



HAL
open science

Modeling swarm mission with COTS characterization: a series of return on experience

Lorraine Brisacier-Porchon, Omar Hammami

► To cite this version:

Lorraine Brisacier-Porchon, Omar Hammami. Modeling swarm mission with COTS characterization: a series of return on experience. INCOSE Annual International Symposium, INCOSE, Jul 2024, Dublin, Ireland. hal-04584817

HAL Id: hal-04584817

<https://hal.science/hal-04584817>

Submitted on 23 May 2024

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.

Modeling swarm mission with COTS characterization: a series of return on experience

Lorraine Brisacier-Porchon
KNDS France
11, allée des marronniers,
78022 Versailles Satory, France
Brisacier.lorraine@gmail.com

Omar Hammami
U2IS, ENSTA Paris, Institut
Polytechnique de Paris
826, boulevard des Maréchaux
91120 Palaiseau, France
omar.hammami@ensta-paris.fr

Copyright © 2024 by Lorraine Brisacier-Porchon and Omar Hammami. Permission granted to INCOSE to publish and use.

Abstract. System design in defense systems is a competitive field, in which economical viability relies on a sequence of architectural decisions, aiming at quality, resource and time (Q,R,T) compromises. We observe that low-cost unmanned ground vehicles (UGV) and drones appear as new threats on current battlefield. To face these new threats, Direction Générale de l'Armement (DGA) have organized challenges around robotization of battlefield, to design future employment doctrines and help technologies to reach maturity in a reasonable time. This article exposes a NATO Architecture Framework (NAF) 3.1-based workflow that includes return of experience from the field over yearly iterations of such challenges. The capabilities depicted are requirements to match, constituent systems are based on Components-Off-The-Shelf (COTS) answering to both edition of the challenge. This article details how manually re-injecting feedback from field back to the system model failed to ensure the next iterations of the challenge. Our works propose conclusions on formulation of the “engineering leakage problem” and how resolution of this problem is NP-Hard and should be addressed using optimization.

Keywords. Architecture Frameworks, MBSE, COTS, return on experience.

Introduction

Recent events in Ukraine have shown the necessity to develop low-cost constituent systems that can work in high-tech collaborative framework. A particular focus is made on design and usage of drones and unmanned ground vehicles (UGV) that can create major destruction over million-euro main battle tanks. In order to create efficient system-of-systems with respect to specific list of capabilities, defense and aerospace systems builders are currently developing digital engineering methods. The development of workflows adapted to each line of products is an investment in itself which enterprises want to use most efficiently so that return on investment estimates guarantees economic success as soon as possible. In the defense and aerospace context, planification of system lifecycle commonly reach 40 to 50 years lifecycle, from early sketch to dismantlement. However, major defense industries often participate to robotic challenges to show their innovative technologies. It is also the place for DGA of those countries to evaluate doctrinal concepts on real-size examples in combat conditions, and to encourage technology development inclusion in existing collaborative combat configurations.

To the best of our knowledge, system engineering practice is not usual practice in such robotic challenges organizations.

If Model Based System Engineering (MBSE) practice has been a major trend in current publications since 2013 (Wolny et al., 2020), practitioners still face emerging resistance to its adoption (Henderson et al., 2023). While ISO 15288 version 2015 and 2023 provide industries with definitions and process to organize technical system development, most companies adapt the process to comply with their own activities. However, “MBSE does not change the practice of Systems Engineering as defined in the

INCOSE SE Handbook” (Papke et al., 2020). We selected the NATO Architecture Framework (NAF) version 3.1 (*NAF:NAF Subviews*, 2010) because of its ready-to-use defense system ontology with costs and delays representations. We consider that the framework, implemented in CATIA Magic (Dassault Systemes, 2022). We use it without modification of the meta-model. It presents 49 views that share artifacts, which makes the meta-model of NAF models composed of interconnected nodes. Figure 1 offers the constellation formed of all views. In this complete illustration of possibilities, a directed connection between two nodes means we want to convey an artefact from a starting node to another.

To the best of our knowledge, presenting military robotic challenges using NAF views is uncommon in state-of-the-art in SE, and unknown in the robotic field. Yet there is a need to specify drones and UGV collaborations capabilities with formally designed constraints in scheduling, costs and quality.

Hence this article provides a digitally continuous workflow based on NAF 3.1 views used to represent capability increments, COTS expected performances and learn from real-scale fielded data, and the effects observed on the field over two iterations. The workflow is modeled as a path running from a specific view to another without human intervention. We expose in this article how our workflow could be completed over two iterations, and what to expect for future increments of the challenge.

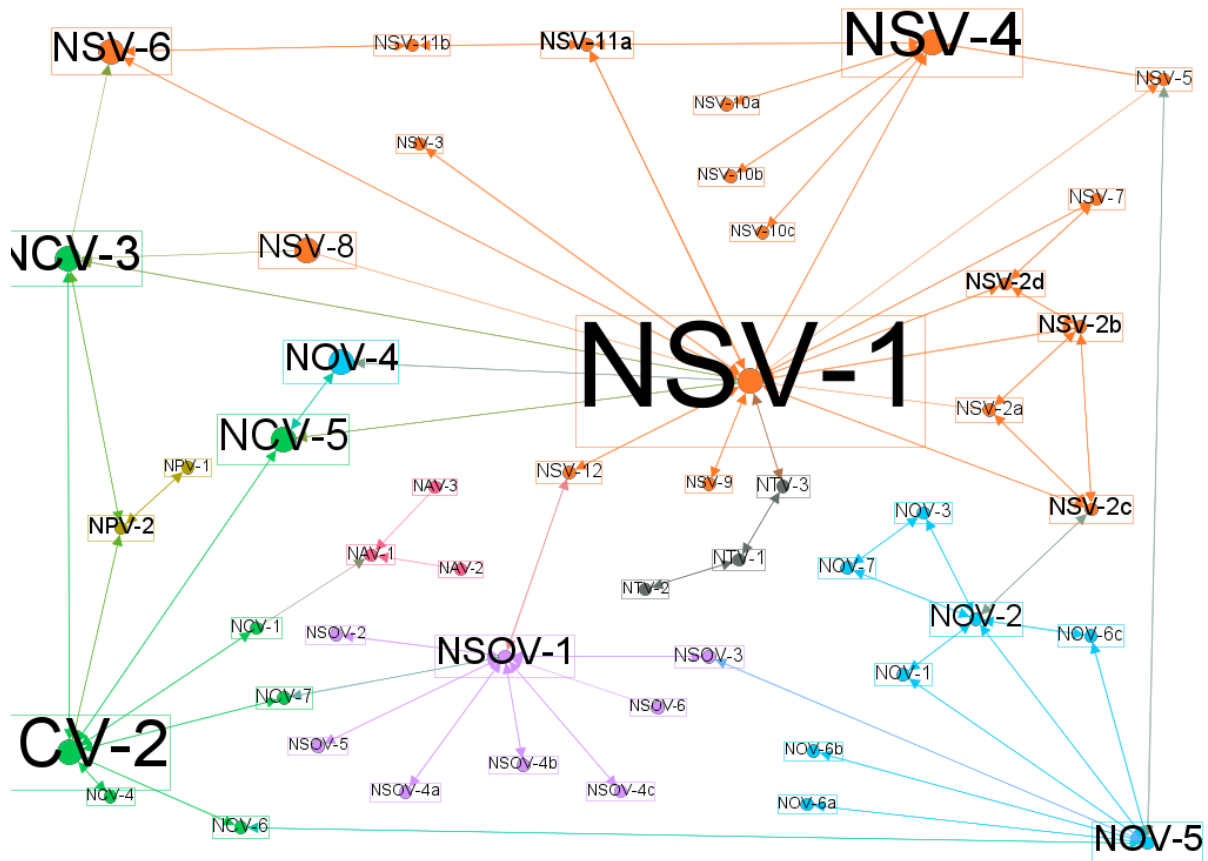


Figure 1: Meta-model of NAF 3.1: a constellation of 49 nodes, colored with different viewpoints

State-of-the-art views selection, enterprise vision, verification considerations

Furthermore, state-of-the-art in Architecture Framework (AF) shows that all views and viewpoints representations are not mandatory. It is the case in all published works using United AF (UAF), such as in (Hause et al., 2016; Rubio & Rigaut, 2023), as well as in NAF shown in (Janer & Proum, 2014). Actually, all publications on AF propose methods that involve a sub-selection of adapted views. Figure 2 presents our own selection to frame both mission from client point of view and system design from a research and development point of view. For the first two increments, no expense of time or budget was

spent in system engineering practice, either in text-based or modeling. The present views are all made after the second challenge to capitalize over the two participations.

The beginning of the path is highlighted in Figure 2 using green color. It holds enterprise vision of the challenge, with the context in which those challenges are proposed. It is freely inspired from robotic challenges trends organized by MoDs and DoDs to encourage technology development to face observed new threats.

We inserted in this view the objectives of challenge 1 is “to recognize” a limited zone, through a given set of “Lines of mark” (LIMA). The objectives of challenge increment 2 is “to capture a position or a zone”, in the same context. We consider that the second capabilities are more elaborate than the first.

The challenge increments are programmed to happen every year, and in this views, the mandatory meeting dates for each increment are defined, as well as the best hoped start of works.

Our own selection of views is based on the amount of information provided by the objectives and constraints given in all mission of the man-machine collaboration challenge. We also use our previous works with objectives to realize the missions a swarm of robots in one year using COTS. The systems that we selected were bought off the shelf and our lab realized all the software design in constrained schedule and resource. Since architecture views are meant to document, maintain and improve systems, we selected a primary set of 13 views illustrated in Figure 2, with the possibility to add new views on the go if found necessary. The cost of adding views is considered minimal, because we rely on a pre-established ontology and meta-model. The set of views are: NATO Capability views (NCV) 1,2,3,5, NATO Operational views (NOV) 2,4,5, NATO System views (NSV) 1, 2a, 4,5,6,8. We proceeded in a specific order to create views, suggesting that our method can be illustrated as paths between views as in Figure 2.

Our goal is to create a model-centered architecture database. We consider a view is necessary if it is able to store information so that the model becomes a dock base including fielded information. Indeed, our verification method holds in checking that fielded data can be included back to the model.

We will observe the effort to create new instances of models with provided tools and methods “as it is”, with no further investment in meta-modeling.

Capabilities input by challenge organizers

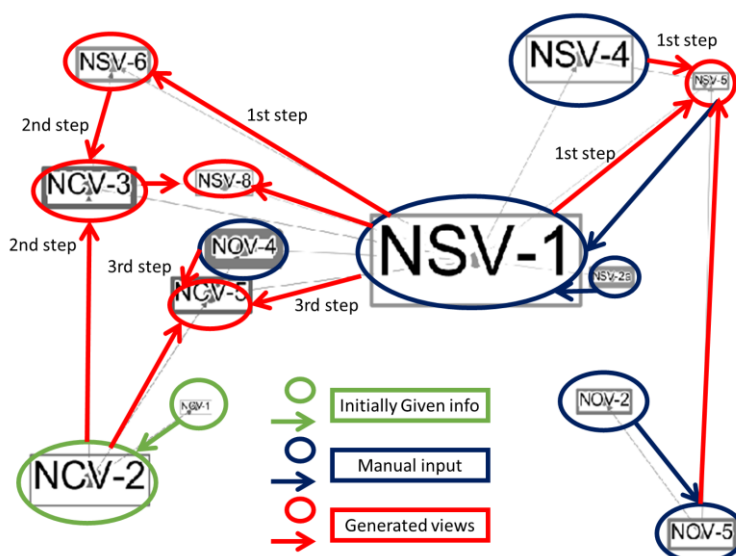


Figure 2: The initial method, a path through views

The challenge organizers gave preferably capabilities to achieve as instructions. In the first increment, challenge organizers specified “to recognize” composed of four capabilities that can be executed in sequence or simultaneously: detect and neutralize traps shown in Figure 3, escort a vehicle to a destination, guide the head vehicles through theater of operations, and communicate information for current mission. Those were the high-level capabilities that we chose to deliver, as instructed. The performance allowed was to execute capabilities in limited time and zone.

The second increment, “To capture a zone or a position”, is composed of “to neutralize enemy”, “to detect enemy”, “to drive out enemy” and “to capture enemy”. If mobility and communication issued in the first increment could be of use, those capabilities require better physical performance, and more complex platforms.

We conclude this subsection in mentioning that all capabilities could be drawn manually in NCV-2 and NOV views after both increments happened, some representations could not, such as the design of targets or environment.

Environment description by challenge organizers

Note that the system design capabilities are accompanied with constraints. To “capture” and ‘hold’ positions comes with enemy characterization in the form of 1 m³ red cubes shown in Figure 3. It also has environmental descriptions such as described in Figure 3 and Figure 4, such as theater of operations descriptions to associate to capability realizations:



Figure 3: example of a target characterized in the challenge

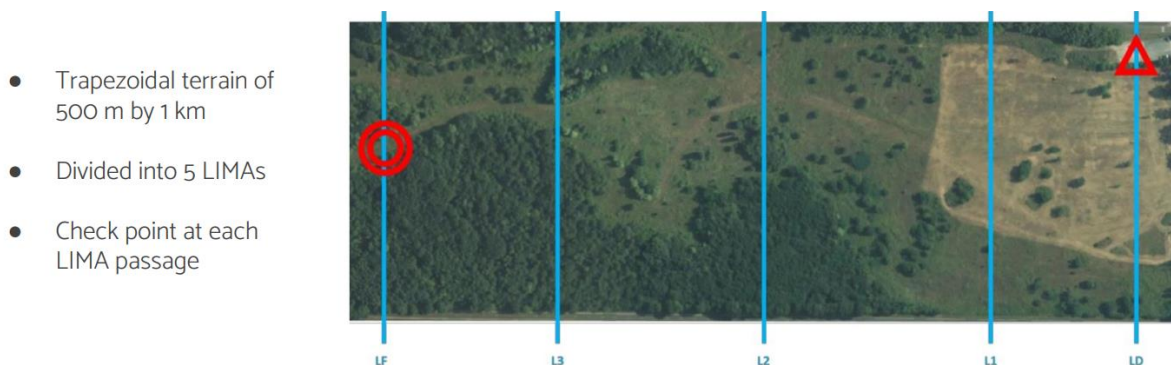


Figure 4: theater of operations to realize capabilities

We input in NOV-4 views constraints on robotic platforms pilots. On the first increment (2022), $N_{operators} \leq N_{platforms}$, and in the second edition (2023), the constraint was $N_{operators} < N_{platforms}$. In the first challenge, we could afford buying 2 drones and 1 UGV for the swarm mission. We entered the contest with 3 pilots. To this was added the chief of squad, to take decisions to realize the mission. All personnel were

to stay inside an armored vehicle called “Command and Control (C2) Vehicle”. Referred to as a VAB in Figure 5.

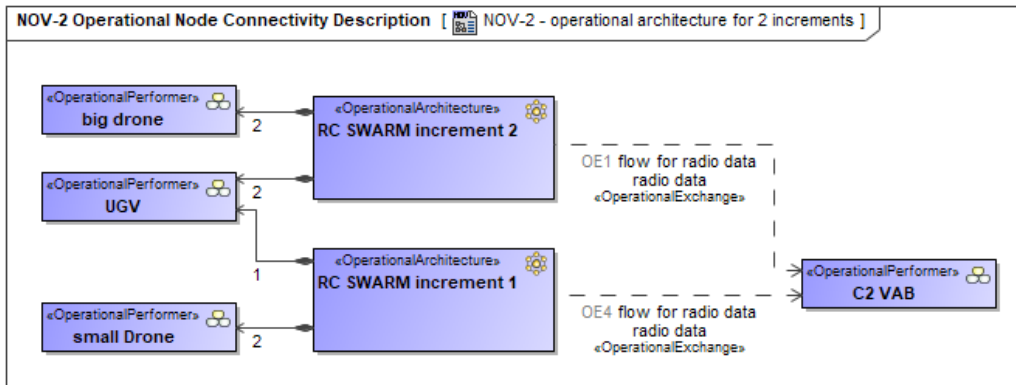


Figure 5: Operational architecture designed for our participation

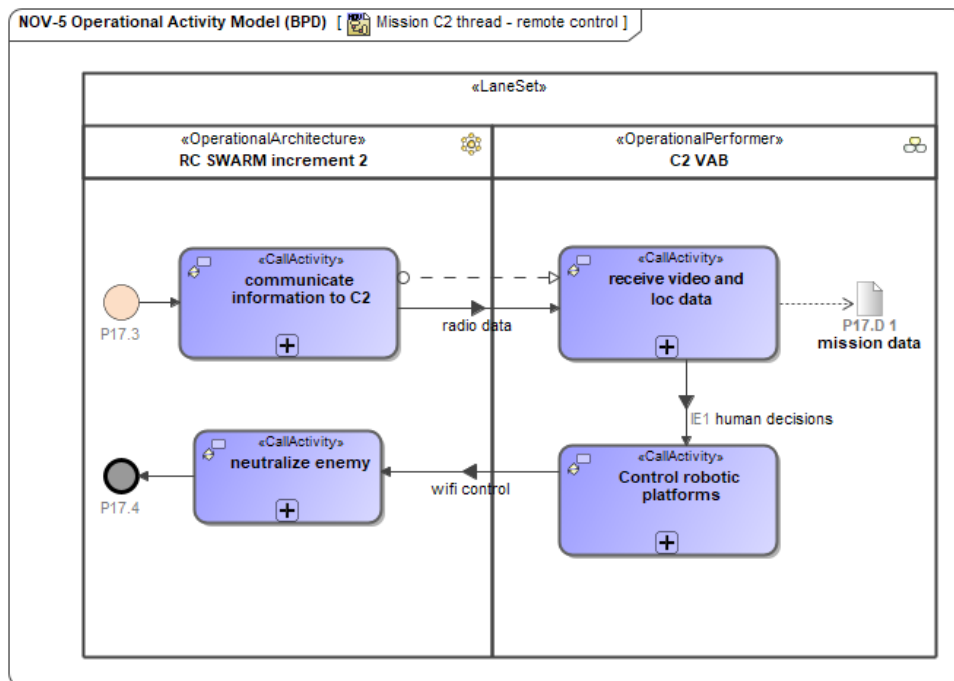


Figure 6: example of one detailed activity with remote control

Although the tool helps combining existing elements and validation of consistence within a view such as Figure 6, association of performers with subprocess has to be done by hand. In this step, we started adding some information that were not primarily issued from the challenge, but some interpretations of how capabilities could be realized.

Input of our own constraints and objectives

Figure 7 presents our given system platforms attributes, constraints and resource. Even though we already published works that contest datasheet format (Brisacier-Porchon & Hammami, 2022), we selected the supposed performance of each platform and inserted it into our system models vision, as shown for example purposes in Figure 7. Insertion of those performances was a tedious task. Indeed, if we consider building a view takes one week, any disruption between datasheet and actual performance of a resource can cause the whole week of re-work. Yet a disruption can be detected at any moment, including twice a week. In explaining this situation, we call for help building views at a higher frequency than problem detection. A model cannot be satisfying if it is obsolete.

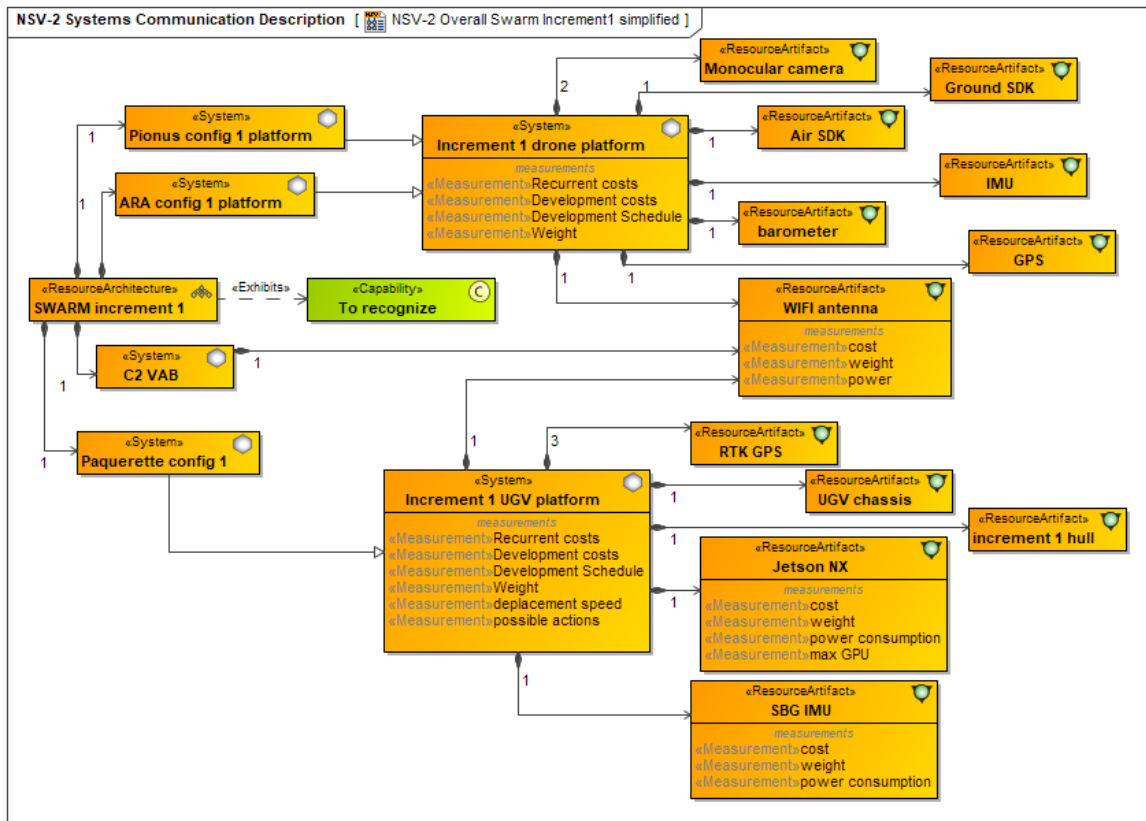


Figure 7: overall simplified view of the system architecture for increment 1 to realize the capability “to recognize” on the specified field

Note that building a top-down approach was not possible, because the laboratory already owned robots, and could not afford to switch within the time constraints. The task of adding manually all performances of all resource in all constituent systems knowing in advance that they will be modified depending on the context might seem daunting. We will refer to this problem as “engineering leakage problem” (ELP).

The observations made on building architecture Frameworks may as well be generalized to any system model in SysML in any version. These observations on ELP may add to emerging resistance to MBSE or SE as described in (Henderson et al., 2023) in organizations.

Focusing on scheduling views architecting

Furthermore, AF provides means to represent program management views in the form of Gantt projects. The challenge setup carries constraints on milestones – happening every year with document delivery strategy. Figure 8 illustrates one of the Gantt representations that are supposed to help program management to iterate. It holds the different milestones constrained by challenge hosts, as well as resources provision from our own estimates.

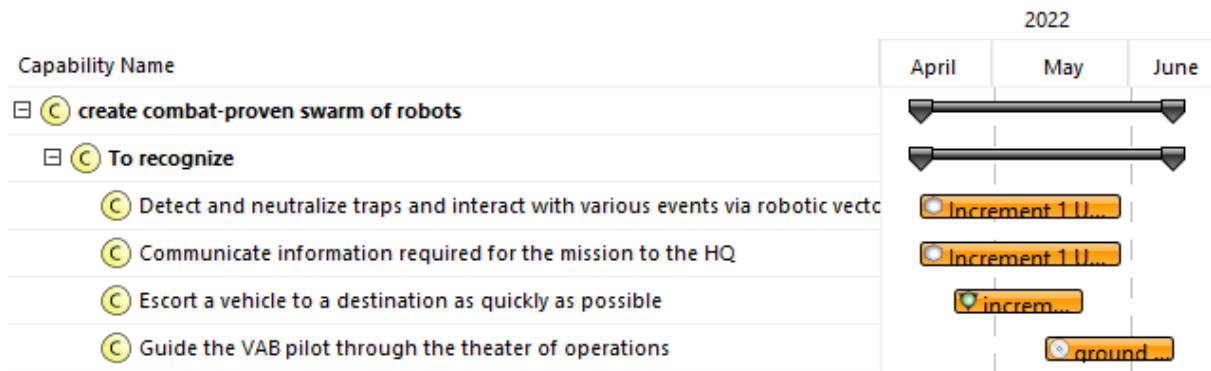


Figure 8: Capability increments for systems

However, at the end of the first challenge, we did not know what capabilities were going to be required for the next step. In our vision, it means that we had the constraints in the form of milestones without performance objectives as capabilities. Therefore, we did not risk methodology modifications and focused on increments based on previously known capabilities. In the same manner as with system views, building schedules with actual fielded information takes a certain amount of time to be conclusive for decisions, considering we can estimate how much time it takes to provide Gantt views. If major hypothesis changes more often than the time required to build views, we build representations that we know will be obsolete as soon as being published, which also feels daunting. We recognize here also an ELP pattern in building management views.

In building a retrospective for both increments, we observed that overall views, even generalized to any tool and method, require minimal architectural information to compute costs and schedule estimates for further increments.

This leads to a set of open questions, that can be asked in a context with or without systems engineering practice:

1. Was it possible in the first place to answer and succeed in the challenge with the given requirements with no systems engineering practice?
2. If I already possess some candidate constituent systems, is it better re-using it or start from scratch in terms of costs and delays?
3. Can we predict a failure to realize the missions?
4. What methods and means do we have to guarantee success of the objectives in a constrained time and budget?

Inclusion of fielded data in an AF model to lead to incremental capabilities

Preamble: environmental conditions in both events

Our strategy of incremental modifications from field observations failed to impact the capabilities performances. Indeed, our overall performance of swarms 1 and 2 were:

- On the first increment we reached the last-but-one “LIMA” line, neutralizing only 20 targets over 50 in one hour. The weather was rainy, and the different soil types were wet forest humus, tarmac, wet short grass, wet high grass. Outside temperature average 20°C

- On the second increment we reached the second “LIMA” line, neutralizing less targets than the first iterations. The weather was shiny, and the outside temperature was average 25°C. Soil was dry forest humus, tarmac, dry short grass and dry high grass.

The improvements based on experience return was founded on text and no system engineering was conducted until after the second increment.

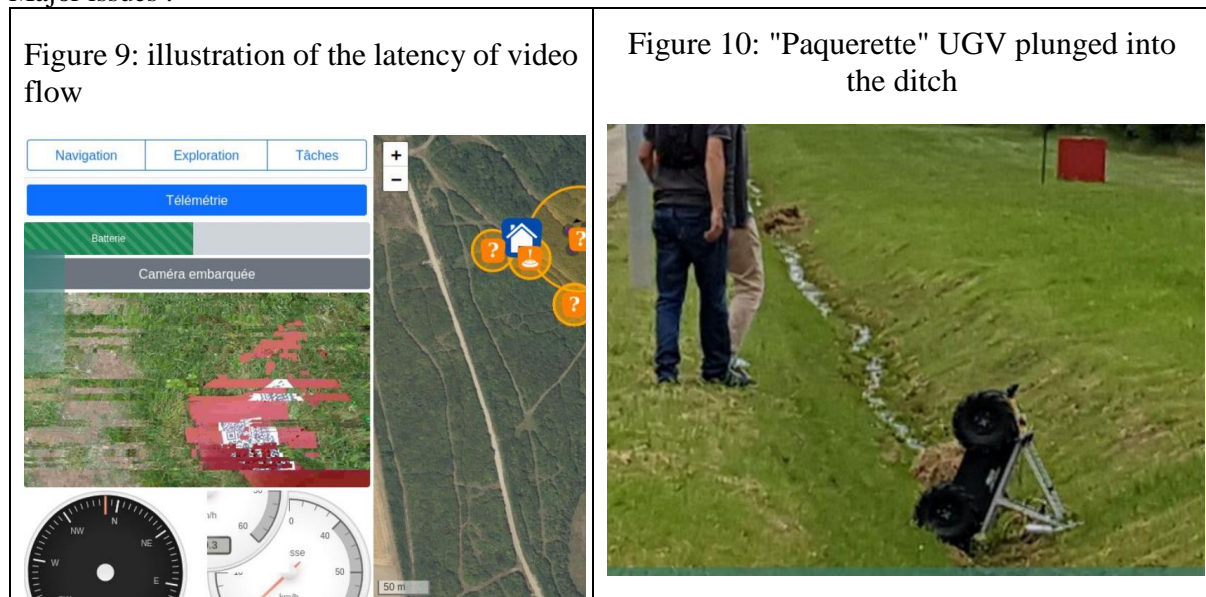
All enemy characterizations and zone descriptions were added as attributes and measurements by hand in our model. Another information stored in our capability attributes is actual hygrometry in percentage of humidity, actual soil type (mud, green grass, high grass, forest humus), actual elevation or slopes, as well as actual weather information.

Results and observations from increment 1

Observation 1 on mission completion: as illustrated in Figure 10, the “Paquerette” instance of UGV fell in the ditch by itself after 15 min of mission.

Context: video latency was too high and unpredictable on the field, as shown in Figure 9, pilot was late in commands. Paquerette was unable to drive itself out. LIDAR and hull were not efficient enough. Position was not precise enough. We believe that the latency of the video flow was caused by the technology choice (Wifi) and the antennae placement on UGV and on C2 VAB. Connection with the drones were easier.

Major issues :



Observation 2 on mission completion: The drones were too light. Their weight was not enough to “dis-arm” necessary targets.



Impacts on AF model made to illustrate decisions:

1. Modeling video flow and antennae performance, but no general assessment, no possibility to model or simulate the impact of the environment. We decided to improvise and move the antennae, create a different comm architecture to separate wifi commands and video flow.
2. Modeling the impact of the weight of the robot was not possible, we simply decided to buy new bigger robots to complete mission, with the same software artifacts.
3. We decided to change some COTS as well, change the IMU on both drones and UGV to gain in precisions.

Could we predict failure ? The participants to the challenge were very different. The systems and principles were heterogeneous and criteria for pass/fail were unclear. The winners of the challenge were notified but experience return was not considered in the second increment.

Results and increments for increment 2

Different mission, more difficulties than in the first increment: moving targets, difficult mission, more constraints on human-machine interactions

Observation 1 on mission completion:

With addition of electronic components and the addition of algorithms, the battery of the two UGVs could not complete a 1h mission. We had to switch 2 times for each UGV on field. We observed also the loss of communications with UGV platforms with the loss of performance of the battery.



Figure 11: the necessary set of batteries required to complete a 1h mission for two UGVs



Figure 12: the introduction of moving targets had an impact on DRI algorithms

Observation 2 on mission completion:

Big rapace caught in high grass, could never take off, see on Figure 13.



Figure 13: the "rapace" was not able to take off, because its algorithms misjudged the nature of the grass

Observation 3 on mission completion:

The UGVs platforms were unable to check the perimeter because they were not high enough, as seen on picture Figure 14.



Figure 14: the UGV platform in higher grass than expected

Decisions taken with return on experience:

- From observation 3 and 1, we decided to change UGV platform to fit for combat situations
- From observation 2, we propose to use systems engineering and to change development methods to include testing and more realistic simulation of the battlefield conditions for all systems.

We observed the same communication problems with the platforms, and were unable to solve them. We bought better IMU systems too, and better LIDARs, but we did not evaluate the real contribution to the global mission of those component replacement.

Back to system design: capitalization at reach

As feedback from field to model, we expose an example of a system “function” description, focusing on radio communication problems that happened during both increments of the challenge. We propose to go back to a partial representation of our model, and use the diagram depicted in Figure 6 that could never happen because the UGV fell in the ditch. We focus on elements that we believe caused communications problems with our UGV “Paquerette” and made it fall in the ditch:

- battery weakness of the UGV that causes its computer to shut down communications
- external problems linking video and command executions due to the choice of wifi technology.

Because rebuilding our platform was not possible in due times, we focused on changing antenna, adding the possibility to communicate on two levels.

In our model, we chose to add a parametric view, with manual input, such as Figure 15, which is suitable to describe necessary parameters. Our general methodology, the path described in Figure 2, is impacted by addition of a new view, that may impact original views such as the ones describing system previously mentioned.

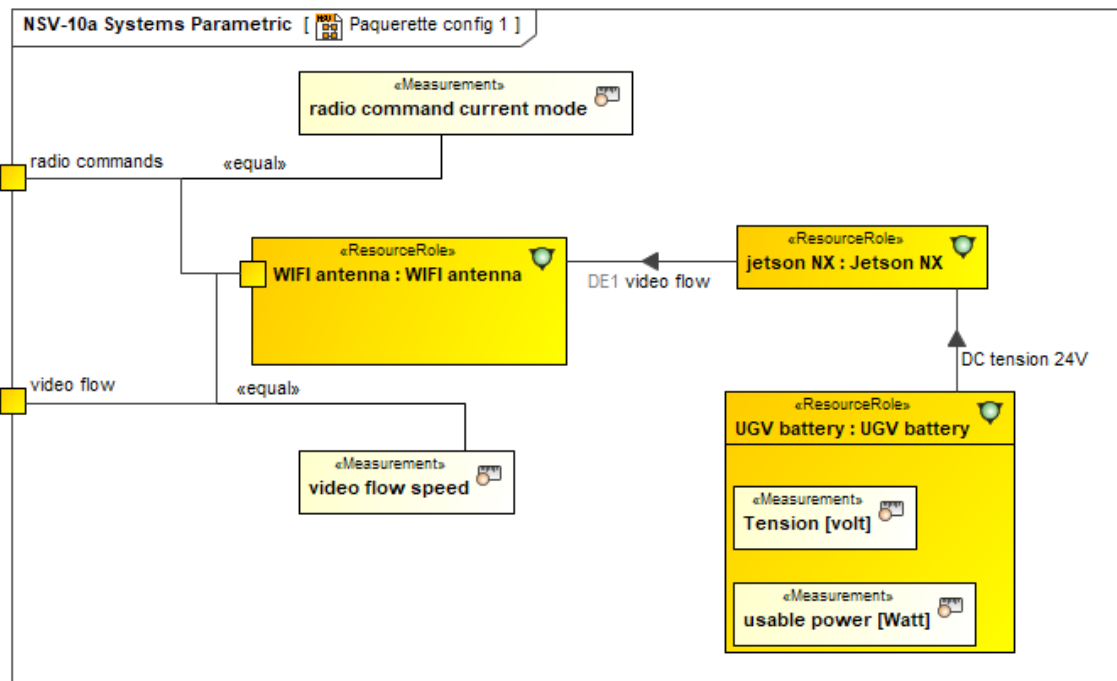


Figure 15: the parametric view example in Paquerette UGV

To conclude on this capitalization loop, we observe that the addition of a new view had questionable impact over the existing model. Indeed, thanks to our overall vision of the meta-model in the form of a graph, we knew which views would be directly affected by our observations. However, we observe that fielded feedback did not help solve the original ELP: adding a view set next time will result in adding modeling time, which may add design delays rather than shortening it in absence of computer-assisted action.

What to expect: anticipating the exponential upscaling of engineering effort

In preparation for increment 3, we want to switch platforms, without losing what we learnt from the first participations. We use the unsolved issues observed with communications on the field, and decided to acquire new ground platforms that physically sustain harder environment;

However, system model or not, we wonder if an increase in costs result in an increase in capability realization? One big platform illustrated in Figure 16 costs five small platform. Will the missions be better covered? Will an increase in mission performance result in exponential costs raise of the system? Can we use system engineering to control expenses?



Figure 16: the problems anticipation with battery logistics in the challenge 2023

We call for the existence of significant methods and/or tools that will link the antennae placement problem to system engineering representations.

On the left of Figure 17, we show the placement of antennae on the roof of the C2 vehicle that we improvised last minute. In the middle of the figure, there is the antennae placement offered “off-the-shelf” by our supplier for the bigger UGV that we plan to use in Increment 3 in 2024. The studies of 3-dimensional spectrum of radio range will have to be studied by the integrator- in this case our lab team. It also means that the integrity of the platform as it is will be modified, regardless of the agreement of suppliers. It also means that capitalization of observations in AF will be repeated at libitum on all platforms, based on regular fielded test results at real scale, or tested in specific domain tools giving antennae spectrum by simulation. Cooperation between parties is mandatory, otherwise its capability realization in collaborative system-of- system will be impossible.

Regardless of the system engineering views selected, the ELP remains if connections between expanses, models such as in Figure 17 and capability realizations are realized by at manual input speed.

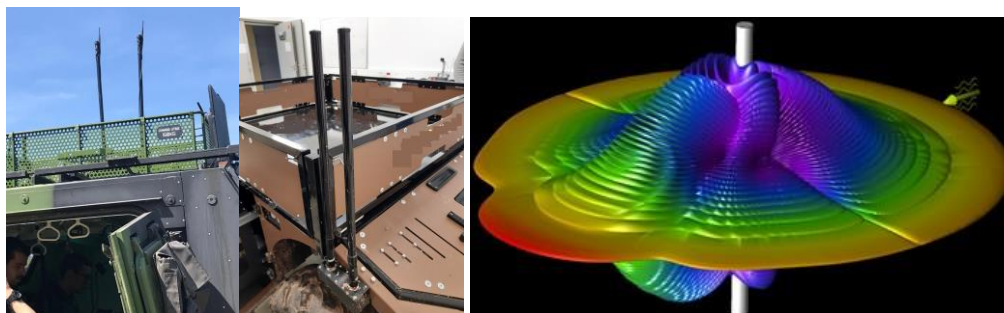


Figure 17: antennae placement: a systems general perspective problem

Outlook and discussion

Future robots will make the problem more complex

Overall emerging behavior of collaboration of robotic platforms will be observed in future challenges, such as the two increments presented in our experiment, and this paper raises questions of the introduction of system engineering practice in robotic battlefield design, and of return on experience capitalization. The new designs and innovative material, such as the ones presented in Figure 18 will also have to be efficiently employed and controlled on battlefield to provide answers to the evolution observed in high intensity combat. Every new technology input such as threats posed by combat proven robotic systems, such as the drones used in Ukraine (Witt, 2022) may change the elaboration of prospective

capabilities computed in technico-operational studies. Yet including accurate details from both environment and combat-proven systems in “capability” specification elaboration reveals the ELP class nature.

Formally, even though a swarm of robots on field might be a set of simple systems, we describe formally the platforms possibilities as such:

$$\begin{aligned}
 & x_{\{ijk\}} \text{ UAV drones,} \\
 & \quad y_{\{ijk\}} \text{ UGV} \\
 & \quad z_{\{ijk\}} \text{ sea drones,} \\
 & \text{with } i \in \{1, \dots, l\} \text{ variations of payload,} \\
 & \quad j \in \{1, \dots, m\} \text{ variations of tasks,} \\
 & \quad k \in \{1, \dots, n\} \text{ energy supply possibilities}
 \end{aligned}$$

It would make $N = 3 \times l \times m \times n$ possible candidates for one job, with $N!$ channels possibilities between platforms. To this number of possible candidates, assuming that each platform group selection to realize a capability is associated to a justification and a domain of validity, hand-made description of swarm will fall into an ELP. And this is without considering the intensity of communications between platforms.

The introduction of new technologies as found in Figure 18 in this context, to be tested in various environments, will eventually create new capability realization possibilities. The already-complex combinatory problem may turn into more complex, as introducing new variants in the swarm creates as many variations as in the first step.

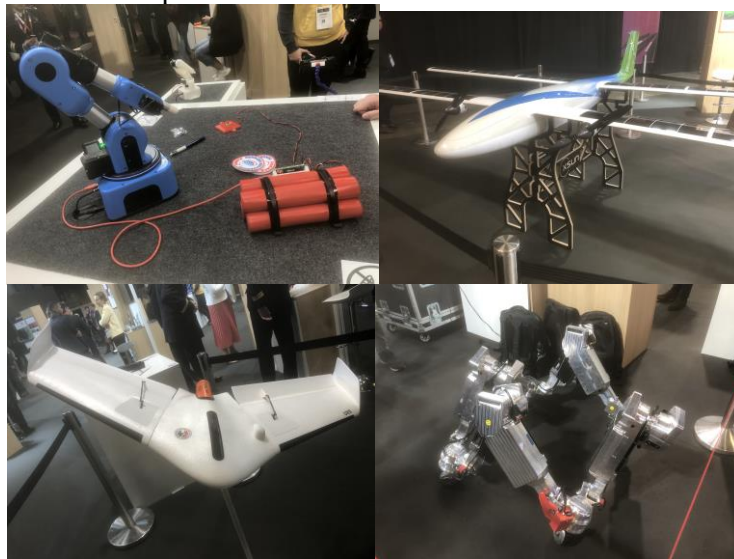


Figure 18: the introduction of new robots designs and controls and capabilities on battlefield

The necessity to define complexity in SE

If complexity is a major incentive to use systems engineering, we have found little information on what complexity means. We picked NP-Hardness to describe our general architecting problem (Hammami, 2018). The definition of NP-Hardness is defined in computational complexity as a decision problem / language is in NP if given an input x , we can easily verify that x is a YES instance of the problem (or equivalently, x is in the language) if we are given the polynomial-size solution for x , that certifies this fact. However, NP problem class is felt to capture the notion of decision problems with “unfeasible” decision procedures, because the set of NP decision problems can be solved in exponential time.

We used graph path representations to model our method. In graph theory, the selection of arbitrary subgraph in a graph that fit specific objectives and constraints can also be NP-Hard. If it were true for

our workflow, then it would explain how we never know in advance the set of necessary views required for an unknown architecture. It can also explain how the sufficient set of views cannot be anticipated at the starting date of the program.

Combination with other forms of descriptions is needed as well, and the sufficient amount of information cannot be known in advance because of the NP-hard nature of ELP problem.

Actually, one symptom of NP-Hardness in systems engineering practice is that if we knew the necessary and sufficient effort required to build an architecture, we would have to know the architecture already. This symptom is described in perception of complexity in details in (Grogan, 2021): “complexity is an intermediate factor with two facets: it enables desired product performance but also requires effort to achieve”.

The example of introduction of robot swarms in a collaborative battlefield illustrates the problem, because it introduces unknown architectures in various environments possibilities with consequent investment before existing, and before verification and validation. Addressing NP-hard problems with inadequate tools creates exponential growth of problems rather than reducing it.

Conclusions

The present contribution describes modeling swarm defense robotic systems in AF in the context of a robotic challenge. The rules of the challenge along with return of experience of two participations are inserted into the model, which lead to the creation of new views and observations on how daunting modeling feels if we want our models to be useful to raise up to challenges. We unveiled “engineering leakage problems” while describing our modeling activity and iterations over incremental capabilities in an innovative context of robotic challenges. We reported observations made from real-scale fielded experiments, and connected the problems to new architectures for the next increments, inspired by current battlefield profiles and new technologies arriving on the market. After describing failures of our experiments in the form of architecture views, we presented recommendations that encourages system engineering practice in robotic development. To represent future capabilities, robotic field state-of-the-art could benefit system engineering practice, and for that purpose, we recommend usage of computational theory in system engineering practice to assess complexity. Along with complexity assessment, tackling system engineering problems such as ELP to existing operation research and their resolution algorithm is required if we are to frame quality, costs and delays trade-offs for system-of-systems on battlefield.

This research is funded by KNDS France.

References

- Brisacier-Porchon, L., & Hammami, O. (2022). *Inconsistent and Incomplete Datasheet: The case for systematic use of requirement engineering*. 17.
- Dassault Systemes. (2022). *CATIA Magic*. <https://www.3ds.com/fr/produits-et-services/catia/produits/catia-magic/>
- Grogan, P. T. (2021). Perception of complexity in engineering design. *Systems Engineering*, 24(4), 221–233. <https://doi.org/10.1002/sys.21574>

- Hammami, O. (2018). System Theoretical Complexity (STC): On the Benefits of System Multiobjective Optimization and Automatic Model Composition. *2018 IEEE International Systems Engineering Symposium (ISSE)*, 1–7. <https://doi.org/10.1109/SysEng.2018.8544445>
- Hause, M., Bleakley, G., & Morkevicius, A. (2016). Technology Update on the Unified Architecture Framework (UAF). *INCOSE International Symposium*, 26(1), 1145–1160. <https://doi.org/10.1002/j.2334-5837.2016.00217.x>
- Henderson, K., McDermott, T., & Salado, A. (2023). MBSE adoption experiences in organizations: Lessons learned. *Systems Engineering*, sys.21717. <https://doi.org/10.1002/sys.21717>
- Janer, D., & Proum, C.-M. (2014). Open Architecture for Naval Combat Direction System. In M. Aiguier, F. Boulanger, D. Krob, & C. Marchal (Eds.), *Complex Systems Design & Management* (pp. 73–84). Springer International Publishing. https://doi.org/10.1007/978-3-319-02812-5_6
- NAF:NAF Subviews. (2010). https://trak-community.org/index.php/wiki/NAF%3ANAF_Subviews
- Papke, B. L., Wang, G., Kratzke, R., & Schreiber, C. (2020). Implementing MBSE – An Enterprise Approach to an Enterprise Problem. *INCOSE International Symposium*, 30(1), 1539–1556. <https://doi.org/10.1002/j.2334-5837.2020.00803.x>
- Rubio, C. G., & Rigaut, T. (2023). An Urban Air Mobility System of Systems UAF & MDAO Application Case. *2023 18th Annual System of Systems Engineering Conference (SoSe)*, 01–08. <https://doi.org/10.1109/SoSE59841.2023.10178495>
- Witt, S. (2022, May 9). The Turkish Drone That Changed the Nature of Warfare. *Annals of War - The New Yorker*. <https://www.newyorker.com/magazine/2022/05/16/the-turkish-drone-that-changed-the-nature-of-warfare>
- Wolny, S., Mazak, A., Carpella, C., Geist, V., & Wimmer, M. (2020). Thirteen years of SysML: A systematic mapping study. *Software and Systems Modeling*, 19(1), 111–169. <https://doi.org/10.1007/s10270-019-00735-y>

Biography



Lorraine Brisacier-Porchon is a system architect. She defended her PHD with ENSTA Paris in November 2023 after ten years of practice in French defense industry. She built her PhD project using applied cases and advocates for optimization as theoretical foundations for systems engineering.



Omar Hammami is a Professor at ENSTA ParisTech, France and an expert in complex systems and system engineering. He has served as an expert for projects evaluation for several international organizations in the field of system engineering with applications in transport and energy, organizations and value chains in semiconductor and embedded systems.

Omar Hammami has published over 200 papers in international conferences and journals in the fields of computer science and system engineering for complex systems. He is involved on a regular basis in the organization of several conferences and events worldwide related to large scale systems. He is actually involved in MENA region for Large Scale System Engineering projects. He holds a Phd in Computer Science from Toulouse University, France and a HDR in Physics from Orsay University, France.

His research interests are in Theoretical Foundations of System Engineering and Multidisciplinary Optimizations.