open MDAO. The paper uses a drone as a case study and focuses discussion on battery usage. The SysML model of the drone is enhanced with a timed automata model of the battery. The SysML model of the battery is populated with results from MDAO analysis. In this context, combining SysML, UPPAAL-SMC and Open-MDAO offers to improve self-confidence in some values lying in the model, improve self-confidence in the requirements satisfaction, as well as to refine some requirements values with better accuracy.

Index Terms—MBSE, MDAO, SysML, timed automata, drone.

I. INTRODUCTION

Over the past few years, drones have been emerging around us in order to accomplish a broad variety of tasks. Drones have several advantages over piloted aircraft. Their smaller size makes them more maneuverable. Further, they are easy to transport from their storage place to their mission zone. In addition, they are more eco-friendly since they use electrical energy to fly. All of this can be achieved at a cost that is lower than the cost of using an aircraft [1].

Drones capture major design problems, particularly when battery management and autonomy are at stake. In particular, the acceleration and climbing phases require more power than the cruising phase does. This drastically reduces the autonomy of the drones. How to manage the state of drone batteries is therefore a complex issue that challenges MBSE approaches.

In this paper, the subject is addressed starting from a SysML [2] model with the perspective of checking the model against design errors and battery-specific requirements. The method proposed in [3] for SysML and free software TTool [4] is first implemented from requirement capture to use case

driven analysis, and to architectural and behavioral design. This paper shows that populating the SysML diagrams with accurate values turns impossible to achieve without adding two complementary modeling activities. First, MBSE in SysML is complemented by an MDAO approach using free software OPEN-MDAO [5]. Second, MBSE in SysML is complemented by MBSE in timed automata, using stochastic model-checker UPPAAL-SMC [6].

The paper is organized as follows. Section II surveys related work. Section III presents the drone that serves as running example throughout the paper. Section IV presents a method that encompasses MBSE and MDAO coupling. Section V explains how the battery of the drone has been modeled in SysML. Section VI and Section VII respectively use a MBSE-MDAO coupling and a MBSE-MBSE coupling to populate the SysML model. Section VIII discusses the potential and benefits of coupling MBSE and MDAO. Section IX concludes the paper and outlines future work.

II. RELATED WORK

Better communication/Information sharing and *Improved consistency* are respectively first and fourth in the top perceived and observed types of benefit from moving to a Model-Based approach of Systems Engineering [7]. Heterogeneous modeling assets being produced and used during a system engineering process, a precise description of their coupling seems to be a natural need.

This section surveys papers that couple systems engineering models. Discussion focuses on joint use of MBSE and MDAO, especially when SysML is the MBSE modeling language.

A. Coupling systems engineering models

In [8], models federation is introduced as a way of coupling systems engineering models. The main idea is to allow experts to stay focused on their legacy models. Commonalities (links between models) are exposed to a conceptual space via technology adapter (one technology adapter being dedicated to one legacy languages). When changes occur in a legacy model, those adapters are then used to maintain consistency among the model set. The approach is supported by the openflexo modeling infrastructure [9]. As far as we know, optimisation models and tools were not part of real life experiments even if

This work was supported by the Defense Innovation Agency (AID) of the French Ministry of Defense (research project CONCORDE N° 2019 65 0090004707501).

[10] gives an illustrative example that links a Microsoft Word asset with a UPPAAL one (via the XML Technology adapter).

In [11] Delmas, Doose and al. associate Model-Driven Engineering and constraints solver for automatic synthesis and quantitative design optimization. They rely on two input assets: a partial modeling (an Ecore partial model) of the system associated to a set of rules (encoded as constraints). These inputs are then processed by a SAT or a SMT solver to produce a full model description. The introduced constraint language is dedicated to the addressed use-cases and is not as general as OCL for example. In addition, the achieved optimizations are not multidisciplinary.

In [12] Leserf, de Saqui-Sannes and Hugues revisit the SysML language and a method associated with Papyrus tool to allow the system designer to make architectural choices in the early stages of the life cycle of systems. Papyrus is linked with solvers. A common point with the approach discussed in this paper is that the output of solvers is used to improve the requirement capture and analysis steps, and to populate the design diagrams. A major difference lies in the fact that this paper uses MDAO tools and not solvers.

B. Coupling MBSE and MDAO

In [13], Shah, Paredis, Burkhart and Schaefer present a framework that formally links SysML models coming from an MBSE approach to General Algebraic Modeling Systems for solving component sizing problems. Although the wording MDAO is not used, one aim and motivation mentioned for this work is to tackle contradictory constraints coming from different engineering disciplines. A GAMS metamodel is introduced and connected to SysML via an extension of the SysML profile specifying needed stereotypes. The study is then illustrated on a hydraulic log splitter design. Transforming the whole constraint problem to GAMS does not preserve the discipline separation that would occur in a strict MDAO approach and may require either a strong collaboration between experts from the different discipline, or that a single expert describes the whole system details. This may be a limitation when moving to a more complex case study.

In their survey [7], Henderson and Salado exhibit a statistical approach of MBSE benefits in the literature. The authors proceeded to register every benefit in the main engineering reviews and order them according to their types. Their empirical research shows that MBSE presents several benefits to improve engineering projects' quality. However, according to their survey, there is not yet a full agreement on what the main benefits of MBSE are.

In [14] Wolny, Mazak, Carpella, Geist and Wimmer survey and categorize SysML papers over the past thirteen years. The survey indicates that SysML and SysML tools have essentially been coupled with other MBSE tools and methods. The authors of [14] do not identify any paper coupling MBSE and MDAO, which open promising avenues for proposals as the one discussed in this paper.

III. DESCRIPTION OF THE DRONE AND ITS BATTERY

A. Mission and Drone selection

We consider a drone with a surveillance mission: flying over a high voltage line and recording data using embedded sensors. The drone may use a camera to evaluate the state of the wires and the pylons, and to analyze the vegetation.

This is a multiple-risk mission. Flight management becomes heavily complex and damage-prone because of collisions due to the proximity between the wires and the drone. Besides drone destruction, electrical cables could be damaged, thus provoking power failure in buildings that receive electric power from the high voltage line.

Weather conditions further influence the mission by limiting the speed of the drone. More power is required in hard weather conditions (wind, rain, storm, and so on) to ensure the lift of the drone. Having limited energy availability obliges to find a compromise in order to avoid the stall of the drone.

For all these reasons, this paper considers a fixed wing drone in charge of traveling a long distance.

The drone selected to accomplish the mission will not be entirely specified in this paper. Discussion focuses on the battery and its usage.

B. Battery

In most cases, drones are powered by Lithium Polymer (LiPo) batteries since the latter provide ten to thirty times the theoretical energy density supplied by lead batteries. Moreover, LiPo batteries are lighter than Nickel-Cadmium ones [15]. Concerning the battery condition, its state of charge (SOC) is the only parameter taken into account in this paper because the mission executed by the drone is one day long. Thereby one may neglect the deterioration of the battery due to its age. For this reason, the KiBaM model, largely presented in [16], is considered as a good candidate to model the battery used as a case study in this paper. Indeed, the KiBaM model provides a good representation of the SOC.

The KiBaM model is made up of two compartments (Figure 1). The left one represents the bound charge and the right one represents the available charge. The discharge of the battery is computed with ordinary differential equations (ODEs) (1) and (2). Once the available charge compartment is empty, the battery is considered as empty and cannot supply energy anymore.

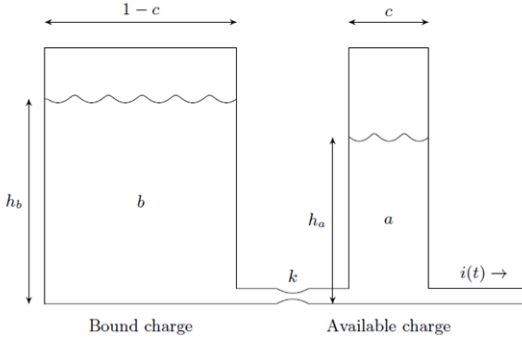


Fig. 1. KiBaM model

$$\frac{da}{dt} = -i + k(h_b - h_a) \quad (1)$$

$$\frac{db}{dt} = -k(h_b - h_a) \quad (2)$$

These notions are necessary to model the battery using MBSE techniques.

IV. METHOD

SysML (Systems Modeling Language) is an international standard at OMG [2] and originates from joint efforts of OMG and INCOSE (*International Council on Systems Engineering*) to define a modeling language for systems engineers. SysML is a notation defined independently of any tool and method. Consequently, SysML needs to be associated with a method supported by tools.

The method proposed in [3] identifies three major steps: requirement capture, use case driven analysis, and design. The output of the design step is an architecture of blocks where each block is internally described by a state machine.

This paper revisits the method proposed in [3]. The novelty in Figure 2 lies in the ‘red cross’ that cuts the direct link between the analysis and design steps. A derivation to MDAO indicates that parameters incompletely elicited during the requirement and analysis steps are transmitted to an MDAO tool to come up with a range of acceptable values for these parameters. The latter are subsequently used to populate the block and state machine diagrams that are created during the design step.

Figure 2 explains the general principles of coupling MBSE and MDAO. Figure 3 explains how these general principles have been instantiated on the surveillance drone.

Figure 3 distinguishes the mission to be accomplished by the drone from the drone itself. Drones indeed differ from other systems by the range of missions they may accomplish. Unlike aircraft that always transport passengers between two airports, drones may accomplish different types of missions, such as inspection or object transportation. The mission is modeled using requirement, activity and sequence diagrams that may be populated using MDAO analysis. This coupling of the mission model with MDAO analysis is not discussed in this paper,

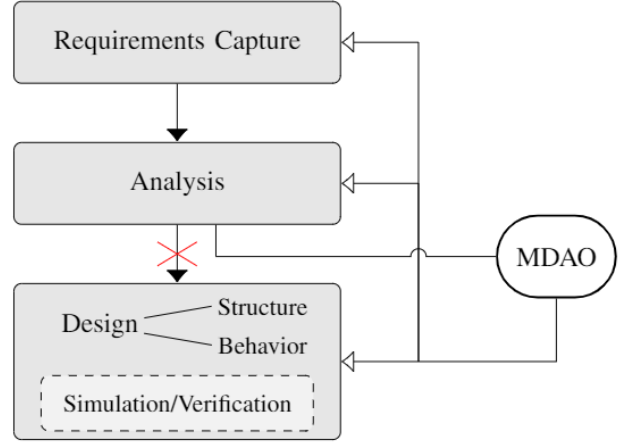


Fig. 2. A method coupling MBSE and MDAO

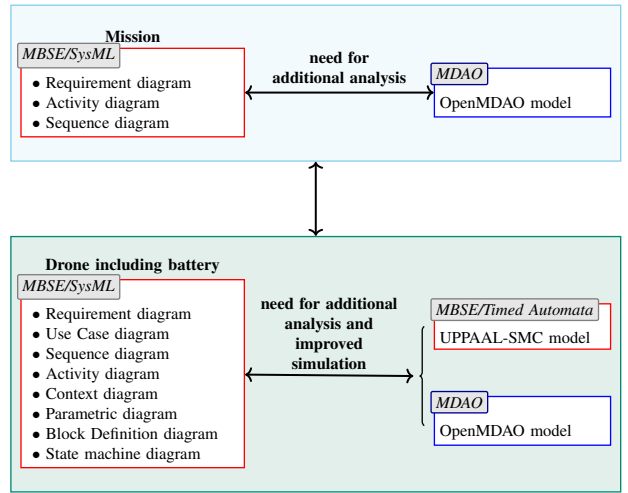


Fig. 3. Methodology followed in order to design the drone battery

which focuses on the drone itself and more particularly its battery.

The lower part of Figure 3 describes how the battery is designed in this paper. Red rectangles represent MBSE models whereas blue rectangles represent MDAO analysis. The SysML model of the battery has been populated with results from MDAO analysis and stochastic model checking of timed automata.

V. DRONE BATTERY MODELING IN SYSML

To take up the methodology introduced in Figure 2, all requirements needed to satisfy the mission are listed in a requirement diagram, an excerpt of which is given in Figure 4.

To take full benefits of drones advantages, increasing batteries operating time becomes essential. The requirement *Autonomy* identified by Bat_R1 in the requirement diagram adds precision to the Battery requirement. Bat_R0 and Bat_R1 are linked by a << refine >> relationship.

A simplified state machine diagram of the drone battery is presented in Figure 5.

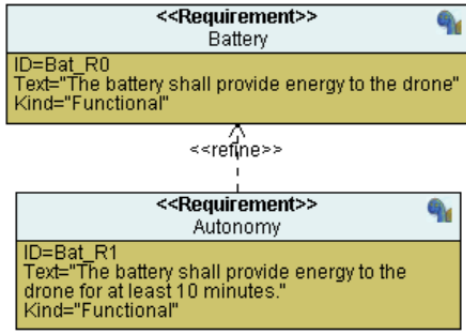


Fig. 4. Excerpt of the SysML Requirement Diagram

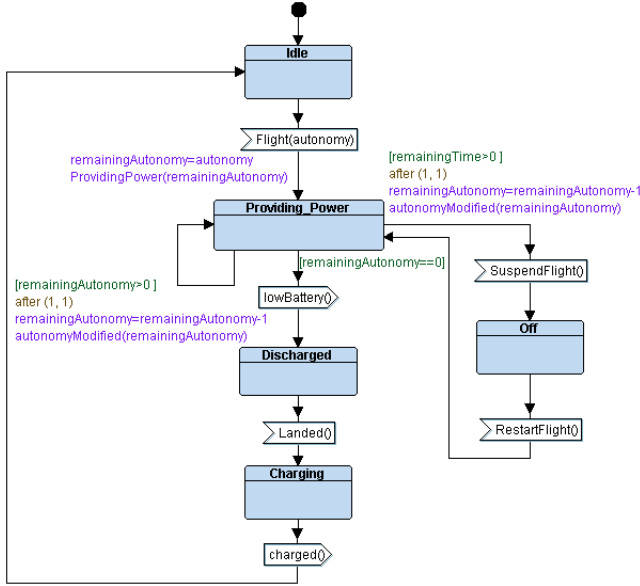


Fig. 5. Battery state machine diagram

When the drone is turning on, the battery is discharging, providing the power required to ensure the flight. In Figure 5, the autonomy of the battery is known before the beginning of the flight. Autonomy is expressed in percentages and decreases of one percent each time the loop transition in Figure 6 is executed. In other words, there is no precise computation done in the battery state machine modeled in SysML.

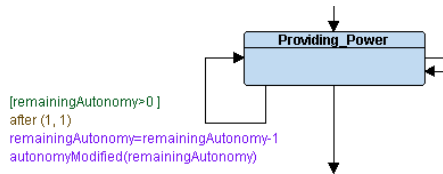


Fig. 6. Loop taken from the battery state machine diagram

This paper proposes solutions to overcome the lack of accurateness in the SysML model of the battery. First, by coupling MBSE and MDAO approaches, as detailed in section VI. Second, the SysML battery model is enhanced relying on statistical model checking of timed automata, as explained in

section VII.

VI. MBSE-MDAO COUPLING TO MODEL A DRONE BATTERY

A. MDAO background

Multi-disciplinary analysis and optimisation (MDAO) [5] is a branch of optimization that aims at solving optimization problems by involving several disciplines. For instance, since the early 80's, MDAO has been applied to aircraft optimization considering aero-structural optimization designs. The benefits of MDAO are twofold. First, an assessment of the impact of individual design changes on the global design. Second, a trade-off support for early system design and conception. To do so, MDAO relies on a strict mathematical and graphical notation: the Extended Design Structure Matrix (XDSM).

In [17] Martins and Lambe propose XDSM as a new methodology to unify already existing MDAO problem architectures. XDSM relies on a Design Structure Matrix approach that represents a system (or a project) as a square matrix. In addition, XDSM extends the previously defined DSM to show data dependency and process flows in the same single diagram. Further, it includes independent data flow lines, iterative process definitions, and ordering processes. XDSM is based on a specific yet common terminology containing the optimization, disciplines, variables and functions used in MDAO problems.

To interface the XDSM problem representation with relevant software frameworks, the Whatsopt plugin was implemented [18]. The latter links the XDSM graphical representation of MDAO problems with the Python OpenMDAO library. It makes implementation of MDAO problems simpler since the user only needs to take care of the graphical aspect of the problem. OpenMDAO is an open source framework developed by the NASA Glenn Research Center and designed to efficiently solve MDAO problems. Written in Python, OpenMDAO supports gradient based optimization with analytic derivatives [19].

OpenMDAO allows the computation of total derivatives (derivatives of models outputs respect to model inputs) very efficiently. As total derivatives are usually hard to express and are costly in terms of computer power, OpenMDAO relies on analytic methods to implement them. Whatsopt enables collaboration between research scientists and engineers on a common framework while using advanced optimization techniques. In the Whatsopt plugin, Python code skeletons are automatically generated and the user may fill them with the values needed for the problem resolution.

B. MDAO model

In [20], Chaudemar and de Saqui-Sannes state that joint use of MBSE and MDAO techniques may help taking design decisions and thus populating the design diagrams and the tools enabling checking of design diagrams.

In this section, the coupling of MBSE and MDAO considers SysML on the MBSE side and OpenMDAO on the MDAO side. Figure 7 depicts the XDSM characterizing the parameters

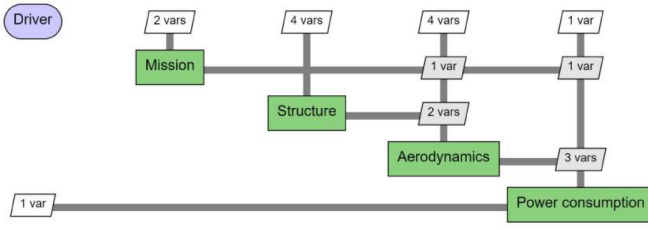


Fig. 7. XDSM matrix for the power minimization problem

to be taken into account to evaluate the power consumed by the drone.

The work presented in this section relies on mathematical models and equations proposed by Zeng and Zhan [21] to derive aircraft's power consumption and energy based on its aero-structure and the external forces applied on it.

The MDAO model in Figure 7 has the form of a function of several parameters. From that MDAO model, we derive the minimal power consumption of the drone. The parameters are part of the aircraft's structure (wingspan, wet surface) and so are the external constraints intrinsic to the mission, such as the wind speed and the payload mass. Their values lie in SysML diagrams (block definition diagrams and requirements diagrams) and feed the MDAO tool. The minimal power consumption obtained as an output of the MDAO analysis helps populating the state machine and requirement diagrams of the SysML model. Thereby, MDAO analysis outputs accurate analytical values that can be reused to compute the state of charge of the drone's battery (see section VII).

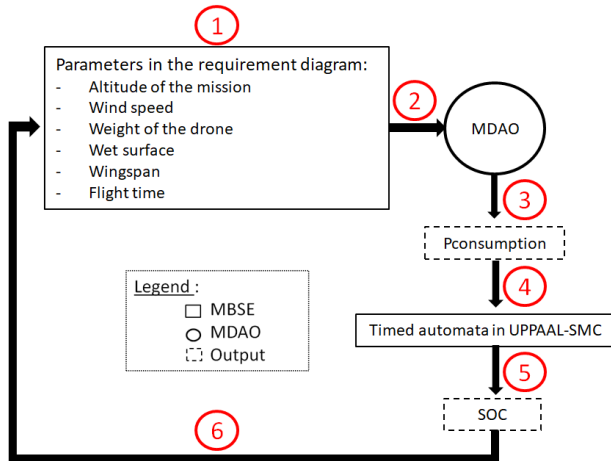


Fig. 8. Populating process

Figure 8 depicts a six-step process:

- 1) The parameters are extracted from the requirements expressed in the SysML requirement diagram.
- 2) The MDAO model is created. The function to be optimized and the constraints are defined in a model that is provided as input of the MDAO analysis tool.
- 3) MDAO analysis outputs one parameter: the minimal power consumption.

- 4) A timed automata model of the battery is created. It is consistent with the SysML state machine of the battery. This timed automata model is populated with the power consumption value obtained at previous step.
- 5) The timed automata model is provided as input of statistical model checker UPPAAL-SMC. The later outputs the state of charge of the battery depending on the flight phase. For instance, the battery will loose more energy during the climb phase as compared to the cruise phase.
- 6) The state of charge is checked against the flight time it permits. This action offers an opportunity to rewrite the requirements associated with the flight time and to make the latter more realistic.

Table I shows the results provided by MDAO analysis.

Climb power [W]	Cruise power [W]	Descent power [W]
290	169	86

TABLE I
POWER VALUES

VII. SYSML AND TIMED AUTOMATA COUPLING

The use of the ODEs (1) and (2) obliges to model the battery as a hybrid automaton. There is a need for a model-checker capable to solve this type of problem: UPPAAL proposes an extension named Statistical Model Checker (SMC) offering the possibility to estimate and compare probabilities [6]. Therefore, UPPAAL-SMC is adopted in this paper to complete the design of the drone battery. Like TTool, UPPAAL-SMC is also an MBSE tool.

Gathering information provided by SysML diagrams and by MDAO analysis, all input data required to build the UPPAAL-SMC model become available. Some of these input data are given in Figure 4, Figure 5 and Table I.

The objective of the UPPAAL-SMC model is to predict the discharge of the battery along the flight of the drone. It is assumed that the battery does not provide energy to other devices, such as cameras and sensors. To respond to the aims set out, a model comprised of three automata is used. These three automata can be textually described:

- 1) KiBaM : it replicates the discharge of the battery following the reaction presented in Section III. This model has been used in various papers (e.g., [22], [23]), and is therefore not shown in this paper.
- 2) Controller : the second automaton triggers actions that take place in each flying phase. Its objective is to oblige the model to take transitions in a given time interval. This way, the totality of the steps modeling the flight of the drone can be undertaken in a correct order and in an acceptable period of time.
- 3) Flight : the last automaton is depicted by Figure 9. It allows one to model the flight of the drone. For the sake of simplicity, the automaton called Flight contains only three phases of flight: 'climb', 'cruise' and 'descent'. Each phase is characterized by a specific current, respec-

tively, I_{climb} , I_{cruise} and $I_{descent}$. Flight automaton is greatly inspired by the Task Model presented in [22].

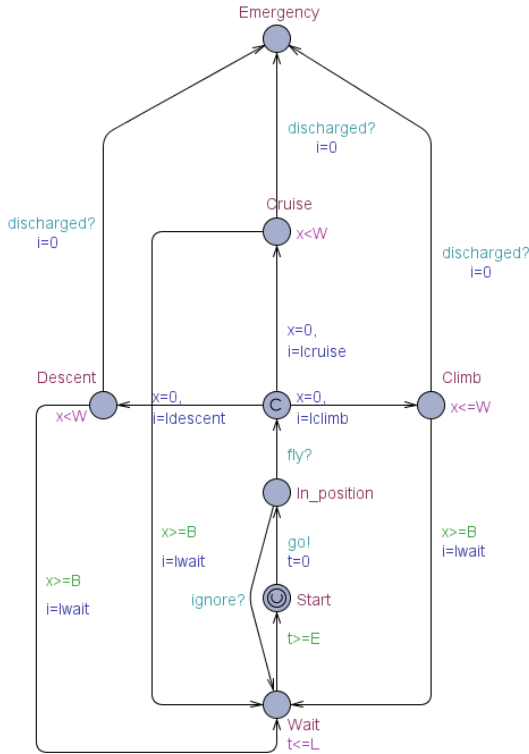


Fig. 9. Illustration of the automaton Flight

During climbing phases, the power required by the drone is more important than in cruising flight. Similarly a descent phase requires less power than a cruising one does. These are the reasons why three different powers were computed (Table I).

Thus, battery autonomy varies depending on the power delivered in the moment. As a reminder, the entire charge of the battery is contained in the two compartments presented in Figure 1. To avoid hazardous situations, it is necessary to precisely know the state of charge of the battery during the flight. The simpler way to estimate it is to use the following formula :

$$SOC(t) = \frac{Q(t)}{Q_i} \quad (3)$$

Where $SOC(t)$ is the state of charge (in percentage) at a time t , $Q(t)$ is the remaining capacity at a time t , and, Q_i is the initial capacity of the battery. The obtained result is displayed in Figure 10. It shows the discharge of the battery along the mission.

The abscissa axis corresponds to the flight time of the drone. The ordinate axis characterizes the remaining percentage of charge in the available compartment.

UPPAAL-SMC allows one to compute several probabilities, to compare a probability to a given value and to compare two

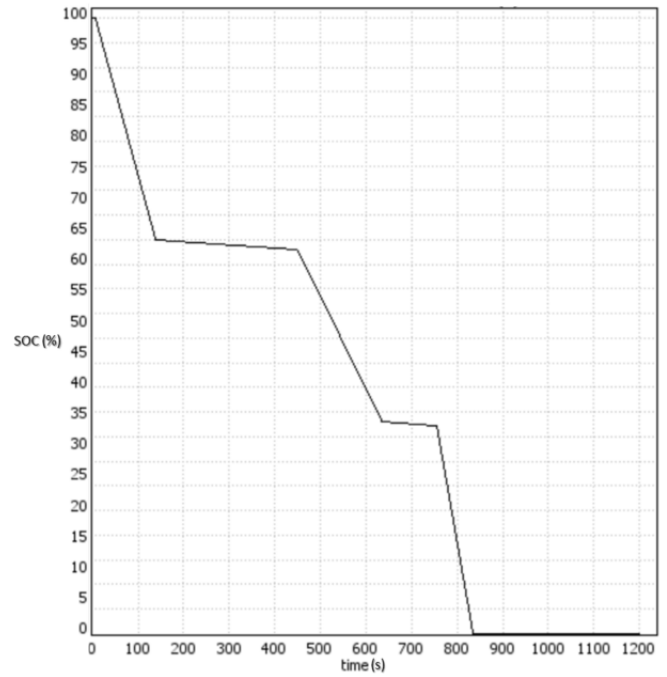


Fig. 10. Time-dependent State of charge

probabilities without computing them. Examples of probability computation and probability comparison are presented below.

- Property 1: it estimates the probability that the battery is empty before 780s (13 minutes) of flight. UPPAAL-SMC provides the result presented in Figure 11.

```
(402 runs) Pr (<> ...) in [0.452505,0.552434]
with confidence 0.95.
Pr [<=780] (<>Flight.Empty) [0.452505,0.552434]
```

Fig. 11. Verification of the first property

After 402 runs UPPAAL-SMC concludes that the probability for the battery to be empty before 780s of flight is in the range [0.45; 0.55].

- Property 2: verify if there is more than 45% of chance to have an empty battery after 600s of flight (10 minutes). UPPAAL-SMC provides the result presented in Figure 12.

```
(2230 runs) H1:
Pr (<> ...) <= 0.44
with confidence 0.95.
Pr [<=600] (<>Flight.Empty) >= 0.45
```

Fig. 12. Verification of the second property

After 2,230 runs UPPAAL-SMC concludes that property 2 is not satisfied. In other words, the risk to have an empty battery flying less than 10 minutes (600 seconds) is low : less than 45%.

- Property 3: verify if it is more likely to have an empty battery before 780s of flight than before 600s of flight. UPPAAL-SMC provides the result presented in Figure 13.

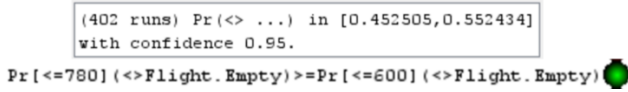


Fig. 13. Verification of the third property

After 446 runs UPPAAL-SMC concludes that property 3 is always satisfied. Property 3 contributes to verify the consistency of the model.

Let us compare the requirement Bat_R1 in Figure 4 to the UPPAAL-SMC results obtained in Figure 10, Figure 11, Figure 12 and Figure 13. It is possible to confirm that the initial requirements are still satisfied at the end of the battery design.

Precision can be added to Bat_R1. Looking at Figure 10, the autonomy of the battery estimated for the mission is 800 seconds when keeping 10% of charge. This corresponds to 13 mins and 20 seconds of flight instead of the 10 mins from the initial estimation. The minimal flight time requirement Bat_R1 found in Figure 4 can thus be updated as shown in Figure 14.

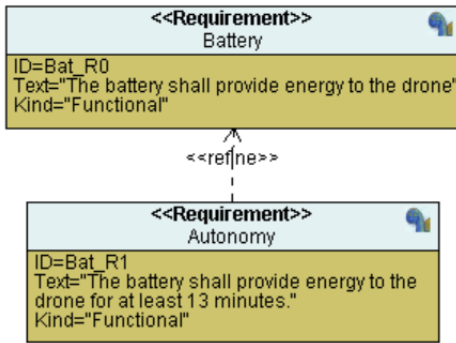


Fig. 14. Updated requirements diagram

The design of the drone battery presented in this paper is now completed. An original method based on the one presented in [3] was used. Advantages and limitations identified during the resolution of the case study addressed in this paper are discussed in section VIII.

VIII. DISCUSSIONS AND FURTHER WORK

MDAO is able to take into account many parameters coming from a wide range of disciplines. In this study, the analysis was restricted to three disciplines: Mission, Structure and Aerodynamics (section VI).

Figure 3 shows the data exchange between SysML and the external tools used in this paper. Moreover, it shows

the communication between two different modeling levels : Mission and Drone battery.

The remainder of this section discusses the work presented in this paper in terms of languages, tools and method.

A. Languages

A SysML model of the drone battery has been developed. Exposition in this paper focuses on the SysML requirement diagram and the way coupling of the SysML model with external MBSE and MDAO models enables improving the accurateness of the values expressed in the requirements. Further investigations are needed on populating other diagrams used during the analysis and design steps of the life cycle of systems.

B. Tools

In this paper, the minimal power consumed during each phase of flight is determined. These values are then recovered by the UPPAAL-SMC model. By combining them with the drone state machine diagram, simulations and probabilities computations are done.

Experience in coupling a SysML model with MDAO analysis has confirmed that launching MDAO analysis or other MBSE techniques from SysML models should be as much as possible automated in such a way analysis results may be automatically integrated into the SysML diagrams. One of the difficulties will be to harmonise all languages to ensure the consistency of the results and the system models.

Once such a stabilized MBSE-MDAO common platform will be set out, the question will arise of adding other tools used to achieve requirement traceability and simulation/verification of SysML models. The earlier a potential mistake is detected, the lower the impacts on the project are.

C. Method

This paper associates SysML tool TTool with OpenMDAO and UPPAAL-SMC tools. Each time new results or modifications are obtained by MDAO or MBSE models, previously completed steps are verified and corrected if necessary. This contributes to implementing an iterative modeling approach that deserves development of methodological assistants [24] that will help SysML model designers.

IX. CONCLUSIONS

Over the past decade, Model Based Systems Engineering (MBSE) and Multidisciplinary Design Analysis and Optimization (MDAO) have been investigated separately. In [20] Chaudemar and de Saqui-Sannes have identified several benefits of coupling MBSE and MDAO approaches. This paper goes one step forward and instantiates the ideas developed in [20].

A drone serves as running example throughout this paper and discussion focuses on battery usage. Two types of coupling are presented. First, a MBSE-MDAO coupling uses OpenMDAO for MDAO analysis. Second, a MBSE-MBSE coupling based on SysML and timed automata uses SysML tool TTool and stochastic model checker UPPAAL-SMC.

The work presented in this paper is a preliminary study in a more general study on coupling MBSE and MDAO. This coupling deserves further, formalized, investigations in terms of languages, tools and method. In terms of languages, defining a SysML profile is merely one option among different modeling languages coupling techniques to be compared. In terms of tools, the idea of populating SysML models with results from MDAO analysis must be implemented by paying attention to user-friendly and smooth integration of MDAO analysis results into the SysML model, in such a way SysML practitioners may apply the proposed approach without becoming MDAO literates. In terms of methods, attention will be paid to keeping the method not exclusive to drones.

ACKNOWLEDGEMENT

TTool has been developed by Prof. Ludovic Apvrille and his team at Telecom Paris, Sophia Antipolis, France. *WhatsOpt* has been developed by Rémi Lafage and his team at ONERA, Toulouse, France.

REFERENCES

- [1] James V. Marcaccio, Chantel E. Markle, and Patricia Chow-Fraser. Use of fixed-wing and multi-rotor unmanned aerial vehicles to map dynamic changes in a freshwater marsh. *Journal of Unmanned Vehicle Systems*, 4(3):193–202, September 2016.
- [2] OMG. *OMG Systems Modeling Language*. Object Management Group, <https://www.omg.org/spec/SysML/1.5>, 2017.
- [3] Ludovic Apvrille, Pierre de Saqui-Sannes, and Rob A. Vingerhoeds. An educational case study of using sysml and ttool for unmanned aerial vehicles design. *IEEE Journal on Miniaturization for Air and Space Systems*, 1(2):117..129, 2020.
- [4] TTool. <https://ttool.telecom-paris.fr/>. Retrieved March 15, 2020, 2021.
- [5] Justin S. Gray, John T. Hwang, Joaquim R. R. A. Martins, Kenneth T. Moore, and Bret A. Naylor. OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, April 2019.
- [6] Alexandre David, Kim G. Larsen, Axel Legay, Marius Mikučionis, and Danny Bøgsted Poulsen. Uppaal SMC tutorial. *International Journal on Software Tools for Technology Transfer*, 17:397–415, August 2015.
- [7] Kaitlin Henderson and Alejandro Salado. Value and benefits of model-based systems engineering (mbse): Evidence from the literature. *Systems Engineering*, 24(1), January 2021.
- [8] Christophe Guychard, Sylvain Guerin, Ali Koudri, Antoine Beugnard, and Fabien Dagnat. Conceptual interoperability through Models Federation. In *Semantic Information Federation Community Workshop*, Miami, United States, October 2013.
- [9] *Openflexo project*. <http://www.openflexo.org>, 2012.
- [10] Fahad Rafique Golra, Fabien Dagnat, Jeanine Souquières, Imen Sayar, and Sylvain GUERIN. Bridging the Gap Between Informal Requirements and Formal Specifications Using Model Federation. In Ina Schaefer Einar Broch Johnsen, editor, *16th International Conference on Software Engineering and Formal Methods (SEFM 2018)*, volume 10886 of *Software Engineering and Formal Methods 16th International Conference, SEFM 2018, Held as Part of STAF 2018, Toulouse, France, June 27–29, 2018, Proceedings*, pages 54–69, Toulouse, France, June 2018. Springer.
- [11] Rémi Delmas, David Doose, Anthony Fernandes Pires, and Thomas Polacek. Supporting model based design. In Ladjel Bellatreche and Filipe Mota Pinto, editors, *Model and Data Engineering*, pages 237–248, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [12] Patrick Leserf, Pierre de Saqui-Sannes, and Jérôme Hugues. Trade-off analysis for sysml models using decision points and csps. *Software and Systems Modeling*, 18(6):3265–3281, 2019.
- [13] A. A. Shah, C. J. J. Paredis, R. Martins Burkhart, and D Schaefer. Combining mathematical programming and sysml for automated component sizing of hydraulic systems. *SME. J. Comput. Inf. Sci. Eng.*, 2(4):041006, December 2012.
- [14] S. Wolny, A. Mazak, C. Carpella, V Geist, and M. Wimmer. Thirteen years of sysml: A systematic mapping study. *Software and Systems Modeling*, 19:111–169, 2020.
- [15] Wolfgang H. Meyer. Polymer Electrolytes for Lithium-Ion Batteries. *Advanced Materials*, 10(6):439–448, 1998.
- [16] James F. Manwell and Jon G. McGowan. Lead acid battery storage model for hybrid energy systems. *Solar Energy*, 50(5):399–405, May 1993.
- [17] Andrew Lambe and Joaquim Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 08 2012.
- [18] Rémi Lafage, Sebastien Defoort, and Thierry Lefebvre. WhatsOpt: a web application for multidisciplinary design analysis and optimization. In *AIAA Aviation 2019 Forum*, page 2990, 2019.
- [19] Justin Gray, John Hwang, Joaquim Martins, Kenneth Moore, and Bret Naylor. Openmdao: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 04 2019.
- [20] J.C. Chaudemar and P.d. Saqui-Sannes. Mbse and mdao for early validation of design decisions: a bibliography survey. In *IEEE SysCon 2021*, 2021.
- [21] Yong Zeng and Rui Zhang. Energy-efficient uav communication with trajectory optimization. *IEEE Transactions on Wireless Communications*, 16(6):3747–3760, 2017.
- [22] Erik Ramsgaard Wognsen, René Rydhof Hansen, and Kim Guldstrand Larsen. Battery-Aware Scheduling of Mixed Criticality Systems. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications*, volume 8803, pages 208–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. Series Title: Lecture Notes in Computer Science.
- [23] Waheed Ahmad, Marijn Jongerden, Marielle Stoelinga, and Jaco van de Pol. Model Checking and Evaluating QoS of Batteries in MPSoC Dataflow Applications via Hybrid Automata. In *2016 16th International Conference on Application of Concurrency to System Design (ACSD)*, pages 114–123, Torun, June 2016. IEEE.
- [24] Erika Rizzo Aquino, Pierre de Saqui-Sannes, and Rob A. Vingerhoeds. A Methodological Assistant for Use Case Diagrams. In *8th MODEL-SWARD : International Conference on Model-Driven Engineering and Software Development*, pages 1–11, La Valette, Malta, February 2020.