



**HAL**  
open science

## Mission-based design of UAVs

Jean-charles Chaudemar, Ombeline Aiello, Pierre de Saqui-Sannes, Olivier  
Poitou

► **To cite this version:**

Jean-charles Chaudemar, Ombeline Aiello, Pierre de Saqui-Sannes, Olivier Poitou. Mission-based design of UAVs. Systems Engineering, 2024, 10.1002/sys.21754 . hal-04531039

**HAL Id: hal-04531039**

**<https://hal.science/hal-04531039>**

Submitted on 3 Apr 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.



Distributed under a Creative Commons Attribution 4.0 International License

# Mission-based design of UAVs

Jean-Charles Chaudemar<sup>1</sup>  | Ombeline Aiello<sup>1,2,3</sup> | Pierre de Saqui-Sannes<sup>1</sup> | Olivier Poitou<sup>2</sup>

<sup>1</sup>ISAE-SUPAERO, Université de Toulouse, Toulouse, France

<sup>2</sup>ONERA/DTIS, Université de Toulouse, Toulouse, France

<sup>3</sup>AIRBUS, Blagnac, France

## Correspondence

Jean-Charles Chaudemar, ISAE-SUPAERO, Université de Toulouse, 10 avenue Edouard Belin, BP 54032, 31055 Toulouse, France.

Email:

[jean-charles.chaudemar@isae-supaero.fr](mailto:jean-charles.chaudemar@isae-supaero.fr)

## Funding information

Defense Innovation Agency (AID), France, Grant/Award Number: 2019 65 0090004707501

## Abstract

Over the past decades, Unmanned Aerial Vehicles (UAVs) have increasingly been used in a wide variety of missions that range from surveillance to delivery. Unlike aircraft that always carry goods and passengers from an airport to another, UAVs do not systematically implement the same type of mission. UAVs are indeed multi-mission during their time in operation, and the systems engineering approaches developed for one mission aircraft must be adapted to the multi-mission context. Therefore, UAV design requires application of mission engineering upstream systems engineering, either to assess there is a UAV system that may accomplish a new mission, or to specify a new UAV system according to a given mission. To achieve that goal, the authors of the paper support the use of Model-Based Mission Engineering. A three-layer architecture - purpose, operation, functions or capabilities - is proposed as a design framework for missions. The Goal-Oriented Requirements Language (GRL) serves as mission description language. The paper extends GRL to better address mission-based design of UAVs. It is proposed to distinguish between internal and external resources. A goal detailing mechanism is introduced. A degraded mode evaluation becomes possible. GRL tools make it possible to evaluate how much a UAV system - at least, an operator, a ground station, and a UAV - may satisfy every stakeholder in both nominal and degraded modes. The proposed approach is applied to a high voltage surveillance UAV. The case study enables the introduction of four actors—Authority, Client, UAV and MissionSupervisor—that turn out to be generic and can be reused for other missions and UAV designs.

## KEYWORDS

goal modeling, MBSE, mission, UAV

## 1 | INTRODUCTION

Over the past decades, Unmanned Aerial Vehicles, or UAV for short, have increasingly been used in military and civilian mis-

sions. In both the military and civilian domains, it is possible to identify delivery and surveillance missions. Delivery missions range from weapons delivery on a battle field to delivery-at-home of parcels and delivery-at-hospital of blood bags. Surveillance missions range from spotting enemy positions to building indoor or outdoor inspection,<sup>1</sup> cattle or crops surveillance, and high voltage lines inspection.<sup>2</sup>

Jean-Charles Chaudemar, Ombeline Aiello, Pierre de Saqui-Sannes, and Olivier Poitou contributed equally to this study.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Authors. *Systems Engineering* published by Wiley Periodicals LLC.

Unlike aircraft that always carry goods and passengers from one airport to another, UAVs do not systematically implement the same type of mission. UAVs are fundamentally multi-mission and the missions a UAV is given may frequently evolve over the life time of that UAV. Therefore, mission modeling is a key issue in UAV design.

Making missions an entire and explicit part of UAVs design increases that design in complexity. Those methods and processes which limit the UAV design to the design of the UAV system need to be revisited. This particularly applies to the methods included in Model Based Systems Engineering approaches (see, e.g., ref. [3]).

mKAOS<sup>4,5</sup> is an example of mission-based design method that is associated with a MBSE approach. mKAOS is not open source. By contrast, the approach developed in the current paper is open source. Therefore, the work in the current paper relies on the Goal-Oriented Requirements Language (GRL,<sup>6,7</sup>), which is open source and further well adapted to high-level requirements expression. Moreover, GRL provides us with extension capacities such as stereotypes, and coloring. The current paper extends GRL's expressiveness for the sake of a UAV mission and simultaneously remains compatible with existing GRL tools.

The rest of the paper is organized as follows. Section 2 highlights the key features of a successful mission-based design from its past applications to the space, automotive, and military domains. Section 3 addresses UAV mission modeling. Missions are modeled using an extended version of the GRL. Section 4 discusses a case study: a high voltage line surveillance UAV. Section 5 concludes the paper and outlines future work.

## 2 | MISSION DEFINITION

In our aim to define a mission properly, we have first identified a list of failed missions that justifies the mission engineering approach. A few concepts considered as important are bolded so as to be used in our proposed approach. Indeed, meaningful characteristics of a successful mission are identified in the literature and set up in our modeling framework for the mission definition.

### 2.1 | Stakes and challenges

In the space domain, the commissioning of the first spacecraft may be jeopardized because the missions are too complex, uncertain or not clearly defined.<sup>8-10</sup> Let us, for example, consider the Mars probe that was lost by NASA in 1999.<sup>11</sup> The probe was "about 100 km off course at the end of its 500-million km voyage – more than enough to accidentally hit the planet's atmosphere and be destroyed."<sup>11</sup> If the probe's mission had been more accurately defined relying on a better knowledge of the **environment** of Mars, the loss of that probe might have been avoided.

Likewise, in the aviation domain, aircraft are not left out. Whether it is pilot error, incidents or even accidents may occur while all aircraft systems operate properly. A lack of clear procedures may lead

to these kind of catastrophic events. Consider the crashes of Lion Air Flight 610<sup>12</sup> and Ethiopian Airlines Flight 302.<sup>13</sup> In general, for the preparation of a flight in terms of the aircraft mission, flight procedures provide the pilot with a set of **actions** to be carried out depending on the context.

In the automotive area, the advent of autonomous vehicles brings the need for a description of the roles for each agent or actor (the car and the human driver). Many accidents involving autonomous vehicles are due to a disengagement, that is, a handover of the car control from the **autonomous software** to the driver.<sup>14</sup> Thus, the mission of the car and the mission of the driver have to be consistent and rigorously defined.

Likewise, in the military domain, considering a battlefield where systems of systems are engaged, a rigorously defined mission is of the utmost importance. Let us remark that the term 'Mission Engineering' was first used by the US Department of Defence for enablers of a mission.<sup>15</sup>

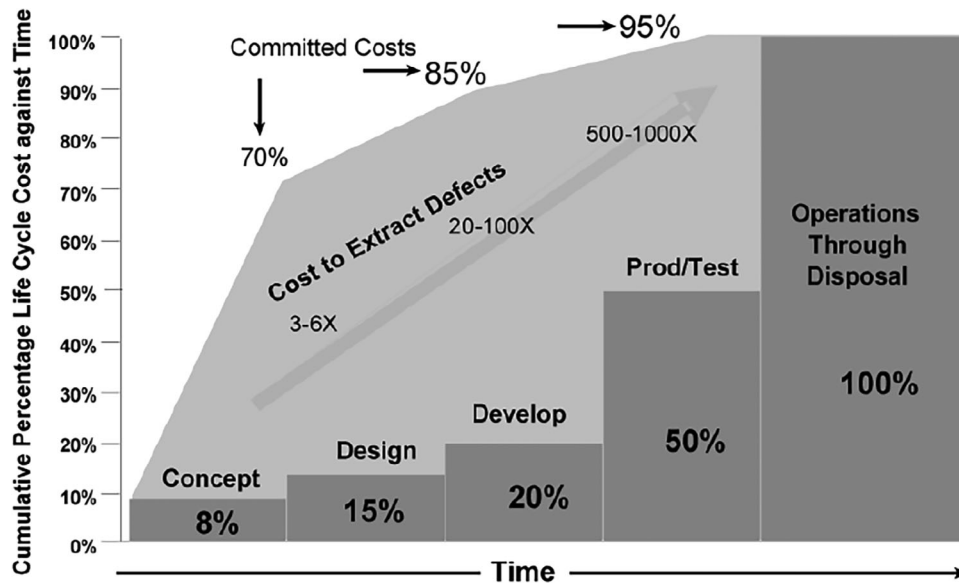
Mission Engineering is a paradigm of Systems Engineering to tackle the technological and operational challenges in the context of defining mission parameters and requirements. According to refs. [16–18], the success of a project or a program greatly depends on the success of its first technical engineering phase, namely "business or mission analysis." Figure 1 shows that making hasty decisions without a proper mission analysis of the System of Interest (SoI) is financially damaging. High costs are engaged from the onset, along with projecting a schedule for the full-scale development, as an enterprise's **strategy**. Therefore, in-depth business models and analysis at the early development phase would help the designer to make right decisions.

### 2.2 | Twofold facet of mission

Mission often pertains to the evolution of a system - product or people - which needs to be forecast so as to highlight the description of this intended system. Different types of missions exist but only two forms of missions are to be distinguished: the missions for technological systems and those for organization. Thus, from both types of missions, we identify a few useful characteristics since we aim at defining a generic mission.

#### 2.2.1 | Mission for technological systems

This mission stems from the need to understand, and to master the intended behavior of a system to be developed. Undergoing predictable **operational situations** is not part of efficient engineering activities.<sup>20</sup> The operational situations rely on the context and the surrounding environment where the system is operating. For instance, in case of windy weather, a UAV might not fly by performing a safe landing procedure. Moreover, throughout operational situations defined at early design phases, the system behavior becomes more tangible and concrete. The description of this behavior, whether it be textual or graphical, enables a better understanding and sharing among all



**FIGURE 1** Committed cost against System of Interest life-cycle.<sup>19</sup>

stakeholders. Thus, the first steps forward from ideas to concepts are indispensable. One of their artefacts is the Concept of Operations (ConOps) document, which ensures the feasibility and acceptability of the intended system.<sup>17</sup>

This ConOps is widely used in other descriptions of the mission by focusing on the process aspects.<sup>21,22</sup> To define the mission, ref. [21] defines several steps integrated into their design methodology. First, they present a functional view of the mission, identifying the various capabilities. Then, they propose to consider the missions that have already been carried out and the assets they can benefit from, before analyzing the new concepts they could incorporate. Once these initial high-level steps are considered complete, the project is defined through the identification of precise mission goals. Moreover, other goal-oriented approaches explicitly propose a modeling of the ConOps in the framework of the interaction between the mission-design and the system design.<sup>4,5</sup> In ref. [23], the ConOps is implicitly formalized through a metamodel which enables to tightly model a set of systems interacting with their environment in a system of systems approach. However, all these studies pass into oblivion the impact of any changes of the ConOps on the mission description. This ConOps is used in our UAV mission-based design approach later on.

## 2.2.2 | Organizational mission

The mission related to organizations or socio-professional categories accounts for a strategic dimension defining a certain vision and ambition. For a given purpose, each organizational entity has to interact with other external entities within an overall system. Thus, the organizational mission describes all inner activities related to these interactions in order to meet a given goal. However an efficient mission statement is not only a checklist of activities.<sup>24,25</sup>

In an administrative framework, the mission statement is the result of heated negotiations that involve many actors and stakeholders, and often remains influenced by political factors.<sup>21</sup> The purpose of the organizational mission is to deal with changes because its main resources are human beings who are changeable and variable by nature. Unlike technology-oriented missions, organization-oriented ones raise significant risks while changing: the system may fail and the associated project may be aborted. Therefore, to define this type of mission, the meaningful concepts are the goals, the stakeholders, the relationships between stakeholders.

## 2.2.3 | Synthesis

Whatever the mission and the mission field, a comparative survey of the literature leads us to identify a tiered architecture characterizing missions as follows:

- *Goal*: the goal level enables to define the *raison d'être* of a mission.
- *Stakeholders*: the stakeholders work out the strategy, which pertains to the business aspects and the political aspects for a technology-wise mission and an organization-wise mission, respectively.
- *Operation*: by enhancing the ConOps, the operation level deals with the activities that are relevant so as to meet the given goal and to be compliant with the proposed strategy.
- *Resource*: the resource level identifies all suitable people and products necessary for the *Operation* level.

## 2.3 | Adopted mission-based design

In ref. [25], the author identifies three steps in a mission: (1) statement of the mission, (2) identification of the mission scope, and

**TABLE 1** Proposed mission framework for UAV, with the focus of this paper shown in blue.

<ul style="list-style-type: none"> <li>• Performances</li> <li>• Ontology</li> <li>• Lessons learnt</li> </ul>	<u>Mission Definition</u>			<ul style="list-style-type: none"> <li>• Simulation</li> <li>• Test</li> <li>• Scenario</li> </ul>	
	<b>Purpose</b>				<u>Mission implementation</u>
	<ul style="list-style-type: none"> <li>• Goals</li> </ul>	<ul style="list-style-type: none"> <li>• Stakeholders</li> </ul>	<ul style="list-style-type: none"> <li>• Strategy</li> </ul>		
<b>Operation</b>					
<ul style="list-style-type: none"> <li>• Actions</li> <li>• Modes</li> </ul>	<ul style="list-style-type: none"> <li>• Environment</li> <li>• Threats</li> <li>• Regulations</li> </ul>	<ul style="list-style-type: none"> <li>• Operator</li> <li>• Situations</li> </ul>			
<b>Functions / Capabilities</b>					
<ul style="list-style-type: none"> <li>• Flight</li> <li>• Surveillance</li> </ul>	<ul style="list-style-type: none"> <li>• Communication</li> <li>• Autonomy</li> </ul>	<ul style="list-style-type: none"> <li>• Payload use</li> </ul>			

(3) analysis of the mission impacts. Moreover, this description of the mission has a great importance in the military domain as the interoperability and the variability in a mission are crucial. Thus,<sup>22</sup> proposes a mission architecture framework based on the DoDAF (Department of Defence Architecture Framework<sup>26</sup>) model, called “OMAF” (Operational Mission Architecture Framework). Relying on OMAF, a generic framework for UAV missions has been conceptualized and drawn up. This concept of mission proceeds from three stages:

- Mission definition,
- Mission analysis, and
- Mission implementation.

### 2.3.1 | Mission definition

Table 1 describes the *mission definition* through three conceptual layers. The first layer pertains to the purpose. The *raison d'être* of the mission, the goal expected by stakeholders, along with the strategy with regards to diverse alternatives, all of these concepts are included in this layer. These concepts play an important role in the formulation of the mission in terms of activities in the next layer.

In the second layer, the mission operation is described by actions carried out through interaction between operator and external actors in an operational environment. The operational environment allows us to highlight a few situations (normal and abnormal) that the UAV can be faced with. These situations trigger operational modes related to the expected behavior of the UAV. Moreover, while in operation, the UAV behavior is resilient to threats, while being compliant with standards and regulations.

The other features in the third layer are directly linked to the design of the intended system, that is, the UAV. This layer consists of depicting the high-level functions and the capabilities. On the one hand, among the functions, fundamental functions for UAV are defined, such as flight (including command-control), communication. For the sake of autonomy, a navigation function is embedded so as to manage flight plans. On the other hand, additional functions or capabilities relate to the payload. For the purpose of surveillance, capabilities are constraints or limitations on functions, for example, payload mass, technology to be used for surveillance. It is worth noting that this list of features for mission definition is not comprehensive and has to be adjusted according to the intended system, together with the operation dealt with in the upper layers.

### 2.3.2 | Mission analysis

In the literature, the *mission analysis* is part of the first process for the system design.<sup>21,27</sup> This technical process enables us to emphasize the properties required for the mission success. A few properties are measurable like performances or expected values in terms of optimization of design variables. Thus, the mission analysis may suggest trade-offs between several alternative solutions.

Another type of properties is mainly qualitative, as the mission features pertain to feedback and lessons learnt. The safety and security analyses are often deduced from abnormal behaviors, incidents and accidents. They are of the utmost importance for aerial vehicles such as UAVs.

Therefore, the whole mission analysis tends to draw up a common definition of the mission towards a kind of ontology so as to propose

a generic and high level formal description. This paper does not tackle these aspects that will be elaborated on in another submission.

### 2.3.3 | Mission implementation

To verify and validate the mission analysis,<sup>21,27</sup> suggest its implementation by simulating it. Running a simulation is a good way to be confident of the mission success. To do that,<sup>27</sup> uses the ExtendSim software to simulate models for discrete, continuous and hybrid systems. However, many tools can be used and test cases can be generated from models too.<sup>28,29</sup>

In this paper, we mainly address the mission definition concept by modeling it in the next section.

## 3 | UAV MISSION MODELING

The mission framework adopted in the current paper is depicted in Table 1. Its first application allows us to select pertinent key concepts to be modeled. In the *Purpose* layer, it is important to describe *Goals* and *Stakeholders*. The stakeholders' needs are indeed crucial inputs throughout the upstream phases. On the contrary, *Strategy*, which mainly deals with business aspects, is overlooked in our models in the next sections. Regardless, *Strategy* is considered as a driver of models, given that the utmost purpose is the acceptance of the end product or system by the customer.

The second layer, *Operation*, is worth being considered because of its concepts. *Actions*, *Operator* and *Regulations* contribute to formalize the upper layer. In the next sections, *Actions* will become *tasks* as they implement activities in order to achieve *goals*. *Operator* is modeled by a specific *actor* later on, called *MissionSupervisor*. *Regulations* concept is symbolized by a specific actor called *Authority*.

Finally, the third layer, *Functions/Capabilities*, enables to highlight concepts such as *Flight*, *Communication* and *Payload use* through a specific actor called *FlightSystem*, a subgoal called *Communication*, and a payload *resource* element, respectively.

### 3.1 | Related work

The literature offers many articles mentioning the terms mission description, specification or modeling. Most of them consider the mission as a set of elementary actions and focus their work on optimizing their allocation to system components, or their planning over time. Although an interesting area, this is not the point we try to address in this work.

Arcadia methodology exposes the mission in an *operational analysis* viewpoint for the sake of stakeholders' needs.<sup>30</sup> These needs are mainly functional since they will be a key driver for the system design. Yet to elaborate on goals is not explicitly covered in Arcadia. Arcadia seems to be also not suitable for applying it to standard architecture frameworks such as the DoDAF. Ref. [31] uses the standardized

Object Process Methodology (OPM) to describe a drone mission as per the DoDAF. The fact that DoDAF is general enough, enables to associate an operational context interpretation to DoDAF operation views concepts, and to describe a high level mission goal.<sup>31</sup> focuses on the ontological work and the formalizing of DoDAF concepts. However, the elaboration on goals in terms of interconnections still lacks in OPM.

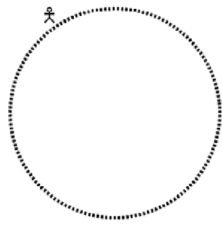
Far less publications consider missions start as stakeholders' high level goals or needs and put a lot of effort into finely capturing and analyzing their interconnections and decomposition relations. Already mentioned is the OMAF,<sup>22</sup> which proposes a mission architecture framework based on the DoDAF but decoupled from the engineering management effort and oriented towards operational stakeholders. Rich in terms of concepts, OMAF is very military organization oriented and lies at the framework level. We used it as a strong source of inspiration, and proposed a more UAV mission specific perspective of the framework overview.

Another group of publications come from the context of systems of systems. Indeed, in this context, every constituent system has an individual mission, that is not dedicated to making successful the global mission of the system of systems. Those two levels of mission hence need to be described precisely, in order to be able to ensure the consistency between the two levels. For example,<sup>32</sup> relies on mKAOS<sup>4,5</sup>/dynBLTL<sup>33</sup> to assess the consistency of individual missions between each other, to assess their compatibility with the global mission, as well as to identify emergent behaviors from joining constituent systems individual behaviors. mKAOS is then a good example of mission-based design method that is associated with a MBSE approach. mKAOS has a strong focus on addressing system of systems issues, that are not the same as the ones we are dealing with here (even if the UAV system could be modeled as a system of systems, which is not the case in the current paper). In addition, mKAOS tooling is not open source while the approach developed in our current research aims to be so.

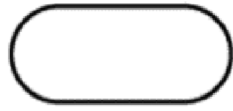
The work presented in ref. [23] shares a number of concerns with us. In the wider process they introduce, the mission description itself lies in two steps: mission decomposition, and mission definition, respectively (the next step "Role Definition and Assignment" could also have been mentioned). The former is closer to our concern. The authors of ref. [23] propose to use a dedicated SysML profile, extending traditional SysML Requirement diagrams. In particular, they stress that the only decomposition available in the Requirement Diagram has a conjunction semantic, while alternatives ("variation points" in the referenced article) are needed in an advanced mission modeling. In their profile, tagged values are also introduced to capture a risk level and a priority level. While this proposition is inline with a great part of our work, we consider that GRL (see below), which is standardized, offers better tools and a stronger basis than SysML Requirement Diagram for this first step. In addition, risk is not directly part of our concerns, whereas our goal decomposition goes beyond the use of OR and AND.

According to ref. [34], the GRL for short, is a good candidate to capture needs from the stakeholders. Like KAOS,<sup>35</sup> i\*<sup>36</sup> and TROPOS,<sup>37</sup> the GRL allows to model systems' goals. Further, it supports dynamic interactions between several components in a system. Therefore, the work in the current paper relies on the GRL<sup>6,7</sup>, which is further

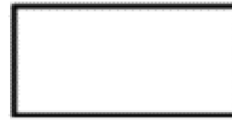
Actor Name



(A) Actor



(B) Goal



(C) Resource



(D) Task

**FIGURE 2** Main items representation in GRL. GRL, Goal-Oriented Requirements Language.

well adapted to high-level requirements expression. The current paper extends GRL and simultaneously remains compatible with existing GRL tools.

In this paper, the objective of using GRL is to capture and model the stakeholders' needs in the context of a UAV mission. Therefore, GRL is used and extended with UAV missions in mind.

This section starts with a presentation of GRL, focusing on the language elements used in our research project. Discussion goes on with a set of extensions that we propose to adapt the GRL to UAV missions descriptions. An extended GRL model is provided as a case study.

## 3.2 | GRL relevant native elements

The GRL, inspired by the *i\** language and the Non-Functional Requirements (NFR) framework, is one of the two subsets of the User Requirements Notation (URN).<sup>34</sup> GRL was created to think of requirements in terms of goal modeling. With GRL, it is possible to represent a goal model through different views that can evolve based on the user's knowledge of the system (the latter being modeled in the goal model). Further, GRL is supported by the open-source tool jUCMNav<sup>7,38</sup> which is an Eclipse plug-in. Users of jUCMNav may model systems using the GRL. They may further analyze GRL models through a quantitative evaluation, a qualitative evaluation, or, a hybrid evaluation. An example of GRL model analysis is presented in Section 4.

Let us now focus discussion on applying GRL to missions executed by UAVs. Four GRL elements have been adapted: *Actor*, *Goal*, *Resource* and *Task*.

### 3.2.1 | Actor

Actors are the active elements of the system or its environment. They expect that *goals* will be satisfied, and for that, the other GRL intentional elements (*tasks* and *resources*) present in the model are taken into account by the evaluation. *Actors* can contain *goals*, *tasks* and *resources*, and they can be linked together through *dependencies*, or the two links described further in this section.<sup>34</sup> The graphical representation of an actor is given in Figure 2A.

Note that the name of the *actor* is written at the top left of the circle representing it.

### 3.2.2 | Goal

The GRL distinguishes between two types of goals: *hard goals* and *soft goals*.<sup>34,39</sup> As far as UAV missions are concerned, the former is the only required one because we assume that all our goals are achievable and measurable. Therefore, *goal* will be used instead of *hard goal* in the remainder of this paper. *Goals* are used to express what the system should achieve. Their graphical representation is provided in Figure 2B.

### 3.2.3 | Resource

Resources describe physical entities that can be used to execute a *task* in order to achieve a *goal*.<sup>34</sup> Integrating *resources* into the GRL model is important because these resources can indicate the mode where the system will operate. Indeed, if the *resource* is available, the system will operate in a nominal mode. Conversely if the *resource* is unavailable, the system will operate in a degraded mode. A graphical depiction of *resources* is presented in Figure 2C.

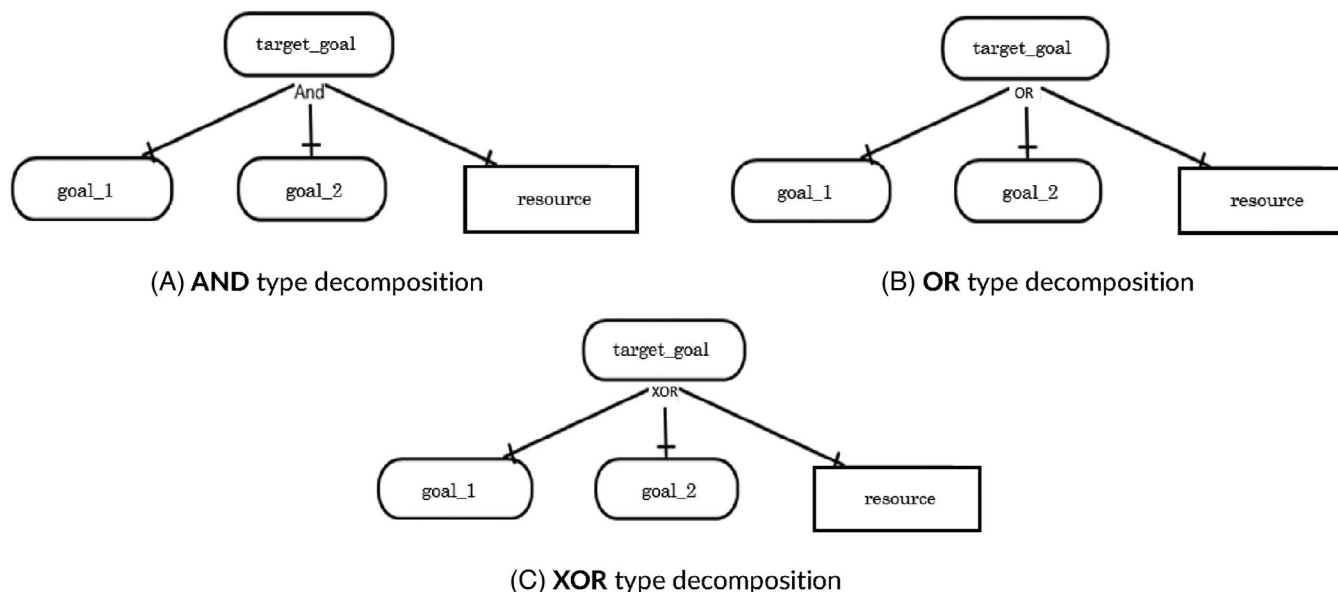
### 3.2.4 | Task

The last GRL intentional element used in this paper is the *task*.<sup>34</sup> Tasks describe the actions to be performed in order to achieve the *goals*. In the current paper, it is assumed that *tasks* are performed by *resources*. Their graphical representation is provided in Figure 2D.

These elements are connected to each other through different kind of links; two of them are used in this paper, and presented below.<sup>34</sup>

### 3.2.5 | Decomposition

There are three different types of decomposition: AND, OR, and XOR (Figures 3A, 3B, and 3C respectively).



**FIGURE 3** Types of decomposition.

One interest in using *decomposition* is to allow users to provide several alternatives to satisfy the target intentional element in the same goal model. *Decomposition* further allows to prioritize elements by differentiating the source intentional elements from the target intentional ones.<sup>34</sup>

### 3.2.6 | Contributions

We use the second link *contribution* to indicate how much a source intentional element contribute to the satisfaction of a target intentional element.<sup>34</sup> The contribution link is tagged with qualitative and quantitative ([-100; 100]) values that represent the ratio of satisfaction being propagated from the source to the target. To improve the readability and the understandability of our models, in the next section, two stereotypes extending *contribution* are added. These stereotypes have a restricted domain of application, thus they illustrate particular cases.

## 3.3 | GRL extensions

We can note that a few GRL elements presented in previous section need to be tailored in order to describe a UAV mission and capture stakeholder's needs. For instance, the client and the operator, two distinct actors, could have goals whose achievement is connected. For instance, one goal of the client could be achieved only if a goal of the operator is also achieved. This kind of relationship allows us to propose a flexible and simple modeling so as to characterize just necessary equipment for the design of a UAV. Of course, this is not specific to our model.

Let us take *resources* as an the example. From a classical GRL model, we are not able to know if a physical element is required in all UAVs

for them to be operational or if it is an additional element added for one kind of mission. A specific camera can be embedded to capture better images during a movie shooting, but this specific camera is not mandatory for the UAV to fly properly.

Therefore, we propose to add five concepts to the existing semantic of the GRL: *relationship stereotype*, *external resource*, *goal detail*, *degraded mode*, *goal hierarchy*, and *multiplicity*.

More concrete examples are presented below, to describe these five concepts and to highlight their utility. The UAV taken as an example in the following sections is willingly complex in order to show that the proposed methodology can handle a wide range of UAV missions and systems.

### 3.3.1 | Relationship stereotype

By definition, a stereotype allows to extend the expressiveness of model elements. In GRL, the stereotype mechanism consists in defining a metadata that is identified by both a name and a value. In addition, Object Constraint Language (OCL) constraints are drawn up in order to verify the static semantics of the model using stereotypes.<sup>40</sup> For the sake of the description of the UAV mission, three stereotypes have been defined: two are related to the relationship *contribution*, whereas the last is in the context of *decomposition*. The first stereotype called *participate* extends *contribution*. It states the relationship between two goals belonging to two distinct actors, by meaning that one goal participates in the satisfaction of the other goal. The second stereotype of *contribution*, named *contribute* depicts the link between one task and one goal, that is, the concerned task contributes to the goal satisfaction effectively. The stereotype of *decomposition*, named *require*, is a link between two goals of the same actor, and means that one goal requires another one for its satisfaction.



These three stereotypes are respectively constrained in three OCL invariants as follows:

```

OCL #1
context Contribution
invariant GoalAsContributionParticipate:
    (self.src.oclAsType(grl::IntentionalElement)).type
=IntentionalElementType::Goal
    implies
    (self.oclAsType(grl::Contribution).getMetadata('participate')
='1' and
    (self.dest.oclAsType(grl::IntentionalElement)).type
=IntentionalElementType::Goal)

```

```

OCL #2
context Contribution
invariant TaskAsContributionContribute:
    (self.src.oclAsType(grl::IntentionalElement)).type
=IntentionalElementType::Task
    implies
    (self.oclAsType(grl::Contribution).getMetadata('contribute')
='1' and
    (self.dest.oclAsType(grl::IntentionalElement)).type
=IntentionalElementType::Goal)

```

```

OCL #3
context Decomposition
invariant GoalAsDecompositionRequire:
    (self.src.oclAsType(grl::IntentionalElement)).type
=IntentionalElementType::Goal
    implies
    (self.oclAsType(grl::Decomposition).getMetadata('require')
='1' and
    (self.dest.oclAsType(grl::IntentionalElement)).type
=IntentionalElementType::Goal)

```

### 3.3.2 | External resource

The new concept *external resource* allows us to distinguish between those resources which are required for the system to work properly from those resources which are optional. For instance, a UAV is made up of an engine, a battery, an onboard computer, sensors, and actuators. All these resources are necessary to allow it to fly. For this reason we define them as *Internal resources*. In contrast, resources such as thermal cameras, robotic arms, and release/drop devices are not required to ensure the flight of the UAV. They can be considered as “optional” because they are embedded by the UAV and required to execute a precise mission, but their presence has no impact on the UAV operation. We define them as *external resources*. To differentiate *internal resources* from *external resources* in a GRL model, we color *external resources* in blue.

### 3.3.3 | Goal detail

To avoid overloading GRL models, we only propose the high level model of the concerned system and of its environment. However, staying at a high level of modeling may result in a lack of data required to fully understand the real behavior and functioning of the system. For this reason, a second extension called *goal detail* is added. This extension can be applied on the goals of the GRL model that would require further details, but incorporating these details would mess up the actual diagram. Such goal details will then be represented in a separate actor, often lying in another diagram. To identify such goals in their initial diagram, they are colored in yellow.

To homogenize the GRL models produced, we have introduced some drafting rules as follows:

- Goal detailed: passive voice sentence (noun + BE + past participle + System).
- Task: infinitive sentence (bare infinitive + noun).
- Resources: nominal sentence (noun).

Applying these rules for the yellow goal “*Flight be ensured*” in Figure 7, the actor that details this *goal detailed* is named “*FlightSystem*.”

In Figure 4, the “*FlightSystem*” actor contains three goals: one goal is to ensure UAV’s movement, another is to control it and the third one is to stabilize it. Also, two tasks are associated with these goals: to provide propulsion and to ensure flight. These tasks contribute to the three goals: the control is provided by the propulsion at 50 and avionics systems at 50 as well, while the displacement is mainly managed by the propulsion and the stability is mainly managed by the avionics. Indeed, the values assigned to the contribution connections reflect this given situation.<sup>41</sup>

### 3.3.4 | Degraded mode

To increase the chances of success of the UAV mission, a third extension is added to make it possible to model the Sol in degraded mode.

A similar functionality is proposed by the tool jUCMNav in its evaluation mode, by allocating a low or zero satisfaction level to a GRL artefact. Being restricted to the evaluation mode, this functionality does not provide the user with the possibility to understand at a glance the behavior of the system when all of its components are not fully efficient. In addition, to use this functionality, a precise scenario should be implemented in the tool and run before addressing the result to the user. By adding the extension “degraded mode”, we offer to the user the ability to quickly see the repercussions of a loss of functionality of a system component. It then becomes possible to predict consequences of a failing, or underperforming, GRL intentional element. Those consequences are evaluated in terms of maximal meeting of stakeholders expectations appearing in the rest of the GRL model, apart from any given scenario. If the results of this analysis are not convincing, one can update the architecture of the GRL model to be able to better

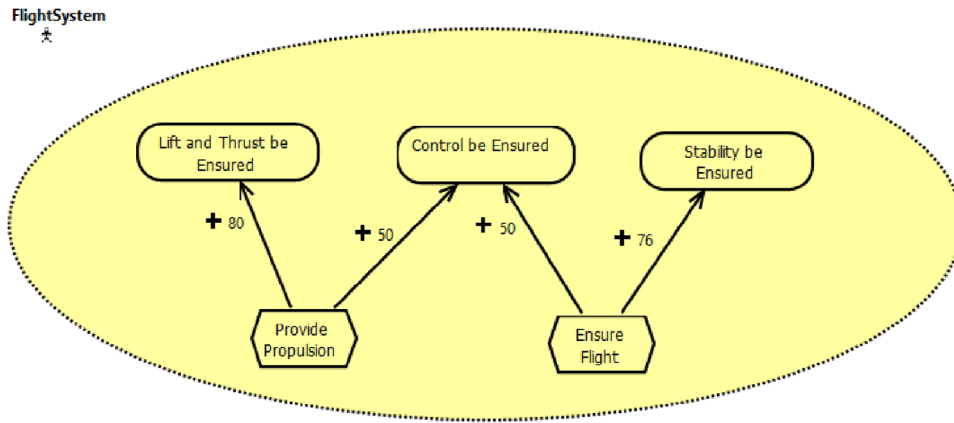


FIGURE 4 Actor detailing the goal "Flight be ensured."

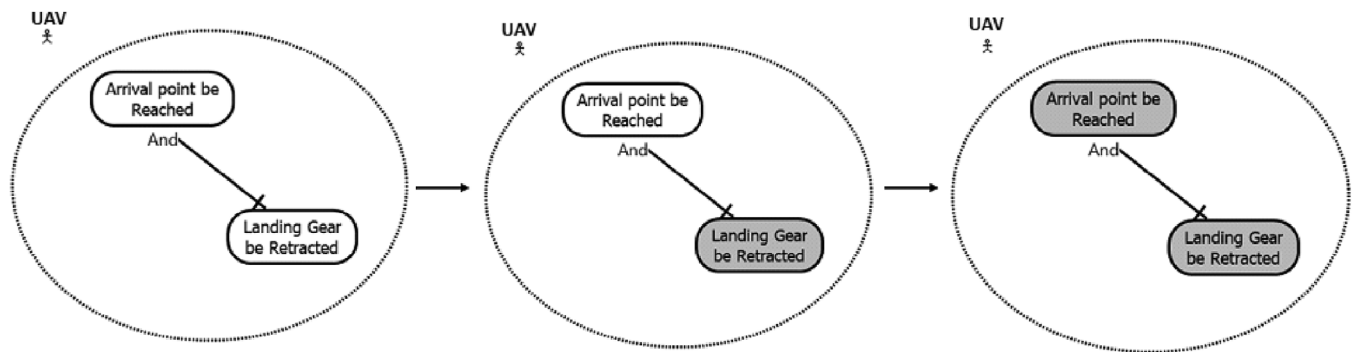


FIGURE 5 Propagation mechanism of a degraded mode in a GRL model. GRL, Goal-Oriented Requirements Language.

compensate the degradation. For instance, it is possible to reorganize links between intentional elements, or, to add resources which make it possible to compensate the loss or underperformance of a component, and avoid the associated impossibility to meet a goal. When the extension *degraded mode* is applied, intentional elements that are not *satisfied* for goals, *performed* for tasks, and *available* for resources, appear in grey in the GRL model. Thanks to a propagation mechanism, it is possible to propagate the fault on the other elements of the model that are impacted by the failure of the satisfaction of the first intentional element (goal, task, or resource). A short example is given in Figure 5.

In Figure 5, the first goal not met is "Landing Gear be Retracted." This fault may compromise, fully or partially, accomplishing the mission and impact the satisfaction of the goal "Arrival point be Reached." Because the goal "Arrival point be Reached" cannot be fully satisfied, it also appears in grey thanks to the propagation mechanism.

### 3.3.5 | Goal hierarchy

To be able to manage the different roles that the goals are based on, their position in the model and the relationships they have with

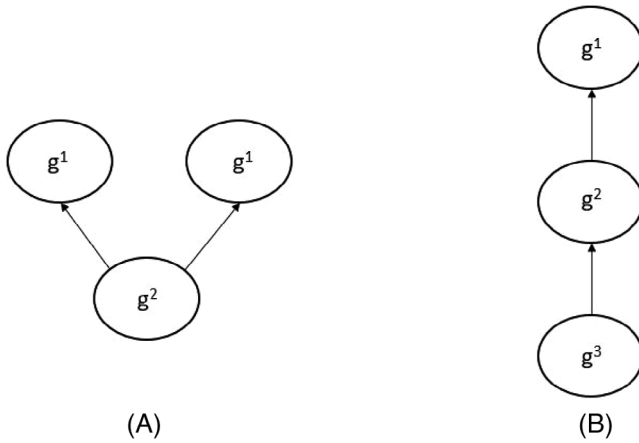
other intentional elements, we establish a hierarchy of goals. Their rank depends on the antecedent goals they have. A goal  $g$  is being considered as an *antecedent* of a goal  $g'$  if and only if  $g'$  is reachable from  $g$ , directly or indirectly, following the contribution or decomposition relationships.

By convention, a goal being antecedent of no goal, is allocated a rank value of one, and is called a *high-level goal*.

When a goal indeed have at least one antecedent goal, its rank value is the maximum rank value among its antecedent goals, plus one. Such a goal, hence having a rank value strictly greater than one, is called a *sub-goal* (Figure 6).

### 3.3.6 | Multiplicity

Finally, we borrowed the concept of multiplicity from the UML language<sup>42</sup> and added it to the GRL to indicate the number of each actor that may be involved. This allows us to know the number of instances for each actor we have in the GRL model. This value will then be usable to support the architectural description of the system, or to check its consistency with the GRL model. Multiplicity may also be considered in the global satisfaction computation, for



**FIGURE 6** Definition of antecedents.

instance by weighting the satisfaction associated to an actor having multiple instances.

As a reminder :

- a natural number  $n$  alone means exactly  $n$  instances,
- an interval of values, denoted  $m..n$ , means that the number of instances is at least  $m$  and at most  $n$ ,
- $*$  may be used instead of a number, to indicate any value that makes sense (in particular  $0..*$  or  $*$  mean several, and  $n..*$  means at least  $n$ )

Graphically, multiplicities are added to the GRL model in a comment box placed next to the name of the actor.

So far, all required elements of the GRL and the extensions proposed are defined. The next subsection discusses a short example. Section 4 discusses a more complete case study.

### 3.4 | GRL modeling in application

This subsection presents the GRL model of the actor UAV (Figure 7). We assume that the unrealistic UAV (because of all the equipment embedded) modeled have a retractable landing gear and carry a payload consisting of a thermal camera, a robotic arm and a release and drop device. Its main goal is to reach its final destination which corresponds to its *high-level goal* entitled “Arrival point be Reached.” Three *sub-goals* are required to satisfy the *high-level goal*: “Landing Gear be Retracted”, “Flight be Ensured”, and “Load be Carried.” The last is satisfied by performing the task Load the Drone and resources linked to this task allow the task to be executed. In the example presented in Figure 7, resources Thermal camera, Robotic Arm, and Release and Drop Device are blue which means they are *external resources*. Regarding the yellow goal “Flight be Ensured”, its detail is previously given in Figure 4, and this goal is met by performing the task Manage Flight. To execute this task several resources are required: Pilot, Actuators, Onboard Computer, and Sensors. The resource Pilot is an *external* one because a flight can be performed autonomously, but the three other resources are necessary to allow the UAV to fly, they are *internal resources*. The resources are

related to each other and to tasks by *decompositions*; here AND and OR *decompositions* are used. The tasks are related to goals through *contributions* and goals can be related to each other with *decompositions* or *contributions*. The nominal mode of this example is provided in Figure 7.

The model of Figure 7 was built with the tool JUCMNav, which also offers the possibility to evaluate the GRL model.<sup>6,7,38</sup> To do that, a few intentional elements present in the GRL model are given an initial satisfaction value which is identified with (\*) on the GRL model. A satisfaction value is an integer, usually taken from the interval  $[-100; 100]$ , with  $-100$  meaning no satisfaction at all (total failure for example), and  $100$  expressing full satisfaction. The values introduced in the following tables are essentially subjective, the focus of this demonstration being on the propagation mechanism. These initial satisfaction values are propagated through the entire model and the satisfaction values of the other intentional elements of the GRL model are computed according to the algorithms of the JUCMNav and the *contributions* values, or the type of *decompositions* used. The algorithms of JUCMNav can perform three types of evaluations: quantitative evaluation, qualitative evaluation, or hybrid evaluation. A quantitative evaluation is run in Section 4. Analyzing the GRL model is important to detect inconsistencies or weaknesses of the proposed architecture in the GRL model. The evaluation of the model is also based on other criteria such as:

- the direction of the propagation which can be *forward*, *backward*, or *mixed* according to the location of the intentional elements that have an initial satisfaction value in the model,
- the order in which *decompositions* and *contributions* are evaluated,
- the function (that can be given by users) used to compute the satisfaction value of the relation (*decomposition* or *contribution* in our case).

Note that the list of criteria that allow to evaluate GRL models presented above is not exhaustive.

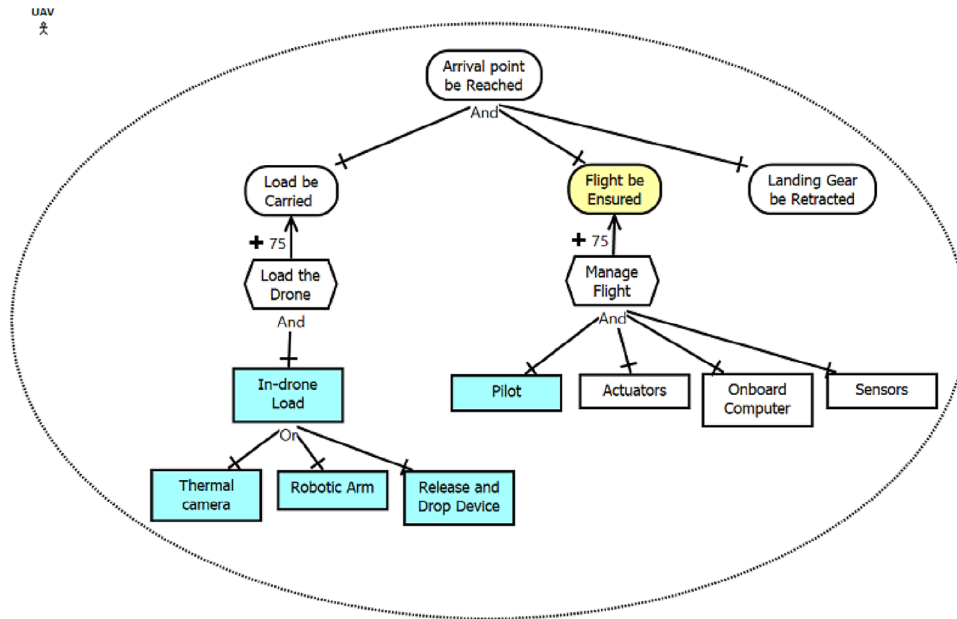
## 4 | CASE STUDY

Section 3 has presented the GRL, the way we extend it and the way we apply all the theoretic notions on a UAV that serves as concrete example. Given that the objective of this paper is to propose a methodology that enables specification and processing of UAV missions before designing a UAV, this section discusses a concrete case study where the GRL is used to complete the level entitled Goal.

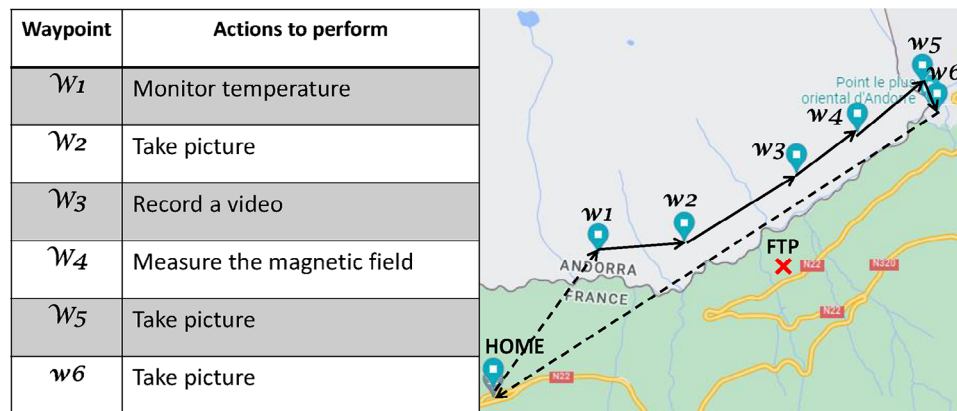
### 4.1 | Mission description

A UAV is given a high voltage power line surveillance mission. Figure 8 presents the actions to be performed by the UAV and the trajectory the latter has to follow.

Figure 8 identifies six waypoints by  $w_1$ ,  $w_2$ ,  $w_3$ ,  $w_4$ ,  $w_5$ , and  $w_6$ . Six waypoints where the UAV is expected to perform specific actions described in the column “Actions to perform.” Between two waypoints,



**FIGURE 7** GRL model of the actor UAV. GRL, Goal-Oriented Requirements Language; UAV, Unmanned Aerial Vehicle.



**FIGURE 8** Description of the high voltage power line surveillance mission executed by a UAV. UAV, Unmanned Aerial Vehicle.

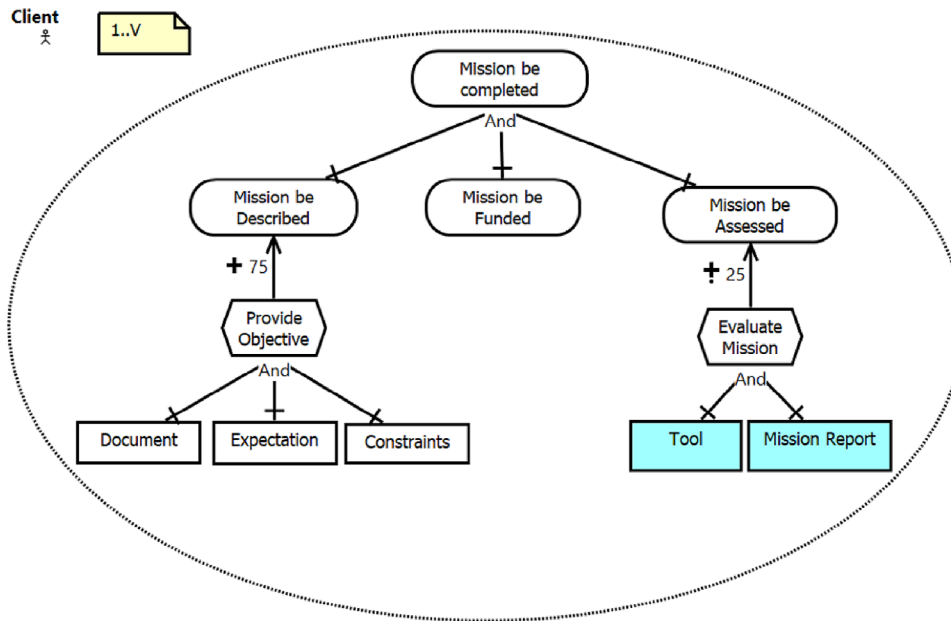
the UAV is inspecting the high voltage power line it is flying over. The point **HOME** locates the base of the UAV, which serves for take-off and landing. The dotted lines represent the trajectories that the UAV must follow between its base and the first waypoint  $w_1$ , and between the last visited waypoint  $w_6$  and its base. The full-line arrows correspond to the trajectory to be taken by the UAV to fly over the high voltage power line and join the waypoints where additional actions are requested. The point identified by a red cross and called **FTP** (Final Termination Point) represents the place where the UAV can land safely in case of emergency (e.g., loss of connection, battery failure).

Among many possible high-voltage lines, the one we selected is located in high mountain. Therefore, weather conditions can be harsh (strong wind, convection currents). Furthermore, the communication between the UAV and the ground station can be unstable due to the flight environment. The surveillance mission itself makes the situation difficult for the UAV. Examples include the proximity that the UAV

will have with the electrical line during the flight, and the autonomy that the same UAV should have to execute the mission safely. It is therefore important to take into account all the parameters related to the mission, and related to the environment in which the UAV will operate during its mission to design a UAV able to achieve the given mission.

#### 4.2 | GRL model

In this part we present the GRL model built to complete the *Goal* level which should help engineers to capture the stakeholders' needs, and subsequently, write the high level requirements. To complete this first stage of preliminary design of a UAV to carry out a high voltage power line surveillance mission, four actors are identified for a modeling in GRL:



**FIGURE 9** Client GRL model. GRL, Goal-Oriented Requirements Language.

- **Client:**

The **Client** actor is the customer of the mission. Its role is to give the objectives to be achieved, the actions to be carried out, and optionally, the means available for the other stakeholders to accomplish the mission. For instance, they can provide or require the use of a specific image processing tool to analyze images taken by the UAV during its operation. In the **Client** GRL model proposed in Figure 9 this entity is represented by the sub-goal “Mission be Assessed”, the task *Evaluate Mission*, and the external resources *Tool* and *MissionReport*. The high-level goal of this actor is “Mission be completed.” For this reason, a precise description of the mission and its objectives is expected by the other stakeholders. This need is expressed through the sub-goal “Mission be Described.” To satisfy this goal the task *Provide Objective* should be performed, and to do that, three internal resources must be available: *Document*, *Expectation*, and *Constraints*. The actor **Client** is also in charge of financing the mission which is modeled with the sub-goal “Mission be Funded.” The GRL model of the actor named **Client** is presented in Figure 9.

- **Authority:**

The second actor **Authority** aims to provide current standards and regulations that the UAV should meet to be authorized to perform the operation that was given by the actor **Client** previously described. In addition to that, to ensure that the current regulations are well respected by the UAV, the actor **Authority** is also responsible for maintaining the regulations and adapting for a while. These two notions are modeled in the high-level goals “Regulation Respect be Ensured”, and “Regulation be updated” in Figure 10. To satisfy the first high-level goal “Regulation Respect be Ensured” the actor **Authority** follows up the UAV certification file that is supposed to be completed by both GRL actors **UAV** (which also includes the UAV designer or supplier) and **MissionSupervisor** described later on. The external resource *FlightManual* and the internal resource *Expert* are

required to perform the task *Follow up Certification File*. To meet the second high-level goal “Regulation be updated”, it is required to perform two tasks:

- *Review Mission*, for which the internal resource *Certification File* and the external resource *Report* are used.
- *Define Operational Restriction*, for which the same external resource *Report* is used such as the internal resource *Environment Data*.

- **UAV:**

This third actor **UAV** represents the UAV system and its supplier (that can also be the designer of the UAV). The high-level goal of this actor is “Survive to the Mission be Maintained” which means it is expected that the UAV would not be damaged during its operation. The goal related to the supplier’s role is the sub-goal “Certification File be Fed.” This goal is satisfied by performing the task *Share Design Data* with the help of the external resource *Specification*. The other sub-goals “Communication be Ensured”, “State be Checked”, “Procedure be Followed up”, and “Flight be Ensured” concern the UAV system. Achieving all these goals shall allow the UAV to conduct its operation without incident and to carry out all the actions requested from itself. To do this, the UAV has several internal resources :

- *Telemetry*: to communicate with the *GroundStation*, an internal resource of the actor **MissionSupervisor**.
- *Loudspeaker*, *Light*, and *Message*: to allow the UAV to indicate its presence and inform the actor **MissionSupervisor** of its state (e.g., report of component failure).

The UAV has also one external resource entitled *Payload*. The last resource is considered external to the UAV because it can be modified according to the missions, and above all, it is not necessary for the UAV to perform its flight, which means to take-off, to move in the air in a stable and controlled manner, and to land. These last actions are described in the GRL model of Figure 12 through the

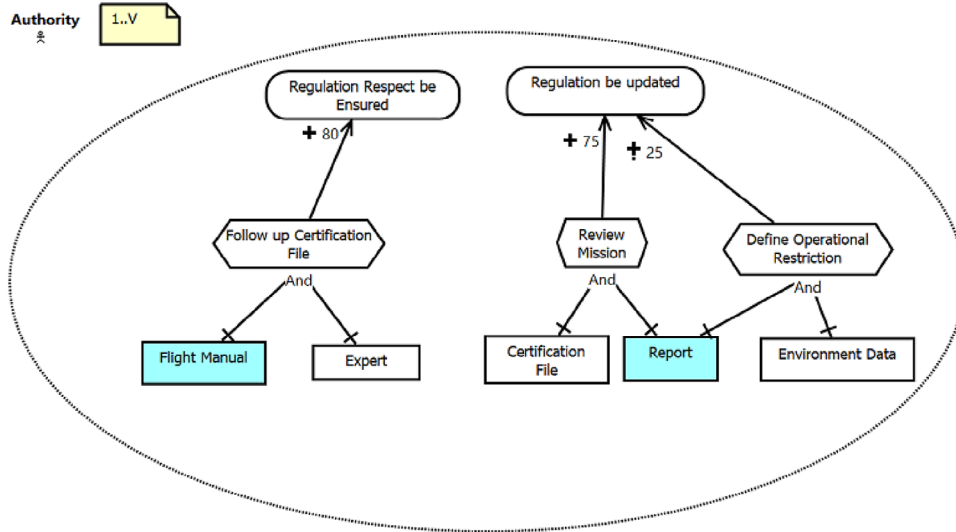


FIGURE 10 Authority GRL model. GRL, Goal-Oriented Requirements Language.

FlightSystem

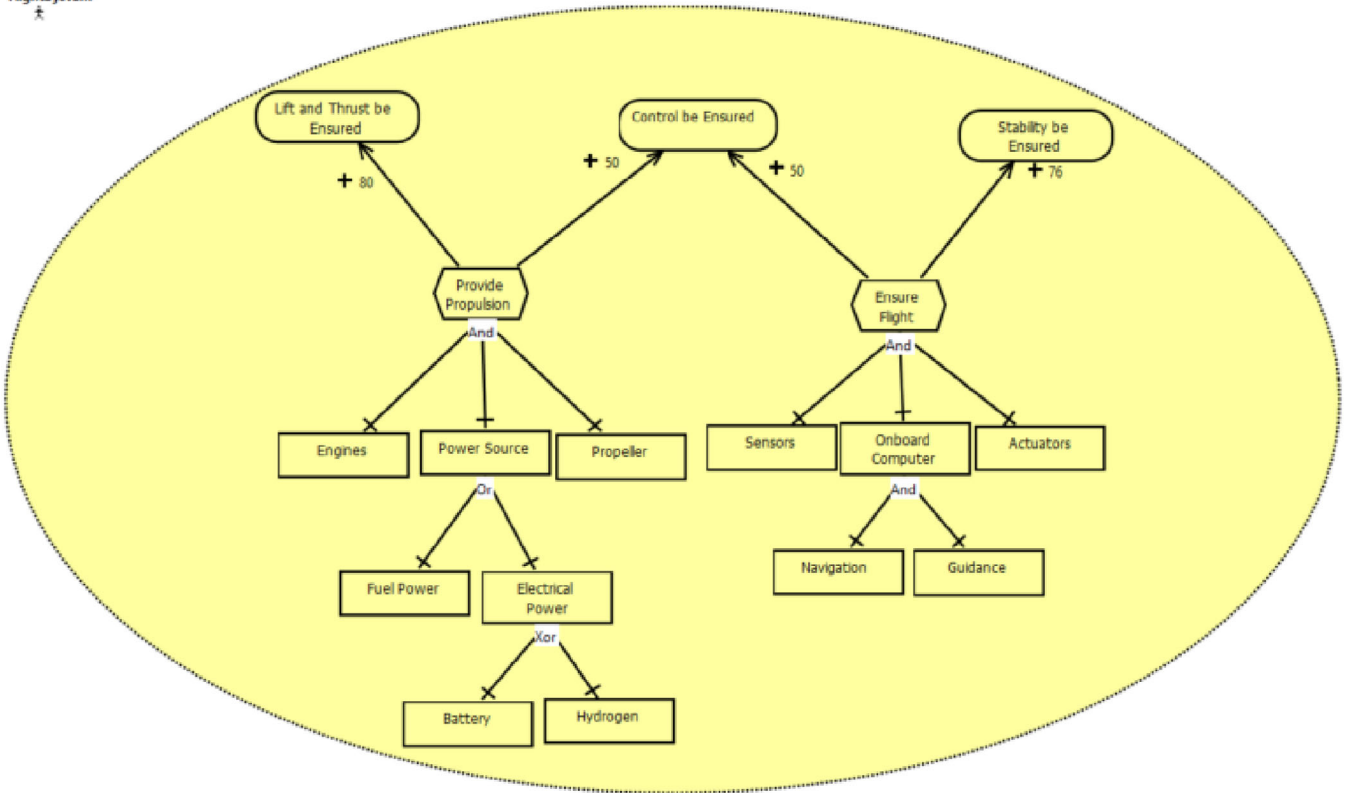


FIGURE 11 FlightSystem GRL model. GRL, Goal-Oriented Requirements Language.

tasks Perform Take-Off, Perform Cruise, and Perform landing that satisfy the sub-goal “Flight be Ensured.” This sub-goal is a *detailed goal* more precisely described in Figure 11.

The actor **FlightSystem** provides the details of the sub-goal “Flight be Ensured.” In this additional actor there are three high-level goals: “Lift and Thrust be Ensured”, “Control be Ensured”, and “Stability be

Ensured.” These three goals are met by performing the tasks Provide Propulsion and Ensure Flight. Using the extension *detailed goal* also allows us to describe more precisely the UAV components through the internal resources and without overloading the initial GRL model which can therefore remain at a high level description. In the actor **FlightSystem** it is specified that the UAV propulsion

is performed with the internal resources *Engines*, *Power Source*, and *Propeller*, where *Power Source* is decomposed into *Fuel Power* or *Electrical Power* that can be powered with *Battery* or exclusively with *Hydrogen*. Regarding the UAV management, three internal resources are required: *Sensors*, *Actuators*, and *Onboard Computer* that can be decomposed into *Navigation* and *Guidance*.

- **MissionSupervisor:**

Finally, the fourth actor named **MissionSupervisor** has a major role in the GRL model of UAV mission that we propose since it makes the link between the three other actors **Client**, **Authority**, and **UAV** as one can see in Figure 12. Its high-level goal is “*Mission be Supervised*” which is satisfied by five sub-goals “*Certification File be Fulfilled*”, “*Mission Action be Performed*”, “*Waypoint be Generated*”, “*Waypoint be Reached*”, “*Data be Assessed*” attached to it. In other words, this actor has multiple roles to play in order to provide the waypoints that the UAV should join and the trajectory that it should follow, but also contribute to the certification file to provide to the authorities, and finally to analyze several data during the operation of the UAV to guarantee the mission success. This actor **MissionSupervisor** contains two external resources that are *Regulation* and *CONOPS*. These resources are helpful to complete the certification file, and thus, participate in the satisfaction of the sub-goal entitled “*Certification File be Fulfilled*.” **MissionSupervisor** also contains internal resources, the major one is the *Ground Station* which is subdivided into *Maintenance Tool*, *Telecommunication*, *UAV Operator*. All these internal resources exchange with the UAV either to guide it during its operation, or to repair it, or to move it, (e.g., from its hangar to its work zone).

The four actors, previously described, are interconnected. They communicate and exchange data during the mission. *Contributions* between **UAV** and **MissionSupervisor** visually represent these exchanges which are required to succeed in a mission. An excerpt of the entire GRL model of a UAV mission is given in Figure 12. Only the details of the **MissionSupervisor** and **UAV** actors are given in Figure 12.

### 4.3 | GRL model analysis

As already discussed in Section 3 part 3.4, jUCMNav allows to analyze GRL models using different algorithms that propagate initial satisfaction values given to some intentional elements of an actor through the whole GRL model. At the end of the analysis, all the intentional elements of the GRL model should have a satisfaction value, filled in by the user, or computed by the tool. According to the results obtained, the user can decide to modify the architecture of the GRL model in order to improve the satisfaction value of one or several intentional elements of the model.

To analyze the GRL model of a UAV mission presented in Figure 12 we used a bottom-up approach which means we give an initial satisfaction value to the intentional elements placed at the bottom of the actors. In our first evaluation strategy for the aforementioned

**TABLE 2** Initial satisfaction values of the intentional elements of MissionSupervisor.

MissionSupervisor	
Intentional element title	Initial satisfaction value
Regulation	88
CONOPS	68
Apply Procedure	93
Maintenance Tool	28
Telecommunication	82
UAV Operator	63

**TABLE 3** Initial satisfaction values of the intentional elements of UAV.

UAV	
Intentional element title	Initial satisfaction value
Specification	90
Telemetry	8
Payload	75
Loudspeaker	38
Light	52
Message	38

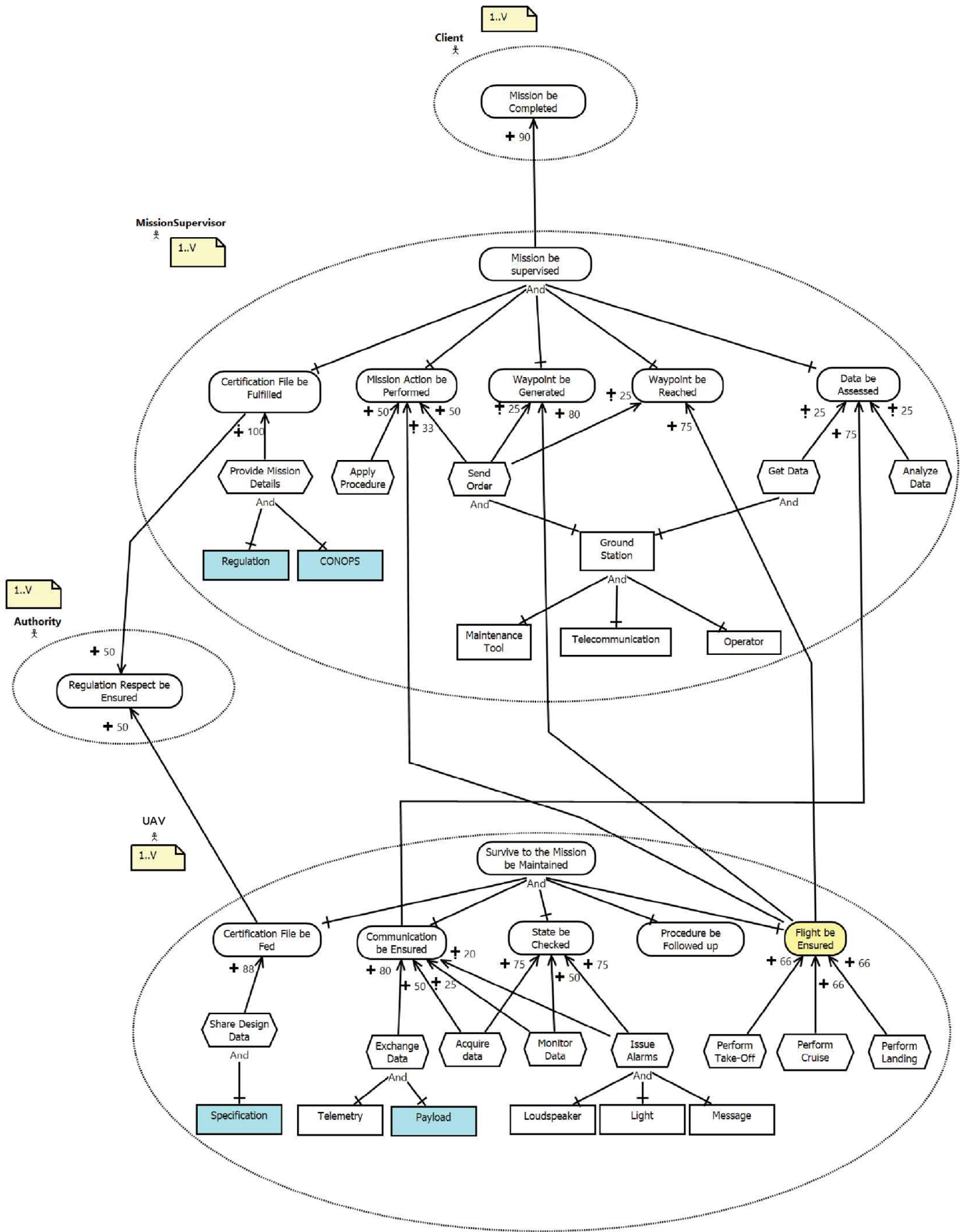
case study, the most influential actor **MissionSupervisor** is used to determine the mission to be carried out. Its intentional elements are assigned satisfaction values to ensure mission success. Values greater than 50 indicate elements in good condition of functioning or meeting the expected functional criteria. Only **Maintenance Tool** is in a condition reflecting either a design deficiency or a defect, with a score of 28. For the **UAV** actor, the most unsuccessful element is the **Telemetry** equipment. In this example, because of the early stages of UAV design, it is obvious to have such low values. However, since we are mainly interested in defining the needs of stakeholders, the modifications concern actors other than the **UAV** itself. Thus, an initial interpretation of a scenario reveals a low level of satisfaction on the part of the **Client**. To correct this, we have made up our mind to a few modifications of **MissionSupervisor** by acting on the weight of contribution links and satisfaction level values.

The initial satisfaction values ([0; 100]) attributed to the intentional elements of the actor **UAV** and the actor **MissionSupervisor** are given in Table 2 and Table 3.

The Table 4 presents the value of three *contributions* used to linked intentional elements.

With the initial satisfaction values given in Table 2 and Table 3 the *high-level* goal “*Mission be Completed*” contained in the actor **Client** obtained a satisfaction value equal to 29. This result is too low to be acceptable. For this reason we have modified some initial satisfaction values of the actor **MissionSupervisor**. They are presented in Table 6.

These changes mean in the real world that we review the ConOps which allow us to increase the satisfaction value of the intentional



**FIGURE 12** GRL model of a UAV mission (overview). GRL, Goal-Oriented Requirements Language; UAV, Unmanned Aerial Vehicle.



**TABLE 4** Initial contributions values.

From		To		Contribution value
Intentional element title	Contained in actor	Intentional element title	Contained in actor	
Provide Mission Details	MissionSupervisor	Certification File be Fulfilled	MissionSupervisor	75
Flight be Ensured	UAV	Waypoint be Reached	MissionSupervisor	25
Communication be Ensured	UAV	Data be Assessed	MissionSupervisor	25

**TABLE 5** Final contributions values.

From		To		Contribution value
Intentional element title	Contained in actor	Intentional element title	Contained in actor	
Provide Mission Details	MissionSupervisor	Certification File be Fulfilled	Mission Supervisor	100
Flight be Ensured	UAV	Waypoint be Reached	MissionSupervisor	75
Communication be Ensured	UAV	Data be Assessed	MissionSupervisor	75

**TABLE 6** Final satisfaction values of the intentional elements of MissionSupervisor.

MissionSupervisor	
Intentional element title	Initial satisfaction value
Regulation	88
CONOPS	85
Apply Procedure	93
Maintenance Tool	75
Telecommunication	82
UAV Operator	63

element CONOPS. And, we assume that more tools and spare parts for the UAV are brought to the work zone so that more small repairs can be made (e.g., change a propeller) which increases chances of the UAV being operational to execute its flight. In addition to that, we have changed the *contribution* values initially presented in Table 4 for those provided in Table 5.

By doing this, the satisfaction value of the *sub-goal* "Mission be Completed" of the actor *Client* has increased from 29 to 73. The analysis of the model containing the intentional element satisfaction values given in Table 3, Table 6, and in Table 5 is presented in Figure 13.

## 5 | CONCLUSIONS

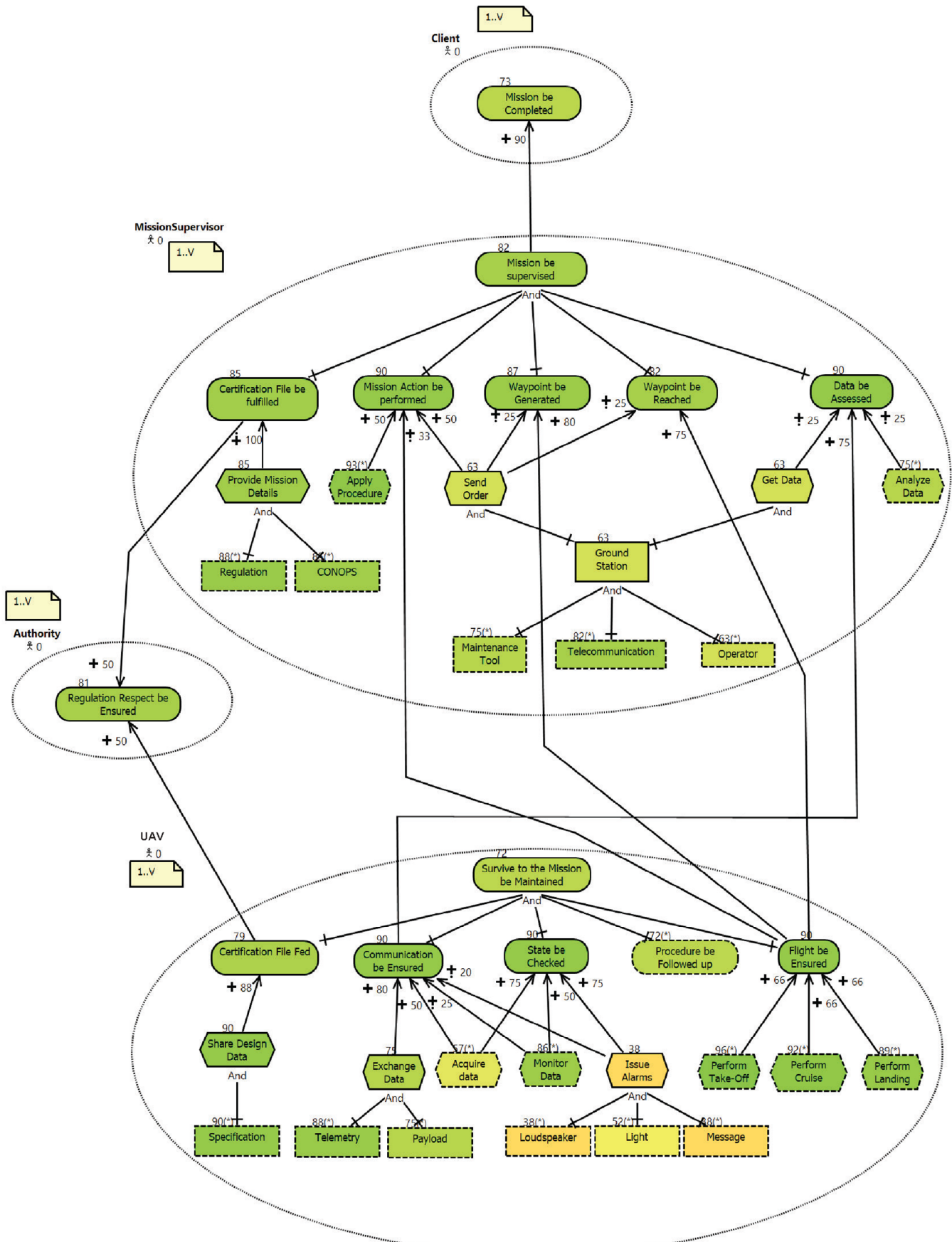
UAV design requires application of mission engineering upstream systems engineering, either to assess there is a UAV system that may accomplish a new mission, or to specify a new UAV according to the mission. To achieve that goal, the authors of the current paper support the use of Model-Based Mission Engineering.

Model-Based Mission Engineering can be addressed in terms of language, tools, and methods. In this paper we focus on the language and methods.

First, we promote GRL as a good option to capture a mission description in a way that make this description: more usable, in particular by facilitating communication between stakeholders; and reusable, being totally independent from the architecture of a system that would have to accomplish it. Indeed, GRL offers the opportunity to first capture the high level goals, expressed by the stakeholders, that keep track of the "Why" and then decompose them with alternative ways to satisfy these high level goals with more concrete ones, hence moving to the "How".

We propose some extensions to improve the resulting model readability, mainly the goal details extension. Indeed, we could observe that, in the description we introduced to illustrate our approach, it was impossible to have the whole diagram of the UAV actor holding in a page and being readable. Hence we consider helpful to keep a readable high level view of an actor, along with a set of more detailed views, easily linked to the previous one through the *goal details* extension. We propose extensions to ease the development of the mission description by introducing the degraded mode exploration during the modeling process, which enables to visualize the degradation impact without launching any analysis. We also propose some extensions to pave the way for evaluating the matching of this mission description with an architectural (structural and behavioral) description of a system, actor occurrences and differentiation between internal or external resources. Indeed in an architectural description of a system, concepts such as multiplicity, clear distinction between composition and aggregation will appear. Having already captured this information from the goal modeling step - when available - will offer to check the consistency of the architectural description with stakeholders expectations.

We then apply our proposition to a UAV system context on an example. This application illustrates a certain applicability of the developed principles. It also introduces a starting point, if not a generic pattern, for building a UAV mission description. In particular, it introduces a set of actors that any UAV system should care of, their high level goals, and a possible decomposition of them into more concrete ones. The goal decomposition should, at least, lead to requirements and their associated property of being assessable.



**FIGURE 13** Analysis of the UAV mission GRL model (overview). GRL, Goal-Oriented Requirements Language; UAV, Unmanned Aerial Vehicle.

How far in the concretization should this process go is context dependant. In our case, we define more concrete goals using MDAO (Multidisciplinary Design Analysis and Optimization). This optimizes a system - here a hypothetical UAV - by playing on several interrelated parameters. In other words, more concrete goals are defined by numerical values, such as lower and upper bounds or threshold values for instance, feeding or provided by multidisciplinary analysis and optimization models.<sup>2</sup>

The proposition introduced in this paper would benefit from being applied to other case studies. The current paper addresses a system containing one UAV at a time. How to extend the proposed approach to a UAV swarm<sup>43</sup> remains an open question.

The links between the description of the mission we promote here, and the architectural design of a UAV system, need to be formalized in order to improve reasoning on the analysis and the implementation of the mission.

In case of promising results, providing associated supporting tools will also be part of the future work.

## ACKNOWLEDGEMENTS

The authors would like to thank Prof. Daniel Amyot for fruitful discussions on GRL and its associated toolkit. This work was partially supported by the Defense Innovation Agency (AID) of the French Ministry of Defense (research project CONCORDE N° 2019 65 0090004707501).

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in "Mission modeling GRL" repository at [https://gitlab.isae-supaero.fr/j.chaudemar/mission-modeling-grl/-/blob/main/Mission\\_Complet.jucm](https://gitlab.isae-supaero.fr/j.chaudemar/mission-modeling-grl/-/blob/main/Mission_Complet.jucm).

## ORCID

Jean-Charles Chaudemar  <https://orcid.org/0000-0003-0400-5266>

## REFERENCES

- Razafimahazo E, de Saqui-Sannes P, Vingerhoeds RA, Baron C, Soullax J, Mège R. Mastering complexity for indoor inspection drone development. *2021 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE; 2021:1-8.
- Aïello O, Poitou O, Chaudemar JC, de Saqui-Sannes P. Sizing a drone battery by coupling MBSE and MDAO. *11th European Congress Embedded Real Time System*. ONERA/ENAC/ISAE SUPAERO Research Federation - X-OAD Team (eXtended-Overall Aircraft Design); 2022:1-13.
- Aprville L, Saqui-Sannes PD, Vingerhoeds RA. An educational case study of using SysML and TTool for unmanned aerial vehicles design. *IEEE J Miniaturization Air Space Syst*. 2020; 1(2): 117-129.
- Silva E, Batista T, Oquendo F. A mission-oriented approach for designing system-of-systems. *2015 10th System of Systems Engineering Conference (SoSE)*. IEEE; 2015:346-351.
- Silva E, Batista T, Cavalcante E. A mission-oriented tool for system-of-systems modeling. *2015 IEEE/ACM 3rd International Workshop on Software Engineering for Systems-of-Systems*. IEEE; 2015:31-36.
- Amyot D, Ghanavati S, Horkoff J, Mussbacher G, Peyton L, Yu E. Evaluating goal models within the goal-oriented requirement language. *Int J Intell Syst*. 2010;25(8):841-877. <https://onlinelibrary.wiley.com/doi/10.1002/int.20433>
- Mussbacher G, Amyot D. Goal and scenario modeling, analysis, and transformation with jUCMNav. *2009 31st International Conference on Software Engineering - Companion Volume*. IEEE; 2009:431-432.
- Tsai W-T, Spencer M, Wu C, Winn C, Kellogg K. SeaWinds on QuikSCAT: sensor description and mission overview. *IGARSS 2000. IEEE 2000 International Geoscience and Remote Sensing Symposium. Taking the Pulse of the Planet: The Role of Remote Sensing in Managing the Environment. Proceedings (Cat. No. 00CH37120)*. Vol 3. IEEE; 2000:1021-1023.
- Conard S, Warren J, Barnouin-Jha O, et al. Contour forward imager on the comet nucleus tour mission. In: Hoover, R, Rozanov, A, eds. *Conference on Instruments, Methods and Missions for Astrobiology VII, San Diego, CA, AUG 03-04, 2003*. Vol 5163. SPIE; 2004:72-83.
- Côme H, Benoist J, Stefanov L, et al. Galileo 5 and 6 leap or how to handle and recover two of the most feared failures occurring simultaneously. *14th International Conference on Space Operations*. ARC; 2016:2506.
- Oberg J. Why the mars probe went off course [accident investigation]. *IEEE Spectr*. 1999; 36(12): 34-39.
- Komite Nasional Keselamatan Transportasi. Aircraft Accident Investigation Report: PT. Lion Mentari Airlines, Boeing 737-8 (MAX): PK-LQP, Tanjung Karawang, West Java, Republic of Indonesia, 29 October 2018. Komite Nasional Keselamatan Transportasi; 2019.
- Aircraft Accident Investigation Bureau. *Aircraft accident investigation preliminary report. Ethiopian Airlines Group. B737-8 (MAX)*. Federal Democratic Republic of Ethiopia, Ministry of Transport, Addis Ababa, Ethiopia, Tech. Rep. AI-01/19; 2019.
- Favarò, FM, Nader N, Eurich SO, Tripp M, Varadaraju N. Examining accident reports involving autonomous vehicles in california. *PLoS One* 2017;12(9):e0184 952.
- Sousa-Poza A. Mission engineering. *Int J Syst Syst Eng*. 2015;6(3):161-185.
- ISO. *ISO/IEC/IEEE 15288:2015 Systems and software engineering - System life cycle processes* 2015. <https://www.iso.org/standard/63711.html>
- Haskins C, Forsberg K, Krueger M. *Systems Engineering Handbook - A Guide for System Life Cycle Processes and Activities*. Wiley; 2007.
- Kapurch SJ. *NASA systems engineering handbook*. Diane Publishing; 2010.
- Group ISHW. *In cose systems engineering handbook v. 3.2. 2. Technical Report, INCOSE-TP-2003-002-03.2. 2*. October 2011.
- Lee M, Weidner RJ, Lu W. Design-based mission operation. *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*. Vol 7. IEEE; 2001:7-3441.
- Fusaro R, Viola N, Fenoglio F, Santoro F. Conceptual design of a crewed reusable space transportation system aimed at parabolic flights: stakeholder analysis, mission concept selection, and spacecraft architecture definition. *CEAS Space J*. 2017;9(1):5-34. <http://link.springer.com/10.1007/s12567-016-0131-7>
- Beery P, Irwin T, Paulo E, Pollman A, Porter W, Gillespie S. Bridging joint operations and engineering management through an operational mission architecture framework. *Eng Manage J*. 2021:1-10.
- Cherfa I, Belloir N, Sadou S, Fleurquin R, Bennouar D. Systems of systems: from mission definition to architecture description. *Sys Eng*. 2019; 11: 22.
- Khalifa AS. Mission, purpose, and ambition: redefining the mission statement. *J Strategy Manag*. 2012; 5: 236-251.
- Wittmann F, Hufnagl M, Roth F, Yorulmaz M, Lindner R. From mission definition to implementation: conceptualizing mission-oriented policies as a multi-stage translation process. *Technical Report*. Fraunhofer ISI Discussion Papers-Innovation Systems and Policy Analysis; 2021.
- US Department of Defense. The DODAF architecture framework version 2.02.2010. Accessed February 5, 2024. <https://dodcio.defense.gov/Library/DoD-Architecture-Framework/>

27. Beery P, Paulo E. Application of model-based systems engineering concepts to support mission engineering. *Systems* 2019;7(3):44. <https://www.mdpi.com/2079-8954/7/3/44>
28. He X, Chaudemar JC, Huang J, Defaÿ, F. Fault tolerant control of a quadrotor based on parameter estimation techniques and use of a reconfigurable pid controller. *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE; 2016:188-193.
29. Abi Badra A, Aiello O, Chaudemar JC. Applying a model-based systems engineering approach to model an unmanned aerial vehicle mission. *2023 IEEE International Systems Conference (SYSCON)*. IEEE; 2023: 1-14.
30. Chaudemar JC, de Saqui-Sannes P. MBSE and MDAO for early validation of design decisions: a bibliography survey. In: *2021 IEEE International Systems Conference (SysCon)*. IEEE; 2021.
31. Mordecai Y, James NK, Crawley EF. Object-process model-based operational viewpoint specification for aerospace architectures. In: *2020 IEEE Aerospace Conference*. IEEE; 2020:1-15.
32. Silva E, Batista T. Formal modeling systems-of-systems missions with mkaos. In: *SAC '18: Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM; 2018:1674-1679.
33. Quilbeuf J, Cavalcante E, Traonouez LM, Oquendo F, Batista T, Legay A. A logic for the statistical model checking of dynamic software architectures. In: *Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications: 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece*. Vol 9952. Springer; 2016:806-820.
34. International Telecommunication Union. *User Requirements Notation (URN) - Language definition*. October 2018. <https://www.itu.int/rec/T-REC-Z.151-201810-l/en>
35. Darimont R, Delor E, Massonet P, van Lamsweerde A. Grail/kaos: an environment for goal-driven requirements engineering. In: *Proceedings of the 19th international conference on Software engineering*. IEEE; 1997:612-613.
36. Dalpiaz F, Franch X, Horkoff J. *iStar 2.0 Language Guide*. 2016. <http://arxiv.org/abs/1605.07767>
37. Bertolini D, Perini A, Susi A, Mouratidis H. The TROPOS visual modeling language: a MOF 1.4 compliant metamodel. *Agent-oriented software engineering technical forum*. Citeseer: Ljubljana, Slovenia; 2005:1-9.
38. Amyot D, Mussbacher G, Ghanavati S, Kealey J. GRL modeling and analysis with jUCMNav. In: *CEUR Proceedings of the 5th International i\* Workshop (iStar 2011)*. 2011:160-162.
39. Amyot D, Akhigbe O, Baslyman M, et al. Combining goal modelling with business process modelling: Two decades of experience with the user requirements notation standard. *EMISAJ* 2022;17:1-37.
40. Amyot D, Horkoff J, Gross D, Mussbacher G. A lightweight GRL profile for i\* modeling. *Advances in Conceptual Modeling-Challenging Perspectives: ER 2009 Workshops CoMoL, ETheCoM, FP-UML, MOST-ONISW, QoS, RIGiM, SeCoGIS, Gramado, Brazil, November 9-12, 2009. Proceedings* 28. Springer; 2009:254-264.
41. Neace K, Roncace R, Fomin P. Goal model analysis of autonomy requirements for unmanned Aircraft Systems. *Requir Eng*. 2018;23(4):509-555. <http://link.springer.com/10.1007/s00766-017-0278-6>
42. OMG. *Unified Modeling Language Version 2.5*. <https://www.omg.org/spec/UML/2.5.1> Dec 2017.
43. Aloui K, Hammadi M, Guizani A, Haddar M, Soriano T. A new SysML model for UAV swarm modeling: UavSwarmML. *2022 IEEE International Systems Conference (SysCon)*. IEEE; 2022:1-8.

**How to cite this article:** Chaudemar J-C, Aiello O, de Saqui-Sannes P, Poitou O. Mission-based design of UAVs. *Systems Engineering*. 2024;1-19. <https://doi.org/10.1002/sys.21754>