



**HAL**  
open science

## Inconsistent and Incomplete Datasheet: The case for systematic use of requirement engineering

Lorraine Brisacier-Porchon, Omar Hammami

### ► To cite this version:

Lorraine Brisacier-Porchon, Omar Hammami. Inconsistent and Incomplete Datasheet: The case for systematic use of requirement engineering. 32nd Annual INCOSE International Symposium, Jun 2022, Detroit, United States. hal-03767046

**HAL Id: hal-03767046**

**<https://hal.science/hal-03767046>**

Submitted on 1 Sep 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.



32<sup>nd</sup> Annual **INCOSE**  
international symposium  
hybrid event

Detroit, MI, USA  
June 25 - 30, 2022

# Inconsistent and Incomplete Datasheet: The case for systematic use of requirement engineering

Lorraine Brisacier-Porchon  
ENSTA Paris

826, Boulevard des Maréchaux 91190  
Palaiseau

[lorraine.brisacier@ensta-paris.fr](mailto:lorraine.brisacier@ensta-paris.fr)

Omar Hammami  
ENSTA Paris

826, Boulevard des Maréchaux 91190  
Palaiseau

[Omar.hammami@ensta-paris.fr](mailto:Omar.hammami@ensta-paris.fr)

Copyright © 202X by Author Name. Permission granted to INCOSE to publish and use.

**Abstract.** The lifecycle of a system is extended from its early conception to its retirement of service. The lifespan of Unmanned Ground Vehicles (UGVs) can be expected to last over 50 years in the defense market. In this context, the rising complexity of UGV systems imposes engineering steps that would ensure both capabilities of the system and resilience to its future inclusion in a system-of-system context. During its operational usage, the UGV is supposed to be maneuvered for specifically designed purposes following user manual datasheet of the components off-the-shelf (COTS) that were integrated. This paper exposes the public user datasheet relevance compared to the system engineering requirements that are the artifacts of system design architecture. The use of connecting COTS user manual to system requirements is discussed, all the more if the systems are to be re-used in a system production line. This article is intended to explore system of system conception methods for future robotized battlefield.

## I. Introduction

Mobile robots are found in a wide spectrum of applications, such as farming, excavation, demining, or military. There are categories of usage and size for ground robots. They are considered as complex systems, because the cost of testing or simulating a significant amount of their behavior exceeds the investment and time that could be spent on design. Therefore, methods and process for prototype documentation is the key to manage its behavior. In this article, we consider the Unmanned Ground Vehicle (UGV) basic functions are all-terrain navigation, heavy loads carrying, minimal maneuverability while being piloted. In the defense domain, where we intend to include swarm of robots, the UGV shall be robust, man walking fast, with the highest battery autonomy. Among others, we benchmarked a robot off the shelf, as if it were a product in development in a fictional company. The supposed expected performance for our system is presented in Table 1 System specification from datasheet. It shows small-scale problems assessed in complex system engineering, and addresses a test scenario that draws integration of a robot in system of system battlefield.

The system mission on the field will be referred to as “mule concept”, which includes remote piloting a mule that would carry loads instead of the pilot in a predefined time at a predefined speed. The degree of autonomy given to the mule is not at stake in this article. We will consider that the capability of “carrying loads” has been exploited to describe the system requirements, and that this set of requirements has been used to build the robot.

In this article, we place ourselves as buyers of such a UGV system. Making the acquisition of such a robot represents an amount of money, and we describe a method to verify the systems expectations.

As described in (Stevens 2017), validation and verification (V&V) time and costs are a gamble for the company as well as the client. In this article, we explore how to use the V&V system engineering documentation in other contexts: user manual description and next product specification reuse.

The goal of this article is to discuss the relation between requirements, system V&V, system capabilities and multi-objective optimization as mathematical foundation for systems engineering. Figure 1 System driven engineering flow shows a process proposition for system development that includes optimization techniques. This foundation could lead to the creation of a method to shift from system engineering process as described in (AFNOR 2015) to a digitally based process (Henderson et al., n.d.) that would directly generate the best suited methods and tool for the mule system design in our fictional company.

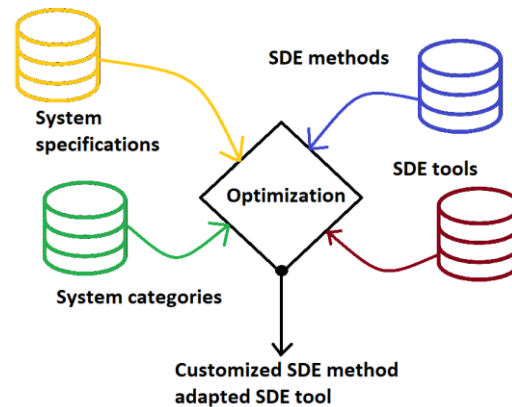


Figure 1 System driven engineering flow

## II. State of the Art

### a. Requirements engineering

Requirement based engineering is the most popular practice for defense system engineering acquisitions and architecture design (Huldt and Stenius 2019). The state of the art on requirement redaction is consistent, and recent publications shows that the interest in “good” requirements expression exceeds the systems engineering domain.

As Model-based engineering (MBSE) practices progress in being associated to requirements (Bonnet, Voirin, and Navas 2019) the definition of system in early design is improving as complexity challenges rises. The return on investment of such practices (Duffy et al. 2021) has been identified. The vision of INCOSE for 2025 (INCOSE 2015a) seems to be carried out. Furthermore, the vision of the future (Voirin et al. 2020) extends MBSE to all models to implement the best systems in the most reduced cost and delays is being profiled.

The profile of model specifications could be adapted to the system context and categorization (Younse, Cameron, and Bradley 2021). It is a widely accepted prolongation to the SysML 1.3 success in System engineering (Wolny et al. 2020), because it adds views and connections to the system requirement expressions. The model aspects based on SysML does not express all required aspects of system design, which makes it incomplete, such as cited in (Younse, Cameron, and Bradley 2021). This is a motivation for acquirers to specify systems in architecture frameworks, and to dig deeper on semantic approach (Duprez and Ernadote 2020) for system specification ambiguity reduction.

Publications on MBSE do not cover all expected transdisciplinary system behavior representations (Watson et al. 2020): mechanical or electrical requirements are rather considered as Model-Driven

Engineering (MDE). If those are the modeled expression of expected behavior, MBSE fails to represent all transdisciplinary aspects of the system. Therefore, the performance of the system will remain ambiguous until tests are performed in a real environment. The system scale compromises defined in system engineering cannot be represented

To the best of our knowledge, none of these foretold systems engineering methods are the basis of system user manual generation. On the opposite, the teaser of requirement-based methodology state of the art is oriented in a top-down approach on the very early stages of system design. (Hahn et al. 2020) states that placing the ‘design phase’ above all considerations in order to gain time and/or costs is no guarantee of money efficiently spent. The article rather states the need for methods and tools tailored to the expected outcome of the product. If whole system lifecycle, for instance 50 years, is taken in consideration, focusing efforts in modeling and requirement expression might be more efficient if it is used during all the lifespan of the system rather than until the system architecture is conceived.

The application of requirements seldom includes the generation of end-user system datasheet. This can be caused by the difference of model details required at the process stage: the user manual exposes details of finite usage of the system, whereas the concept elaboration requires high level information. The mix of model granularity and fineness blurs decision making. The problem of multiple level of models that work together is addressed in Digital Twin usage questions such as in (Bachelor et al. 2020), but no information can be found on how to mix them.

To conclude, system specifications is used to engineer the systems. User manual is written to make the best usage of the system. Yet, the effort to ensure maintenance and usability seems to be redundant with the effort to specify the system, especially during its integration phase. Since we have not found publications that would imply this, we expose a method that establishes a connection between requirements and datasheet to avoid redundancy. This two-ways connection could simultaneously improve the next-in-line product while reducing the costs and delays of engineering in comparison to its predecessor. The effect of our proposition will also be beneficial to the end-users, because system capabilities description as requirements should be used to generate user manuals.

### ***b. Experienced feedback on MBSE and IVV of complex systems to check the COTS engineering quality***

The engineering process that has resulted in the creation of the COTS is easily identifiable from the behavior of the system. If the supplier has executed system engineering to design the COTS for the system, then the system passed a V&V qualification procedure. The testing environment that resulted in the system performance claims would be easy to reproduce. The method applied in this article is exposed in that way: what if the acquirer of the UGV COTS were to verify the system capabilities?

The set-based vision of verification strategies (Xu and Salado 2019) would be of help in passing the tests, but we would want to keep close to real-life situations where the scenario for our mule concept is too simple to afford long considerations before use. Furthermore, the changes in process and company organization to make this set-based vision a reality leaves the last call decisions to business investment. As resumed in (Huldt and Stenius 2019), a lack of knowledge to integrate a model-based approach with current business processes is one of four arguments that prevent the development of MBSE. If the views are not provided, the common business/engineering vision cannot be built. Investment in “best effort” often prevails.

There are also publications measuring the benefits of the MBSE approach in managing the complexity of a robotic space system (Younse, Cameron, and Bradley 2021). But it makes no mention on how to re-use the specifications and how to organize all verifications that cannot be represented in SysML 1.3-like languages, such as measuring the tension of the battery in relation with the complete vehicle system. The multi-disciplinary requirements that should be the basis of system scale V&V

cannot be represented in one model, and the best suited method and tools options for Software development objectives cannot be the best compromise on a multi-objective point of view.

The absence of formal expressions, methods or process associated to the language, as well as to poor investment in early stages of requirement specification (INCOSE 2015b) is proven to be a plausible cause of ambiguities in engineering (Duprez and Ernadote 2020). But there is no practical mention on how to enhance the existing process while reducing engineering costs and time in engineering. Furthermore, the mention of a model that would measure system outcome linked with investment in testing methods and tools for all physical performances is not discussed. Furthermore, the client satisfaction depends on how well the announced performances are met as described in datasheet. To the best of our knowledge, there are no recent research that would combine the user manual descriptions and qualification datasheet, contractors trust and risk management. This research question was raised in (Stevens 2017) but not exploited yet from a sub-system contractor point of view.

### ***c. Theoretical proposition: optimization process***

The state of the art in multidisciplinary and multi objective optimization is also consistent. The mention of a need for a semantic extension to MBSE (Wach and Salado 2020), is still an issue to cover specific system categories and requirements and there are no quantified application feedback of those methods. (Pate, Gray, and German 2014) presents graphical representations of non-dominated compromises in a system design evaluation. Optimization offers a consistent model for multidisciplinary approach. But it contains no link to a specific system goals or category in particular. We have chosen to use anonymized datasheet on the specifications used as our usage profile for the UGV system in order to organize the tests sets the optimization objectives as systems engineering process, tools and methods selection guide, depending on equation:

$$(Q(x) \rightarrow \max, R(x) \rightarrow \min, T(x) \rightarrow \min) \text{ (GOAL 1)}$$

The engineering flow proposed is selected by computer assisted search in which illustrated in Figure 1. It proposes system-driven engineering, based on four pillars: specifications, system categories, method, tool. All pillars are linked with optimization through a common language: a combination of models that are either objectives or constraints. Its main advantage is the focus on the multi-disciplinary decisions that can be made, evaluating simultaneously the best process options for system development, and at the same time sketching the expected resulting system architecture. Detailed design costs are anticipated in that way. Figure 1 illustrates a proposition of method that would implement SDE. Instead of standard system specification and validation with a “V” or “W” cycle, the representation focuses on the revision of specification (functional and non-functional), functional architecture, component modeling, physical architecture followed by multi-objective optimization in order to pick the most valuable information from the previous specifications. In this article, Quality maximization is at glance. It is called Method-Tools-System Engineering Problem (MTS-EP). It states Q as a function depending on system specifications and category. Let  $x \in S$  be a system design possibility. The MTS-EP problem can be formulated as:

$$Q(x) \rightarrow \max \text{ subject to } g(x) \leq 0 \text{ and } h(x) = 0 \text{ (MTS-EP 1)}$$

The goal for Q maximization is therefore to find the  $x \in S$  combinations that achieve the system defined objectives subject to constraints. The detail of the mathematical model is related to the investment and time input to the system concept elaboration. This problem alone is NP-Hard based on Knapsack problem. Functions g and h in (MTS-EP 1) are the mathematical expression of constraints of the problem, issued from specification, system category, and company’s system experience from past projects. The flow in Figure 2 matches the representations for adoption of MBSE Methodology sketched in (Wu et al. 2019), and adds a constraint that is not underlined in Wu’s publications, which is the end of the process, when allowed budget or schedule for system conception ends. This is an

application of the principles of (Wheaton and Madni 2018) on UGV system category: the identification of a Pareto Front where the system scale budget and schedule constraints the concept choices.

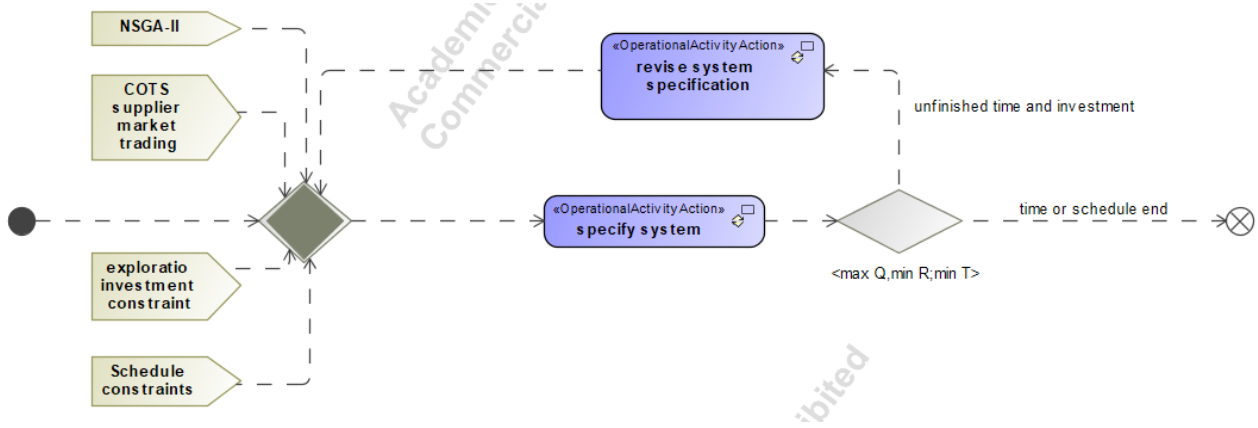


Figure 2 : System specification flow proposition inspired of (Wu et al. 2019)

In this article,  $x$  is instantiated by the UGV as it was designed and integrated. We want to put the UGV to a test and evaluate its Quality as a mule system, and evaluate the quality of its specification. If the system concepts definition quality is seen as an optimization problem, we could evaluate the performance of the acquired system with respect the specification, and at the same time make propositions that would simultaneously increase quality of the system and reduce method and tools investment to an acceptable compromise. The expected outcome of our System Specification Flow Proposition SAFP can be measured in the next product that inherits of lessons learned from the previous solution. If system definition ambiguities can hardly be measured, its decrease can be measured. The adoption of our method can result in mastering of current expenses on concept exploration, direct generation of user manual automatically generated from the validation datasheet, maybe even obsolescence management, following (Morgan, Holzer, and Eveleigh 2021) adding multi physical dimensions to their claims.

#### ***d. A mobile robot: UGV system instance description***

Our instance is an all-terrain unmanned ground vehicle (UGV) for rapid prototyping and research applications and a robotic development platform. The capabilities of UGV can be expanded by multiple accessories such as a LIDAR, a GPS, a Camera and an IMU. It can be used in the perception, navigation, manipulation and teleoperation and it is fully supported by ROS (Robot Operating System). Figure 3 shows UGV views through different angles. The choice of the UGV instance (UGVi) is motivated by its popularity, modularity and availability. Its hull design is supposed to host a load plate, a radar, a camera, a radio command. All mountable equipment components off-the-shelf COTS are not equipped in our system, because the mule concept scenario does not include all features available. We do not exploit all possibilities.



Figure 3 Different views of the UGV instance

UGVi is a satisfying example for mule concept because the datasheet in Table 1 offers a 50kg payload UGV. Our integrated human-augmented future battle field required mules that would follow humans to help them carrying loads. It is also popular and has been the subject of publications for years in specific fields such as SLAM integration techniques, UGV positioning and vision/static and dynamic exploration, mixing optimization and further needs for UGVs, UGVi manipulations, specific objectives that can be stated for the UGVi in a software point of view. But the state of the art does not present multi physical propositions for UGVi usage modeling, nor its V&V process while being included in a wider system of system. Yet mono-disciplinary optimizations on the UGVi can be questionable regarding the expected multi-disciplinary use of the robot.

These aspects could be treated using the methodology depicted in Figure 2, giving multidisciplinary dimensions to cited applications of MBSE. Theoretically, the more precise and reliable COTS specifications are, the more chances system of system capabilities can be realized in acceptable costs and delays. The replacement of datasheet with specifications, all the more augmented by MBSE and optimization models to elicit the link between specifications and project costs and delays would lead to system of system simulation for early validation. This article depicts a practical experimentation to observe that theory in practice: if the datasheet proves to be incomplete and/or inconsistent, system of system modeling and simulation will not be available before buying 20 k€ robots and spending hours of time to describe its behavior.

### III. Datasheet presentation

#### a. Datasheet as requirements

Table 1 System specifications from datasheet

1	Dimensions	990 mm length	39 in length
		670 mm width	26.4 in width
		390 mm height	14.6 in height
	Track	555 mm	21.9 in
2	Wheelbase	512 mm	20.2 in
3	Weight	50 kg	110 lbs
4	<b>Maximum Payload<sup>1</sup></b>	<b>75 kg</b>	<b>165 lbs</b>
	<b>All-terrain payload<sup>2</sup></b>	<b>20 kg</b>	<b>44 lbs</b>
5	<b>Speed (max)</b>	<b>1.0 m/s</b>	<b>3.3 ft/s</b>
6	Ground clearance	130mm	5 in
7	<b>Climb grade</b>	<b>45°</b>	<b>100% slope</b>
	<b>Traversal grade</b>	<b>30°</b>	<b>58% slope</b>
8	Operating Ambient Temperature	-10 to 30° C	14 to 86° F
9	<b>Operating time</b>	<b>3 hours typical</b> <b>8 hours standby (no motion)</b>	
10	Battery	24V 20Ah Sealed Lead Acid	
11	Battery charger	Short-circuit, over-current, over-voltage and reverse voltage protection	
12	Charge time	10 hours	
13	User Power	5V / 12V / 24V	
		Each fused at 5A	
14	Communication	RS-232	
		115200 Baud	



15	Wheel Encoders	78,000 ticks/m
16	Internal Sensing	Battery status Wheel odometry Motor currents

For the experimentation, we have chosen to exploit only four of the presented specification: numbers 4, 5, 7 and 9. It is an experimentation field that we figured sufficient to address optimization quality model problem on practical view, because it presents contradictory requirements that induce concept dimensioning choices, while being easy to measure and to confirm. The battery autonomy will be highlighted as a substantial multi physical compromise. If the Operating time defined in #9 is typically 3 hours, in which conditions can it be expected to reach this performance? If the laws of physics apply, the more load the mule is carrying, the more energy will be consumed. What is “typical” referring to? The experimentations in place all refer to the battery extinction.

For example, it is mentioned that the operating time was about 3 hours, but it was not explained on which type of ground, the supplied energy, with how much Payload and with which speed, and if it were compatible with slope climbing. Furthermore, if the UGVi was optimized for requirement #9, we imagine it would be so in all cases explained in requirement #4, supposedly on all grounds with maximum payload.

On the other hand, it is expected not to outreach the expected performance, otherwise it would mean that the system is overqualified for some features. A perfect match between datasheet and real performance is at stake.

This selection of focus on four requirement exploitation, however arbitrary, covers all questions raised in the theoretical problem (MTS-EP 1). It addresses multi physical capabilities of the system with respect to its expected usage.

### ***b. Maintaining the integrity of the specifications***

The Mule concept is designed to follow human pilot, while carrying his supplies for 3h at man walking speed (5km/h). We have observed previously that connection between the remote control and the robot starts to fail when the battery capacity lowers. If the tension falls, the electronical devices that controls the connection will observe a disconnection. The witness of the robot indicating the battery levels do not indicate the voltage threshold that makes the communication fail. In order to measure the decaying power, we plugged a voltmeter to our experiment, which we found not intrusive.

Our system specifications at glance are the following:

ID	Title	Requirement (mission)	Test method
001	Maximum Payload various ground	The system shall carry 50kg load following an 3h mission on campus.	Test pass if the battery runs out in more than 3h with load = 50kg on various grounds (tarmac and dry grass)
002	All-terrain payload dry grass	The system shall carry 20kg load following a 3h mission on dry grass.	Test pass if the battery runs out in more than 3h on grass with load = 20kg



003	All-terrain payload gravel soil	The system shall carry 20kg load following a 3h mission on gravel soil.	Test pass if the battery runs out in more than 3h on gravel soil with load = 20kg
004	All-terrain payload tarmac	The system shall carry 20kg load following a 3h mission on tarmac.	Test pass if the battery runs out in more than 3h on tarmac with load = 20kg
005	Speed (max) on flat tarmac	The system shall maintain a 1m/s speed when following the mission on tarmac.	Test pass if the speed is measured to be $\geq 1$ m/s with 20kg Load on average on 10 m
006	Climb grade on various grounds	The system shall climb 45° slope carrying 50kg load when following the mission in the campus.	Test pass if the robot climbs 5.5° and 25° on tarmac, and if it climbs 18.5°, 22°, 24° with load = 50kg
007	Climb test on dry grass	The system shall climb 45° slope carrying 20kg load when following mission in the campus.	Test pass if the robot climbs 35° slope with load = 20 kg on dry grass.

Note that our use case scenario was executed at the maximum speed the robot could reach, which is 3.6 km/h. It is far from our objective of 5 km/h.

#### IV. Tests and benchmarking

We were inspired by the frameworks for vehicle tests: vibration, slope Military standard procedures with adapted requirements. We have focused on battery level measurement to follow, inspired by (Dogru and Marquez 2018). In this section, we will test the autonomy and energy specifications, the speed specification, as well as the climb capacities.

##### *a. Measurements methods and means*

In order to carry out these tests, we used some measuring instrument:

- Distance ruler precision  $\pm 0.5$ mm.
- iPhone telephone timer to calculate the duration  $\pm 0:05$ s
- Voltmeter at the entry of the battery precision  $\pm 0:05$ V

##### *b. Energy tests*

###### **i. TEST 001: Energy test, 50kg load on various ground**

The environmental conditions, the nature of the soil and the payload have the biggest impact on the robot mobility. This has motivated us to test the 3 hours battery specification of requirement n°9 in Table 1 with 50kg payload and various ground nature as “best effort”.

- Test stop condition: Battery running out.
- Environmental conditions presented in table II

Temperature	17°	
Humidity	52%	
Precipitation	0%	
Wind	23 km/h	
Weather	Cloudy weather	
Load	50 kg	
Powered components	Lidar, Gps, calculator, Wifi, camera	
Flat Ground	Type of soil	Duration
	Tarmac	1h15
	Wet grass	1h

**Observations: test fail**

- 1) After 1h30 of operation, we started to lose the communication with the robot (about 12 times in 30 min)
- 2) After 2h15 of operation, we totally lost communication. The autonomy requirement has not been met in our conditions.

Table 2 Test log on various grounds, 50kg load

Since our first test was unsuccessful, we planned to be more specific with the ground nature and load, and to report the behavior of the power system. The system will be loaded with 20kg in the next tests. The ambiguity of table 1 requirements will be resolved with our next tests.

**ii. TEST 002: Energy test, 20kg load on grass**

- test stop condition: battery running out
- Test results in table 3

	Starting time	Ending time
Time	13h45	16h30
Temperature	26°	27°
Humidity	42%	39%
Precipitation	0	0
Wind	7.2 km/h	7.2 km/h
Weather	Sunny weather	Sunny weather
Voltage	27.55V	24.05V
Load	20 kg	
Powered components	Lidar, Gps, calculator, Wifi, camera	
Flat Ground	Type of soil	Duration
	Dry grass	2h45



Figure 4 Dry grass test environment

Table 3 Test log energy on grass, 20kg load

**Observations: test fail**

- 1) After 1h55 of operation, we started to lose communication with the robot (about 8 times in 50 min), we measured the voltage at each communication loss, and we presented the values in the following table 4:
- 2) After 2h45 of operation, we have totally lost the communication. The 3 hours have not been reached.
- 3) We noticed that the communication with the robot depends on the Wi-Fi, but the minimum voltage and current required for normal operation of the Wi-Fi are 24VDC and 0.3A, this can explain the total loss of communication in this test and which occurred when the voltage was 24.05V.

Table 4 Voltage measurements test grass, 20kg

Time	Voltage
15h40	24.53V
15h57	24.45V
16h09	24.27V
16h16	24.16V
16h20	24.12V
16h24	24.09V
16h27	24.05V
16h30	24.05V

4) The site chosen for this experiment is a dry grass, which is relatively flat, without tall grass, and therefore the robot can move forward without wasting much energy to submerge the grass, and there is not up or down process during the operation.

5) Compared to test b.i., the battery autonomy was extended by 30 minutes in similar conditions. This is a clue to the impact of the load on the system autonomy

6) from 16h20, i.e., in the last ten minutes of the experiment, even if the robot is controlled to go straight ahead, it cannot drive straight ahead (when controlled to go straight ahead, its trajectory is affected

**iii. TEST 004: Energy test, 20kg load on tarmac**

- Test stop condition: Battery running out or loss of communication.
- Test results in table 5



Figure 5 UGVi on tarmac

	Starting time	Ending time
Time	14h20	17h55
Temperature	26°	27°
Humidity	45%	46%
Precipitation	0	0
Wind	10.8 km/h	10.8 km/h
Weather	Sunny weather	Sunny weather
Voltage	27.47V	23.79V
Load	20 kg	
Powered components	Lidar, Gps, calculator, Wifi, camera	
Flat Ground	Type of soil	Duration
	Tarmac	3h35

Table 5 Test log energy on tarmac, 20kg load

**Observations: test successful**

1) Compared to test ii, the battery life was extended by another 50 minutes this time. This could indicate an influence of the ground nature on the battery autonomy.

2) After 3h35 of operation, we totally lost communication, but before that, we did not lose it. However, during the last almost 20 minutes of the experiment, the response speed of the robot to various operations degraded, and the rotation speed was significantly reduced.

3) We chose a flat tarmac. Compared to grass test b.ii., the tarmac has fewer gullies, so the robot has fewer bumps when moving forward. As a result, the power consumption was slower, and the communication system had less “chattering” because the direct current was not drawn to make punctual efforts to pass gullies.

**iv. TEST 003: Energy tests, 20kg load on gravel soil**

- Test stop condition: Battery running out or loss of communication.
- Test results in table 6:

	Starting time	Ending time
Time	14h20	17h10
Temperature	26°	27°
Humidity	45%	45%
Precipitation	0	0

Table 6 Test log energy on grass, 20kg load

Wind	10.8 km/h	10.8 km/h
Weather	Sunny weather	Sunny weather
Voltage	27.47V	23.92V
Load	20 kg	
Powered components	Lidar, Gps, calculator, Wifi, camera	
Flat Ground	Type of soil	Duration
	Dry soil with gravel	2h50



Figure 6 Gravel soil illustration

### Observations: test fail

- 1) After 1h45 of operation, we started to lose the communication with the robot (about 7 times in 45 min), we measured the voltage at each communication loss, and we presented the values in the following table.

Time	Voltage
16h35	24.36V
16h49	24.28V
16h58	24.15V
17h03	24.06V
17h06	23.99V
17h08	23.95V
17h10	23.92V

- 2) After 2h40 of operation, we have totally lost the communication.

3) There is a lot of coarse gravel on the ground, which can increase the friction when the robot moves, and make the progress of the robot very bumpy, where the bumps are much bigger than on tarmac and on dry grass.

3) Due to these gravels, the rotation speed of the robot decreased significantly, the rotation became very difficult and bumpy.

4) Although the experiment lasted 2 hours and 50 minutes in total, at the beginning it took us about 40 minutes to bring the robot to the test site, which is the ground with gravel. During these 40 minutes, the robot was moving on tarmac. In comparison with test b.iii, we assume that 40 minutes on tarmac could only improve the performance on our test on gravel.

- 5) This test should be run again in order to know if the performance on gravel is better than grass or not. But the fact that the test was still unsuccessful even with 40 minutes on tarmac and extra care on that day proves our point for this article: the requirement on “3h typical” was not met in those conditions.

Table 7 Voltage during test b.iv.

### c. TEST 005: Load and speed tests

The maximum speed of the robot supposed to be equal to 1 m/s (as indicated on the Datasheet), So we did a speed test with 20kg load on Tarmac to put to test requirement n°5 from table 1.

Table 8 Test log combining load and speed on tarmac

Temperature	25°C
Humidity	42%
Precipitation	0%
Wind	11 km/h
Weather	Sunny weather
Load	20 kg
Flat ground	Tarmac

### Observations: test fail

- 1) The robot covered 9.57m in 9.90s (on average) (for 5 trials: 10.00s, 9.98s, 9.78s, 9.71s, 10.02s) on tarmac. speed= 0.967 m/s.
- 2) The robot covered 17.94m in 18.47s (on average) (for 5 trials: 18.55s, 18.45s, 18.26s, 18.62s, 18.46s) on tarmac. speed= 0.971 m/s.
- 3) We used the lines perpendicular to the edge of the road as the starting line and the ending point, but they are not completely vertical, so the robot was tilted when it moves forward, so the actual walking route is greater than the distance between the lines that we

measured. Therefore, the measured speed is less than the actual speed at that time. If the robot tilts as it moves forward, the further it moves, the larger the distance measurement error and the larger the speed measurement error. At the end of the experiment, we found that if the side of the robot is aligned with the edge of the road at the beginning, the robot can better avoid tilting.

For accuracy, we have +/-0.5 mm at 1 m for the tape measure that was used to measure the distance and the error of the iPhone timer was +/-0.05s. However, we believe that the main source of error in the timing of the experiment is human reaction time. The coordination of the start timing, the reaction time when crossing the finish line, these errors will be greater than the instrument error and will depend on the timekeeper. This introduces a bias in the validation team that would motivate automatic simulation in later testing process.

#### d. TEST 006: Climbing tests

The maximum climb grade of the robot supposed to be 45° as indicated on the requirement n°7, so we did a climb test on wet soil, with minimal payload. Figure 7 represents the conditions:



Figure 7 UGVi climbing test 35° slope on wet grass

Temperature	12°	
Humidity	72%	
Precipitation	10%	
Wind	24 km/h	
Weather	Cloudy weather	
Load	20 kg	
Powered components	Lidar, Gps, calculator, Wifi, camera	
Sloping Ground	Type of soil	Slope angle
	Wet soil	34.5°

#### Observations: test fail

1) We tested the robot on a 35° slope on wet soil: the robot failed to climb it. It climbed about one meter up the slope then it started to diverge to the right and started to tumble down the slope. as seen in figure 7 The requirement was not met in our conditions.

Table 9 Test log climbing 34.5° slope, 20kg load

#### e. TEST 007: Stress tests

Temperature	12°	
Humidity	72%	
Precipitation	10%	
Wind	24 km/h	
Weather	Cloudy weather	
Load	50 kg	
Sloping Ground	Type of soil	Slope angle
	Tarmac	5.5°, 25°
	Wet grass	18.5°, 22°, 24°

The robot is supposed to climb 30° slopes with 20kg load, so we did a stress test with 50kg load on different slopes. This exceeds the all-terrain recommended payload in requirement n°4. Figure 8 represent the Tarmac test conditions; and figure 8 to 12 represent the wet grass tests conditions.

Table 10 Test log stress slopes, 50kg load



Figure 8 UGVi on tarmac 5.5° slope



Figure 9 UGVi on tarmac 25° slope





Figure 10 UGVi on grass  
22° slope



Figure 11 UGVi on grass  
18.5° slope



Figure 12 UGVi on grass  
24° slope

### Observations: test pass

- 1) We tested the robot on two slopes less than 20 °, the first one is about 5.5° on tarmac and the other one is about 18.5 ° on wet grass:
- 2) We tested the robot on three slopes less than 30°, two slopes on wet grass of 22 ° and 24 °, and a 25 ° slope on tarmac.

This indicates that on some requirements, the UGVi performs better than expected. This indicates that for this purpose only, the specification is not efficient, because these performances were never clearly specified.

## V. Discussion

The V&V on our UGVi demonstrated that the datasheet furnished no insight/hint to system usage, and that the UGVi system was not fit for our Mule capability. Moreover, since almost no datasheet performance was met, we could not determine whether the system is intended for our purpose or not. At this point, we would have preferred to read the initial requirements, which are meant to express the system functions. Last, if the system were picked from a set of non-dominated optima solution to (MTS-EP), there would be a perfect match between the original intended capability of the system and its performance. The following actions could be set: lower the buyer's expectations, which could lead to the end of business, improve the datasheet, and improve the system performance toward some chosen capabilities.

We have introduced a loop that ensures the increase of company knowledge, and at the same times links directly the system requirements to its intended use cases and user documentation. Failing all tests on UGVi performance gives enough reasons to invest in a method that would benefit both user manual descriptions and correlation between system and its intended capabilities. The inclusion of MBSE in the process of capability description would decrease ambiguities on the intentions of the builder in the requirements and in the user manual. Our process can also include physical models at the highest level of conception. The complete optimized System-driven engineering architecture conception flow will definitely include some re-engineering to transform the UGVi into an optimized mule system. This involves re writing the system specifications and re-designing it in the first step, and then propose a re-design in those objectives that would fit the objectives better. The solutions could be explored with simulation rather than real-life V&V, in order to reduce the costs of re-design. The test logs could be smartly combined in order to improve the next specifications for future product developments.

We have considered “mule concept” as the systems expected capabilities. There were no declared capabilities on the UGVi design details. There is no proof that the UGVi was designed as a “mule concept” more than other purpose. The concept architecture capabilities have never been exposed by the builder. This could be improved with more visibility on the system capabilities by the builder,

using for instance the operational views in domain specific architecture frameworks and/or detailed representations of the system instance in model-based design or MBSE.

The SSFP design flow described is a proposition that matches requirement improvements, V&V practice and user manual redaction challenges. It introduces a method that enhances the quality of the system V&V context, shares the best-known intended performances of the system, and generates automatically the user manual datasheet. Furthermore, the SSFP flow introduces a re-engineering method that would continuously improve engineering specifications while mastering products engineering costs and delays over the concept and maintenance phase of the system lifecycle. This could smoothly replace the “best effort” system engineering by a promising optimized system driven engineering. Further research shall focus on quantification of the gains using such a method UGV-category product development.

The correlation between the objective expression will be subject of other publications, and will bring insight to the complete GOAL equation rather than only the “MTS-EP”. If our proposition could help enhancing Quality of the system, the outcome on resource and time of design are considered “best effort”. Further publications will fill the other goals of (GOAL) introduced in the state of the art.

For defense system definition, robots on battlefield will follow the military standards given by ISO and STANAG, which will advise contract managers on how V&V for system behavior on battlefield will be considered (Michelson 2021). Therefore, the COTS which cannot prove to be compliant with the standards will be excluded from defense contracts because it will be impossible to integrate them in the larger system of system capability forecast.

## VI. Conclusions

Datasheet are not adequate to reflect the properties of a system, but can be replaced by system engineering specifications which improve accuracy of subsystem behavior description. The lack of completeness and ambiguity carried out by the very structure of a usual datasheet ends up with cumulative ambiguity and over-testing with no guarantee of results for end systems. Should datasheet be banned in favor of systematic requirement engineering files or documents requires that all suppliers of components and subsystems are trained and master system engineering process.

On the opposite, datasheet easily reveal the lack of knowledge in system engineering and requirement engineering. Therefore, companies have three options: (a) stick to the usual “reliable” supplier with no investment in writing the “should-be” requirements and avoid evolving equipment (b) stick to the usual “reliable” supplier with consistent investment in writing the “should-be” requirements and pay for the component evolutions (c) impose systems engineering exchange protocols. Some industries such as the semiconductor and the embedded system industry are not familiar with system engineering and make extensive use of component and subsystem datasheet. If the aim is to lay out seamless system engineering flow down to physical architecture and up to V&V, there is a major overhaul to be conducted.

In our multi-physical UGV<sub>i</sub> example, system tests failed: technical validation was not reached. If the system was sold to a client, none of the claimed capabilities could be covered as proved in our benchmarking. The only objective that was reached was not even specified, therefore would have generated no outcome. The experiment presented here illustrates the necessity of introducing environmental conditions and objectives to technical specification. It also indicates that the original requirement artifacts for the system would do a better job at describing the expected performance of the system, especially if it is enhanced by architecture frameworks and MBSE concept design, which are the trends to seek completeness and unambiguous system description.



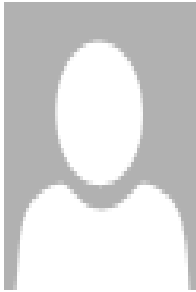
## References

- AFNOR. 2015. *ISO/CEI/IEEE 15288:2015 Systems and Software Engineering — System Life Cycle Processes*. AFNOR. [www.afnor.org](http://www.afnor.org).
- Bachelor, Gray, Eugenio Brusa, Davide Ferretto, and Andreas Mitschke. 2020. “Model-Based Design of Complex Aeronautical Systems Through Digital Twin and Thread Concepts.” *IEEE Systems Journal* 14 (2): 1568–79. <https://doi.org/10.1109/JSYST.2019.2925627>.
- Bonnet, Stephane, Jean-Luc Voirin, and Juan Navas. 2019. “Augmenting Requirements with Models to Improve the Articulation between System Engineering Levels and Optimize V&V Practices.” *INCOSE International Symposium* 29 (1): 1018–33. <https://doi.org/10.1002/j.2334-5837.2019.00650.x>.
- Dogru, Sedat, and Lino Marquez. 2018. “A Physics-Based Power Model for Skid-Steered Wheeled Mobile Robots.” *IEEE Transactions on Robotics* 34 (2): 421–33. <https://doi.org/10.1109/TRO.2017.2778278>.
- Duffy, James B, Robert Combs, Jingyao Feng, and James P Richardson. 2021. “Return on Investment in Model-Based Systems Engineering Software Tools,” 15.
- Duprez, Jean, and Dominique Ernadote. 2020. “Towards a Semantic Approach of MBSE Frameworks Specification.” *INCOSE International Symposium* 30 (1): 1405–19. <https://doi.org/10.1002/j.2334-5837.2020.00794.x>.
- Hahn, Heidi Ann, Nick Lombardo, Ann Hodges, Mitchell Kerman, and Frédéric Autran. 2020. “Implementing Systems Engineering in Early Stage Research and Development (ESR&D) Engineering Projects.” *INCOSE International Symposium* 30 (1): 433–48. <https://doi.org/10.1002/j.2334-5837.2020.00732.x>.
- Henderson, Kaitlin, Virginia Tech, Perry St, and Durham Hall. n.d. “Is CAD A Good Paradigm for MBSE?,” 15.
- Huldt, T., and I. Stenius. 2019. “State-of-Practice Survey of Model-Based Systems Engineering.” *Systems Engineering* 22 (2): 134–45. <https://doi.org/10.1002/sys.21466>.
- INCOSE. 2015a. “A World in Motion - SE VISION 2025.”
- . 2015b. *INCOSE Systems Engineering Handbook 4e*. WILEY.
- Michelson, Brian. 2021. “Why NATO Needs Lethal Autonomous Weapon Standards.” <https://cepa.org/why-nato-needs-lethal-autonomous-weapon-standards/>.
- Morgan, Markeeva, Thomas Holzer, and Timothy Eveleigh. 2021. “Synergizing Model-based Systems Engineering, Modularity, and Software Container Concepts to Manage Obsolescence.” *Systems Engineering*, June, sys.21591. <https://doi.org/10.1002/sys.21591>.
- Pate, David J., Justin Gray, and Brian J. German. 2014. “A Graph Theoretic Approach to Problem Formulation for Multidisciplinary Design Analysis and Optimization.” *Structural and Multidisciplinary Optimization* 49 (5): 743–60. <https://doi.org/10.1007/s00158-013-1006-6>.
- Stevens, Jennifer Stenger. 2017. “Warranting System Validity Through a Holistic Validation Framework: A Research Agenda.” *INCOSE International Symposium* 27 (1): 654–71. <https://doi.org/10.1002/j.2334-5837.2017.00385.x>.

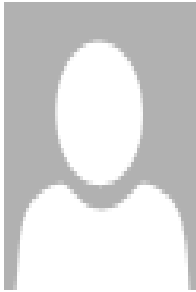
- Voirin, Jean-Luc, Olivier Constant, Eric Lépicier, and Frédéric Maraux. 2020. "Dream the Future: Systems Engineering in 2030." *INCOSE International Symposium* 30 (1). <https://doi.org/10.1002/j.2334-5837.2020.00754.x>.
- Wach, Paul, and Alejandro Salado. 2020. "The Need for Semantic Extension of SysML to Model the Problem Space." In , 9.
- Watson, Michael, Azad Madni, Bryan Mesmer, and Dorothy McKinney. 2020. "Envisioning Future Systems Engineering Principles Through a Transdisciplinary Lens." *INCOSE International Symposium* 30 (1): 1517–32. <https://doi.org/10.1002/j.2334-5837.2020.00801.x>.
- Wheaton, Marilee J., and Azad M. Madni. 2018. "Model Based Tradeoffs for Affordable Resilient Systems." *INCOSE International Symposium* 28 (1): 30–39. <https://doi.org/10.1002/j.2334-5837.2018.00465.x>.
- Wolny, Sabine, Alexandra Mazak, Christine Carpella, Verena Geist, and Manuel Wimmer. 2020. "Thirteen Years of SysML: A Systematic Mapping Study." *Software and Systems Modeling* 19 (1): 111–69. <https://doi.org/10.1007/s10270-019-00735-y>.
- Wu, Quentin, David Gouyon, Sophie Boudau, and Éric Levrat. 2019. "Towards a Maturity Assessment Scale for the Systems Engineering Assets Valorization to Facilitate Model-Based Systems Engineering Adoption." *INSIGHT* 22 (4): 37–39. <https://doi.org/10.1002/inst.12274>.
- Xu, Peng, and Alejandro Salado. 2019. "A Concept for Set-based Design of Verification Strategies." *INCOSE International Symposium* 29 (1): 356–70. <https://doi.org/10.1002/j.2334-5837.2019.00608.x>.
- Younse, Paulo J., Jessica E. Cameron, and Thomas H. Bradley. 2021. "Comparative Analysis of a Model-based Systems Engineering Approach to a Traditional Systems Engineering Approach for Architecting a Robotic Space System through Knowledge Categorization." *Systems Engineering* 24 (3): 177–99. <https://doi.org/10.1002/sys.21573>.

## Biography

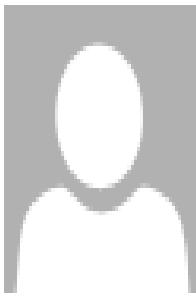
*For the initial paper submission, do not include this section. For the final paper submission, provide a short biography (50 to 100 words) of each author here and delete these instructions. A small headshot may be included with each biography. The photo's resolution should be at least 300 dpi. The photo size should be approximately 1" x 1.5" or 25 mm x 38 mm.*



**Author Name.** Provide a short biography of the author. Provide a short biography of the author. Provide a short biography of the author.



**Author Name.** Provide a short biography of the author. Provide a short biography of the author.



**Author Name.** Provide a short biography of the author.