Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Command and Control Agility: a Software Product Line Approach

Junier Caminha Amorim

Tese apresentada como requisito parcial para
conclusão do Doutorado em Informática

Orientador
Prof. Dr. Vander Ramos Alves

Brasília
2022

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

Amorim, Junier Caminha
AA524c Command and Control Agility: a Software Product Line Approach / Junier Caminha Amorim; orientador Vander Ramos Alves. -- Brasília, 2022.
173 p.

Tese (Doutorado - Doutorado em Informática) -- Universidade de Brasília, 2022.

1. Command and Control. 2. Software Product Lines. 3. Command and Control Agility. 4. Channel Systems. I. Alves, Vander Ramos, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Command and Control Agility: a Software Product Line Approach

Junier Caminha Amorim

Tese apresentada como requisito parcial para
conclusão do Doutorado em Informática

Prof. Dr. Vander Ramos Alves (Orientador)
CIC/UnB

Prof. Dr. Dante Augusto Couto Barone
INF/UFRGS

Prof. Dr. Alexandre Cabral Mota
CIn/UFPE

Prof.a Dr.a Genaína Nunes Rodrigues
CIC/UnB

Prof. Dr. Ricardo Pezzuol Jacobi
Coordenador do Programa de Pós-graduação em Informática

Brasília, 2 de março de 2022

# Dedicatória

Dedico esse trabalho a:

- meus pais Flávio e Zilmar, a quem devo a minha vida; e

- minha esposa Kelen e a meus filhos, Pedro Henrique e Manuela, que me propiciaram inspiração sem limites.

# Agradecimentos

Agradeço primeiramente a Deus, sem cuja vontade nada se realiza.

Aos ilustres professores Dr. Vander Alves e Dr. Edson Pignaton, Orientador e Co-Orientador, respectivamente, por todas as orientações e motivações a mim destinadas, sempre acreditando em minhas capacidades, sempre preocupados com minha saúde mental.

Ao Exército Brasileiro, que permitiu e apoiou a execução do meu curso.

À FAP-DF (Fundação de Apoio à Pesquisa do Distrito Federal) por financiar a minha visita à Universidade de Namur (Bélgica), Edital nº 01/2018, criando as condições favoráveis para o intercâmbio e a troca de conhecimentos com os colaboradores.

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior), por apoiar a visita dos colaboradores da Universidade de Namur à Brasília, através do projeto THCS (Testing Highly Configurable Systems), Edital nº 29/2017 (CAPES/WBI).

Ao Professor Pierre-Yves, ao Dr. Gilles Perrouin, ao Dr. Moussa Amrani, e ao James Ortiz, valorosa equipe da Universidade de Namur (Bélgica), pela relevante colaboração durante esses anos.

Aos meus amigos Thiago Castro e Ricardo Fernandes, por todas as ideias, questionamentos intrigantes e compartilhamento de experiências, que colaboraram significativamente com o aprimoramento do trabalho e do processo de escrita desta tese.

Ao Cel Kohl pela motivação, experiência e ideias compartilhadas, baseadas na sua larga experiência no domínio Comando e Controle (C2) dentro do Exército Brasileiro.

A todos os Irmãos de Farda que compartilharam experiências e contribuíram com ideias e conceitos.

# Acknowledgements

I first thank God without whose will nothing happens.

To the professors Dr. Vander Alves and Dr. Edson Pignaton, my advisor and co-advisor, respectively, for all the guidance and motivation given to me, always believing in my abilities, always concerned about my mental health.

To Brazilian Army, which supported my formally.

To FAP-DF (Research Support Foundation of Distrito Federal - Brazil) for supporting my visit to University of Namur (Belgium), Public Notice No. 01/2018, creating good conditions for interactions and knowledge exchanging with the researchers.

To CAPES (Coordination for the Improvement of Higher Education Personnel) for supporting the University of Namur collaborators' visit to Brasília, based on the project THCS (Testing Highly Configurable Systems), Public Notice No. 29/2017 - CAPES/WBI.

To Prof. Dr. Pierre-Yves, Dr. Gilles Perrouin, Dr. Moussa Amrani, and James Ortiz, valued team at the University of Namur (Belgium), for the relevant collaboration and friendship during these years.

To my friends Thiago Castro and Ricardo Fernandes, for all the ideas, exciting questions and experiences sharing, which significantly contributed to the improvement of the work and the writing process of this thesis.

To Col. Kohl for his motivation, experience, and shared ideas, based on his extensive experience in the Command and Control (C2) domain within the Brazilian Army.

To all my brothers-in-army who shared experiences and contributed with ideas and concepts.

# Resumo

Comando e Controle (C2), em sua origem histórica, está relacionado à aplicação de estratégias militares clássicas onde havia um único comando centralizado e uma inflexível cadeia de comando entre os elementos que compunham as forças atuantes. C2 não é um fim em si mesmo, mas um processo cujo objetivo é otimizar a aplicação de recursos para cumprir uma missão. Entretanto, em um contexto moderno de C2, o dinamismo da missão, da equipe e do ambiente é um pressuposto necessário e, portanto, a organização da equipe para cumprir uma missão torna-se um desafio que requer constantes adaptações. Esta capacidade de adaptação às novas circunstâncias caracteriza a agilidade C2. Entretanto, o estado da arte não avalia como esta capacidade é afetada pelas escolhas da abordagem de C2, representada pelo nível de disseminação das informações, pela organização da equipe e pela capacidade de tomada de decisões. Além disso, trabalhos recentes não consideram a medição dos Atributos de Qualidade (QA), o que torna os modelos e simulações pouco aderentes à realidade das missões, onde pelo menos o custo pode ser um obstáculo à sua realização. Para abordar estas questões, aplicamos conceitos de Sistemas Auto-Adaptativos (SAS) com uma abordagem que utiliza Linhas de Produtos de Software Dinâmico (DSPL) para representar os elementos que compõem o sistema de C2 e que estão organizados em times. Baseado na configuração e coordenação, propomos dois modelos que buscam garantir a agilidade de C2. Estes modelos proporcionam a escolha da abordagem de C2, combinada com a capacidade de reconfigurar os membros da equipe a fim de garantir agilidade para lidar com as mudanças nas circunstâncias que possam ocorrer. Para avaliar os modelos propostos, realizamos um conjunto de simulações para indicar o nível de agilidade obtido pela abordagem, e aplicamos questionários aos especialistas do domínio de C2 para validar a usabilidade dos modelos e a compatibilidade com cenários realistas enfrentados pelos especialistas do domínio.

**Palavras-chave:** Linha de Produtos de Software Dinâmica, Comando e Controle, agilidade

# Abstract

Command and Control (C2), in its historical origin, is related to the application of classic military strategies where there was a single centralized command and an inflexible chain of command between the elements that composed the acting forces. C2 is not an end in itself, but a process whose goal is to optimize the application of resources in order to accomplish a mission. However, in a modern C2 context, the dynamism of the mission, the team and the environment is a necessary assumption and, thus, the organization of the team to accomplish a mission becomes a challenge requiring constant adaptations. This ability to adapt to new circumstances characterizes C2 Agility. However, the state-of-the-art does not assess how this ability is affected by the choices of C2 approach, represented by the level of information spread, by the organization of the team and by the capacity of decision making. In addition, recent works do not consider the measurement of Quality Attributes (QA), which makes the models and simulations poorly adherent to the reality of missions, where at least the cost can be an obstacle to their achievement. To address these issues, we apply concepts of Self-Adaptive Systems (SAS) with an approach using Dynamic Software Product Lines (DSPL) to represent the elements that make up the C2 System and that are organized into teams. Relying on configuration and coordination, we propose two models that seek to ensure C2 agility. These models provide for the choice of the C2 approach, combined with the ability to reconfigure the team members in order to ensure agility to face the changes in circumstances that may occur. To evaluate the proposed models, we perform a set of simulations to indicate the agility level obtained by the approach and we apply questionnaires to C2 domain experts to validate models' usability and compatibility with realistic scenarios faced by domain experts.

**Keywords:** Dynamic Software Product Line, Command and Control, agility

# Contents

# List of Figures

# List of Tables

# Acronyms

**BSP** Bulk Synchronous Parallel.

**C2** Command and Control.

**CDS** Systems Development Center - Brazilian Army.

**COTER** Ground Operations Center - Brazilian Army.

**CS** Channel System.

**DoD** US Department of Defense.

**DSPL** Dynamic Software Product Line.

**ECEME** Command and General Staff College.

**ESAO** Captains Career School.

**FM** Feature Model.

**GAP** Generalized Assignment Problem.

**IBAMA** Brazilian Institute of Environment and Renewable Natural Resources.

**IoBT** Internet of Battle Things.

**MAPE** Monitor-Analyze-Plan-Execute loop.

**MAPE-K** Monitor-Analyze-Plan-Execute-Knowledge loop.

**MAS** Multi-Agent System.

**MD** Ministry of Defense.

**MPL** Multi Product Line.

**NATO** North Atlantic Treaty Organization.

**NCW** Network Centric Warfare.

**NEC** Network Enabled Capabilities.

**OCS** Open Channel System.

**PVS** Prototype Verification System.

**QAs** Quality Attributes.

**QoS** Quality of Service.

**SAS** Self Adaptive System.

**SPL** Software Product Line.

**TS** Transition System.

**UAV** Unmanned Aerial Vehicle.

**USA** United States of America.

# Chapter 1

# Introduction

According to Alberts and Hayes [10], Command and Control (C2) is about focusing the efforts of a set of entities and resources towards the achievement of some task, objective, or goal. The entities may represent individuals or organizations, and resources involve everything manipulated by the entities, including information exchange. Originally developed in the military domain, C2 was initially based on the idea of a central command concentrating information, decision making, and power over required elements to accomplish the mission [9].

With the increasing importance and distribution of information at all operational levels, C2 processes have grown to incorporate technological tools that support and improve information exchange and decision making. This structure keeps the C2 objectives, i.e., best resources utilization to accomplish a mission according to the imposed restrictions [91]. According to Mason and Moffat [92], Stanton et al. [116], C2 relies on the information and awareness sharing, not being a simple data spread, but such an information must satisfy the rules and logical conditions to reach the right target and to cause the expected effect.

The result of these developments has made possible the application of C2 in several domains, such as Civil Defense during disaster relief operations, and financial operations managing resources to maximize results [50, 83]. Besides, C2 applications include critical domains such as nuclear weapons control [28], national mass-vaccination campaigns [72], and orchestrating different government and research organizations in the COVID-19 pandemic scenario to find a mitigation or solution to the problem and providing responses to the society [56, 111, 114].

In addition, many common situations related to our daily routine use the C2 concept to deal with problematic situations, e.g., traffic coordination by local traffic departments. As another example, some organizations coordinated a large-scale operation to deal with the oil spill on the Brazilian coastline [30]. In such a case, the Brazilian Navy, as the coordinator, together with Brazilian Institute of Environment and Renewable Natural Resources (IBAMA) and research centers, tried to figure out what caused the problem and how to mitigate the damage to the environment. Similar to this kind of coordinated operation is a set of federal organizations, coordinated by the Brazilian Army, operating to fight against the forest fire in the Amazon Region. All these operations aim to apply resources efficiently to perform the mission established, characterizing a C2 process, applied to obtain a suitable awareness and performance for the circumstance. We call this process C2 approach. Each C2 approach provides distinct awareness through a certain level of information sharing, a level of decision rights, and the interaction structure among the entities.

However, realistic scenarios are subject to change. *C2 Agility* provides the capacity for dealing with different circumstances [6]. According to Alberts and Hayes [9], this agility is a fundamental aspect to obtain success in the mission, particularly in the Information Era [9]. Based on this, C2 agility comes as an important element to the organizations to achieve the suitable awareness to face distinct scenarios. This scenario's dynamism requires a monitoring mechanism in order to keep the team focused and to provide best resource allocation. Essentially, C2 agility is a timely and effective response to context changes, and this is a valuable resource for successful mission accomplishment. The context is composed of the mission, the environment where the C2 system runs, and the entities that perform the mission.

In particular, C2 agility provides the capacity to adapt the teams and members for a new situation, or even to change the structure and procedures to permit the adaptation to a new condition incorporating additional elements [6]. Additional resources provided to the entities are often sufficient to cope with these changes. However, some scenarios require changes in the collaboration between entities to improve the awareness [95]. In addition, the surrounding environment can increase the scenario complexity with constraints that limit the C2 approach change. It makes the awareness attainment more difficult, or even unfeasible.

The remainder of this chapter is organized as follows. Section 1.1 shows the problems faced by the domain referenced by this thesis, highlighting the corresponding motivations. Section 1.2 details the limits addressed by this study and describes the objectives to be achieved. Section 1.3 gives an overall description of the research methodology applied by this work. Section 1.4 lists the main contributions of this thesis. Last, Section 1.5 describes how this work is divided and organized.

## 1.1 Problem and Motivation

The complexity of existing scenarios in different C2 domains stems mostly from the inherent dynamism present in all of them because of potential changes in circumstance or context. For instance, the sudden lack of entities or the increased risk of collapse or flooding are some problems that characterize circumstance changes of C2 applied in the Civil Defense domain. Such changes can affect the mission, the environment, or the entities, and they characterize the context dynamism and increase complexity. Based on this, C2 agility is a required capability to deal with these dynamic scenarios.

Nevertheless, there is a lack of evidence on how to provide C2 agility. Many of these adaptations are made by trial and error, without the possibility of trying to systematize such behavior given the infinite possibilities involved. Indeed, state-of-the-art and state-of-the-practice approaches present limitations when they explore context changes. Those that do so focus on randomized network-level reconfiguration rather than on entity-level reconfiguration or changes of mission granularity. Thus, they do not deal adequately with the adaptation of the mission accomplishment, which at the end does not characterize the C2 agility definition [5–8, 10, 121]. On the other hand, dynamism abounds in realistic scenarios, which are under frequent circumstance changes [121]. This aspect highlights C2 agility's relevance to deal with dynamic context to provide a capability of adaptation.

We identify a lack of studies on C2 agility, particularly on how to provide C2 Agility due to all the involved complexity and heterogeneity in the C2 process. We then highlight the main problem that guides this thesis:

| There is no evidence on how to provide C2 Agility. |
|---|

To achieve C2 agility, it is necessary to monitor and to share information about the changes in the context. Context modifications must be monitored in order to maintain the team's focus on the mission [6]. Based on all these characteristics, there is a lack of ways to model the elements that make up a C2 scenario to represent adequately the agility in dealing with context changes. Particularly in the simulated environment, the representation of the various entities and their interactions, common in realistic scenarios, do not sufficiently represent their ability to respond to new circumstances. This limitation occurs because of entities' self-configuration missing or not collaborative with each other.

Realistic scenarios with application of C2 concepts are naturally complex and involve many dependable variables [93]. In particular, the cost variable is significant, and it is not used only as a metric for assessing quality, but also mission feasibility and meeting requirements. Similarly, costs are related to the changes in the system to deal with new circumstances. For example, in the military domain, the capacity to incorporate extra elements, to adapt the teams and members to a new situation, or even to change the structure and procedures to permit the adaptation to a new condition is a basic requirement to perform a mission successfully [7].

In addition, the more recent studies on C2 Agility do not consider quality attributes nor any cost values during analysis and simulations [6, 8, 28, 83]. Such works explored agility by analyzing results from simulation and retrospectively from realistic operations, where C2 approaches applied to obtain results. The assessment focused only on mission accomplishment. Furthermore, the applied models consider only static aspects and they show limitations to represent the dynamic dimension that exists in C2 context problems. These characteristics highlight an important gap when compared with real scenarios.

Indeed, to provide C2 agility in dynamic scenarios in which members are performing a mission, it is necessary to guarantee efficient management of available resources to deal with context changes within the expected outcomes. The ability of the members to adapt themselves according to information received or obtained from the environment via sensors permits the system to keep running, even under new circumstances. The challenge is adapting and maintaining quality attributes within acceptable levels.

## 1.2 Scope and Objectives

This study defines C2 System as a structure composed of entities, e.g., software, robots, Unmanned Aerial Vehicle (UAV), or personnel, operating a C2 Approach to perform a set of tasks so-called mission. This C2 Approach provides the coordination among the entities to obtain awareness in mission execution. An underlying assumption is that the system is always operating a C2 Approach, so there is always some level of coordination strategy among the entities: from the most extreme case, with segregated entities and no communication between them, to the most connected scenario, where all entities communicate with each other to exchange information about the context and the mission, we always have a C2 Approach operated.

Based on this, we choose the military domain to validate our proposal and to define terms and processes. Such a choice was motivated because of easy access to military domain experts to collect data and information. In addition, military operations are naturally under dynamic conditions that require different C2 approaches according to the circumstances. Such C2 approach changing occurs according to the roles distribution, being compatible with the military domain rules [5] [6].

In addition, our study considers an external entity, out of the system and not represented in the models, that sends the mission with the tasks to be performed. Similarly, it receives results in execution report from the entities in operation. This outsider does not alter the autonomy level of the entities. Besides, we do not insert detailed information about the network topology in the proposed models. We consider the key characteristics of each C2 Approach to define a suitable network topology to represent it. Based on this, we encapsulate all network details to give flexibility in different scenarios.

Limited by the scope elements previously listed, this work aims to propose two models capable of identifying elements, their interactions, and actions necessary to provide C2 agility. In addition, these elements can be beyond technology components, but this work focuses on homogeneous forces to reduce complexity during simulations [50]. Based on this, we use simulated drones to represent the elements in our C2 system. However, the proposed models can represent complexity structures with distinct elements. With such an assumption, we can list the following partial objectives of this work:

- Propose a hierarchical static model to represent a C2 system;
- Propose a dynamic model that represents the C2 system elements and their behavior based on roles;
- Perform the proposal validation through simulations;
- Assess ease of use and usability level of proposed models conducting a survey with military domain experts; and
- Validate definitions and relevance of our study in the military domain conducting a survey.

In summary, the focus of this work lies in the representation of a C2 system and its elements. Such a representation allows us to identify requirements to provide C2 agility in order to deal with context changes.

## 1.3 Research Strategy

The present work follows an action research procedure [42] combining different research methods. Based on this, we use simulation and survey to collect evidence of our proposal validity and compatibility with military domain requirements. Figure 1.1 shows the major steps followed by this thesis to address the problem previously presented. We highlight these steps during the following detailing.

We begin our research with a literature review to identify the state-of-the-art C2 agility principles and application (step **Literature Review**). This review pinpoints a lack of studies about such a subject and thus motivates our study. Based on this first step, we perform a simulation based on the solution presented by Schwarzrock et al. [108] to assess their proposed allocation algorithm in a dynamic context, and to improve it

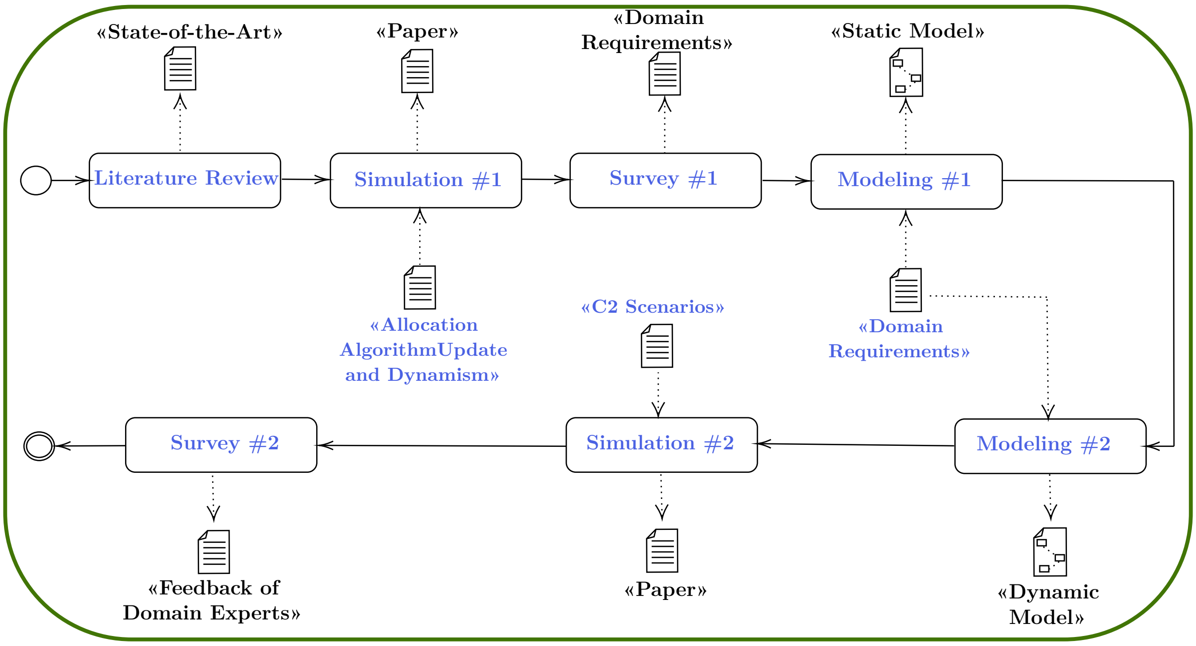



**Figure 1.1:** This work's road map with the main milestones.

in such context (step **Simulation #1**). Such a dynamism, characterized by changes in the entities, the mission, or even in the environment, is the key element to insert complexity in scenarios. This experiment indicates the relevance and effects of dynamism on the simulation environment. In addition, we published the results [13] and highlighted the opportunity to apply C2 concepts to optimize the team performance during mission accomplishment in a dynamic context.

To validate the relevance identified with the first simulation performed, we conducted a survey to present the scenarios created in the simulator and the definitions used based on C2 theory (step **Survey #1**). We submit this survey to military domain experts and their feedback confirms the relevance of dynamism in realistic scenarios and the effects of C2 concepts application. Such an interaction with military domain brings a set of requirements and realistic scenarios faced by them with C2 concepts application. These requirements are important elements to guide our proposal models design. In addition, this survey is used to collect ideas to extend the simulation scenarios by including important existing elements and features in real scenarios in the military domain.

Next, based on the feedback from the first survey, characterizing the domain requirements based on realistic situations, we propose static and dynamic models to improve the C2 system representation. We combine all information collected with the theory to propose a strategy to deal with context changes with C2 agility. The static model, defined by a metamodel, brings information to leverage scenario's elements complexity, being able to represent complex structures such as team of teams (step **Modeling #1**). The dynamic model represents the entities' behavior and their interaction. However, in order to simplify its representation, we omit cascaded structures that are represented in the static model. We use the domain feedback obtained with the Survey #1 to identify the main elements of this model based on channel system theory (step **Modeling #2**). The idea is to use both models complementarily.

Aiming at the proposed model's validation, we perform a second simulation (step **Simulation #2**). We implement the simulation with distinct scenarios following the models' roles and properties. The implementation uses an agent-based system based on the experience obtained in the first simulation. The simulation results show evidence of C2 agility with the capability of dealing with dynamic context.

Finally, to extend the validation of our proposal because of inherent simulation's limitation, we submit the proposed static and dynamic models to domain experts analysis. We perform this evaluation with a survey to assess the ease of use and the usability of each model (step **Survey #2**).

## 1.4 Contributions

The strategy described in Section 1.3 looks for combining Command and Control (C2) and software engineering to motivate our study. Overall, this thesis makes the following contributions:

- A C2 System static model to provide a complex and hierarchical C2 System modeling;
- A C2 System dynamic model to represent elements under C2 process in order to provide C2 Agility;
- A structure that combines multiple collaborative DSPLs;
- Validation with simulations implemented according to the models; and
- Validation of relevance and usability of the proposed models, besides the used definitions, with a survey applied to experts in the military domain.

So far, the contributions have been peer-reviewed as follows:

1. Junier Caminha Amorim, Vander Alves, Edison Pignaton de Freitas: Assessing a swarm-GAP based solution for the task allocation problem in dynamic scenarios. Expert Syst. Appl. 152: 113437 (2020) (DOI: 10.1016/j.eswa.2020.113437);

2. Junier Caminha Amorima, Eduardo Lemos Rocha, Luigi Minardi, Vander Alves, Edison Pignaton de Freitas, Thiago Castro, Moussa Amrani, James Ortiz, Pierre-Yves Schobbens, Gilles Perrouin: Providing Command and Control Agility: a Software Product Line Approach ( Manuscript No. ESWA-D-20-06351R1 under review, initial submission on Nov. 24, 2020; major revision submitted on Oct. 28, 2021).

The first paper collects evidence of dynamism's impact on a team performing a reconnaissance mission through an exploratory study. We submit such a scenario to changes with the entities, e.g., onboard sensors' issues, or autonomy limitations. It is useful to identify the relevance of entities' adaptation to deal with context changes. We submit the paper's results to military domain experts analysis to evaluate the relevance level and compatibility with realistic scenarios.

The second one implements the dynamic model proposed by this work to evaluate its compatibility with military domain concepts and its usability to model a C2 system. It provides C2 agility following the roles defined by the proposed model and their interactions. In addition, it opens the opportunity to extend the study and simulations in order to support complex structures. Similar to the exploratory study previously described, we extend the validation submitting the results to the domain experts through a survey.

In summary, the main contribution of this work is to provide a complementary pair of models able to identify the key elements of a C2 system. These models may identify the involved entities' behavior in order to provide C2 agility. The communication between the elements acting in the mission occurs under rules and according to the roles performed by them.

## 1.5 Outline

The remainder of this work is organized as follows:

- Chapter 2 presents a required background about Command and Control, Software Product Lines, and Channel Systems, for a proper understanding of the issues addressed by this study;

- Chapter 3 describes the state-of-the-art and the problem statement with a motivating example, in addition to the assumptions considered by this work;

- Chapter 4 shows the proposed static model using a metamodel to improve the structure information addressing complex and hybrid teams. In addition, it shows the dynamic model's details based on Channel System concept to model the C2 system structure and its entities behavior;

- Chapter 5 shows the proposed models assessment through simulations and surveys;
- Chapter 6 presents the related work and highlights the topics that motivate our study; and
- Chapter 7 presents concluding remarks reviewing the problem statement, limitations of this work, and highlights for future work opportunities.

# Chapter 2

# Background

This chapter presents the fundamental concepts and definitions necessary to understand the proposals presented in this work. Section 2.1 introduces historical aspects, and the elements related to Command and Control (C2). Section 2.2 overviews Software Product Line (SPL) and its strategy of modeling and implementation. Last, Section 2.3 presents Channel System and the modeling guidelines.

## 2.1 Command and Control

Command and Control (C2) originally referred to the classical military strategies applied in the XIX and XX centuries, with a central commander and a rigid hierarchical chain of command [10]. The idea of keeping a total control over a group of men in combat was the guideline for this concept for long time. The US Department of Defense (DoD) [117] defines C2 as the exercise of authority and direction by a properly designated commander over assigned and attached forces in the mission's accomplishment. Such a definition suites better when we have a static scenario where changes do not occur. Here, the entities involved have predictable behavior, and they do not need to have a mechanism of constant monitoring.

The North Atlantic Treaty Organization (NATO) extended this definition to the functions of commanders, staffs, and other C2 parties in maintaining the combat readiness of their forces, preparing operations, and directing troops in performing their tasks [6]. Based on this, C2 becomes a concept always related to dynamic context, involving scenarios under changes possibility [9]. According to Mason and Moffat [92], Stanton et al. [116], C2 relies on information and awareness sharing rather than simply data broadcast. Information must follow the structure created by the rules and the logical conditions to reach the right target and to cause the right effect. This information sharing occurs on a network topology that satisfies the context requirements, including mission details and context restrictions. Such an information sharing is a key element to provide conditions to accomplish the mission.

Empowering the individuals on the organization's edge, giving all required information and leaving the decision making to them, is the new challenge to support a wider C2 definition. Alberts and Hayes [9] presents the term *Power to the Edge* to define such awareness and autonomy level. The awareness level guides modern warfare and environments, and the information sharing is crucial to define the results of the mission accomplishment. Based on this, last information sharing strategies update the C2 concept. Alberts and Hayes [9] identifies the team's power to action increment caused by the best awareness level. However, there is no unique formula for all scenarios. It required the suitable change in some C2 components to optimize the results achieved with this process.

Research works extended the C2 concept to address other domains besides the military to become applicable through the process that increases the action power through a shared awareness. This awareness can be synchronized using a specific network structure and organization to create entities interaction, so-called Network Centric Warfare (NCW) - USA - or Network Enabled Capabilities (NEC) - England [3]. NEC paradigm aims at achieving the effect with the best usage of information systems. Many scenarios figure out the C2 potential to provide results increasing with optimization of resources' application.

### 2.1.1 C2 Functions

To apply the concepts and principles related to C2, it becomes necessary to perform some functions associated with it. Alberts and Hayes [10] list the following functions in order to provide the proper management of an undertaking:

- *Establishing intent*: the mission definition;
- *Determining roles, responsibilities, and relationships*: behaviors enabling to provide individuals interaction and organization;
- *Establishing rules and constraints*: governance of the entities; and
- *Monitoring and assessing the progress*: timely response to the situations and results obtained during the mission execution.

The functions above summarize the required activities to apply C2. However, it is fundamental to consider the leadership factor. Basically, C2 depends on how the commanders or managers are good as leaders guiding the team to the mission accomplishment. The following functions can represent such a factor that is related to human aspects and behaviors:

- *Inspiring and motivating*
- *Training and education*

These last functions bring together intangible but equally important aspects in the successful mission execution. The ability and competence to lead people and situations can decide the outcomes. Such human aspects, in real and high-pressure scenarios, have a significant impact and are decisive in scenarios where there is no autonomous machine, i.e., where the human interaction and decision are required [62]. Thus, these aspects make such scenarios hard to model and to measure.

In addition, the C2 functions depend on how we apply C2 and how multiple individuals, groups, or organizations can perform them. The C2 quality level depends on how these functions are performed.

Finally, C2 application involves the allocation of existing resources and the planning of new resources' acquisition. This brings another function so-called *provisioning*. Provisioning is critical to the mission accomplishment and can become it unfeasible. Besides, there are resources changes during the mission execution, e.g., resources losing or receiving, and they require re-planning and adjustments.

### 2.1.2 Endeavor Space

A realistic scenario involves a lot of variables that become the modeling process complex and, sometimes, unfeasible. This aspect characterizes the domain uncertainty. However, some circumstances are predictable and we can identify and map them in a planning phase. Alberts et al. [6] defines the Endeavor Space as the set of all predictable circumstances and conditions of a mission. This variability describes the uncertainty level about the mission performance. Precisely, the operation's planning phase has the Endeavor Space as a fundamental element to plan circumstance changes [6] and to reduce the risks. It aims to mitigate the inherent uncertainties during mission execution.

Each region of the endeavor space represents a distinct mission situation, and it requires a suitable application of C2 concepts to deal with such a circumstance. Based on this, the endeavor space is comparable to a search space for the C2 functions shown in Section 2.1.1. Thus, a logical perspective shows that, at distinct moments, these functions have a specific region of endeavor space as domain values guiding to the mission accomplishment. The total area of endeavor space, showing the overall entities ability to act, is a signal of their agility level [7].

Alberts and Hayes [10] identifies three areas of changes that lead to unique positions in the endeavor space: self, mission, and environment. These three elements define what we call context. The self represents the entities engaged in the mission and responsible for performing it. An entity can be an organization, a single element, or even a set of organization or groups. Heterogeneous structures have entities of different types and responsibilities, making C2 implementation more complex. Mission is what the entities must accomplish. To provide a better plan and execution conditions, we divide the mission into a set of tasks. Such tasks must be, totally or partially, performed by the entities. Last, the environment involves everything that surrounds the entities and the mission execution, e.g., weather, hazard level caused by the enemy's presence, communication constraints, security constraints, or luminosity. The environment has the most complex modeling, involving many variables and even affecting the mission. The environment can become the mission or any its task unfeasible.

### 2.1.3 C2 Approach

Alberts et al. [6] defines C2 Approach or C2 Strategy as a way to perform the C2 functions (cf. Section 2.1.1) in some level in order to accomplish the mission. Basically, it represents adjustments made in C2 to provide a unique approach. These adjustments are performed in one or more C2 key dimensions presented by Alberts and Hayes [9]: Allocation of Decision Rights, Patterns of Interaction and Distribution of Information. These dimensions are orthogonal and they compose the C2 Approach Space described by Alberts and Hayes [10]. Figure 2.1 shows such dimensions graphically and a cube that represents the space with known C2 approach values to deal with the endeavor space. Each position in this space represents an existing C2 Approach, i.e., a different management process, operated during the mission accomplishment.

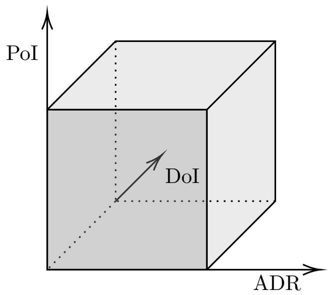



**Figure 2.1:** Dimensions of C2 Approach Space: Distribution of Information (DoI), Pattern of Interaction (PoI), and Allocation of Decision Rights (ADR) (Adapted from Alberts et al. [6]).

A fine-grained change in one or more C2 approach dimensions provides a suitable C2 approach to deal with a context change. Table 2.1 shows a description of the C2 dimensions.

Table 2.1: C2 approach dimensions description.

| **Allocation of Decision Rights (ADR)** |
|---|
| Decision rights are the capability of some individuals to make choices related to a specific topic. The allocation of this capability aims at the unity of purpose. Such an attribute defines the ADR axis in the C2 Approach Space and it represents the total capability to take decisions for other members in the structure. With the ADR more distributed, the leadership changes according to the information shared. In addition, the decision rights handled by each member result from their roles or responsibilities [6]. It is possible to identify the leadership level of each entity based on its decision autonomy level.<br>In summary, the allocation of decision rights is closely related to the roles played by each entity. Each scenario faced by the entities requires multiple roles assigned to the entities. In addition, an individual or entity can have over one role to play during mission execution, and many entities can play the same role in coordination. Based on this, such roles are fundamental to provide management and coordination among the entities. |
| **Distribution of Information (DoI)** |
| Besides defining how the entities will interact, it is necessary to define what is being shared. The type and volume of information shared impact the results of the mission. In addition, the form of transmission, e.g., protocol and technologies used, is decisive in timely obtaining awareness.<br>Particularly, this dimension defines how the information is distributed, as well as which information is involved in such communication, following the Information Exchange Requirements (IRE) to get shared awareness. Abar et al. [1] uses this dimension to define the information's dissemination level. Based on this, they apply a central repository with a high control to define which information and data are shared. |
| **Pattern of Interaction (PoI)** |
| To apply C2, the individuals or organizations must have a structure of data exchange. It is necessary to have a formalism about the way of entities interaction. Basically, the pattern of interaction defines the network topology used by the entities to exchange information and orders among them. Alberts and Hayes [10] describes four different networks, not related to hierarchy, that make a C2 approach in Information Age and found a position in C2 Approach Space, listed below: |
| *Continued on next page* |

Table 2.1 - *continued from previous page*

- Fully connected: All entities are connected to each other, enabling an interaction between all the entities;
- Random networks: Each entity has the same probability of interacting with any other, generating a spread connection structure;
- Scale-free networks: A few entities have a very large number of connections or interactions with other entities. It generates nodes of connections concentration; and
- Small World networks: Very high cluster coefficient, with a number of connections around $log(N)$, where $N$ is the total of nodes [126].

None of these topologies is suitable for all circumstances. Depending on the context complexity, it may require multiple strategies simultaneously within the structure. Alberts and Hayes [10] presents the richest network as a composition of scale-free network at high level, a small-world network at the intermediate level, and the fully connected at local level. Such a structure is created by combining teams of teams, comprising entities that are actually groups of entities that interact through another network topology.

Zhuge and Sun [130] proposed a solution to represent a Small-World network using a virtual ring topology. Such a work presents some aspects that allow a simpler approach to represent this kind of communication structure. Here, the interaction of every entity is limited to two other entities.

C2 concepts were extended to become applicable to multiple domains through the process that increases the action power through a shared awareness. We can synchronize this awareness using a specific network structure and organization to create entity interactions. The architecture that applies such a definition is so-called Network Centric Warfare (NCW) - USA - or Network Enabled Capabilities (NEC) - England [3]. Despite of performing a fine tuning in any of C2 space dimension, such a space has well-defined regions of most relevance to C2 agility studies. Based on the NEC paradigm, Alberts et al. [5, 6] present five approaches to C2 described as follows:

- *Conflicted*: there is no allocation of decision rights, no interaction among entities, and the entities have only internal information to base the decision make;

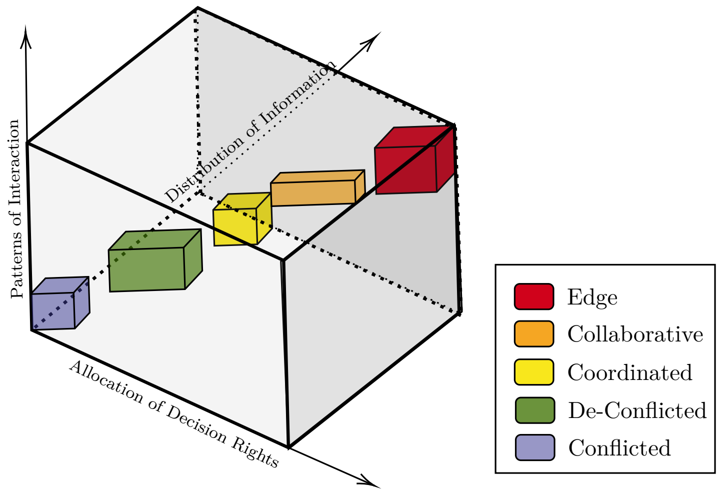



**Figure 2.2:** NEC approaches to C2, indicated by different positions within the C2 Approach Space (Adapted from Alberts et al. [6])

- *De-Conflicted*: there are constraints to the allocation of decision rights, the interactions among entities are very limited with limited information exchanges restricted to constraints and joints;
- *Coordinated*: the allocation of decision rights follows a coordination plan, the patterns of interaction are limited and focused, and there is additional information concentrated with some nodes;
- *Collaborative*: the allocation of decision rights follows a collaborative process and shared plan, the entities' interaction is broad, and there is additional information across collaborative areas and among coordinators or concentration nodes;
- *Edge*: there is a self allocation of decision rights dynamically and according with requirements, the entity interactions are unlimited, and all information is available and shared.

Figure 2.2 illustrates these five pre-defined C2 approaches in the C2 approach space. Each C2 Approach provides a different communication structure, i.e., a specific network topology, with a distinct awareness level and it is suitable for specific contexts. Alberts et al. [5] highlights that there is no best unique solution to satisfy all requirements and restrictions of a circumstance.

From a structure where there is no information and decision sharing, so-called conflicted, to a fully connected structure where all entities share information and decision capability, i.e., Edge, these approaches combine different levels of decision making, data sharing, and information to provide the right change to deal with a specific context. Each one looks for distinct entities' awareness levels got through a suitable communications improvement to deal with more complex scenarios and mission conditions [5, 9]. This complexity increases with context changes, i.e., changes in the environment, in the mission, or in the entities. Particularly, the environment involves everything that surrounds the entities, such as enemies' characteristics, weather, and environmental conditions. In summary, the adjustments of any C2 approach space dimension causes changes in the entities' communication structure and protocol, and therefore affecting their awareness level. Based on this, Table 2.2 relates each C2 approach to its key characteristics and the network type used to implement it.

**Table 2.2:** Network types associated to each C2 Approach predefined in C2 approach space

| **NEC Approach (C2 archetypes)** | **Archetype Characteristics** | **Network Type** |
|---|---|---|
| Edge | All elements are connected to everyone; There are $\frac{n(n-1)}{2}$ connections | Fully connected |
| Collaborative | Shared resources; entities interdependence; suitable to holistic problems that can not be fully decomposed; peer-to-peer communication | Scale-free Networks (a few nodes have a very large number of connections) |
| Coordinated | Suitable to problems that can be fully decomposed; share necessary information to execute what was defined | Random Networks (same probability to connect) |
| De-Conflicted | The situation can be decomposed with no cross impact; sub-optimized; not suitable to dynamic scenarios | Small World Networks ($\log N$ connections |
| Conflicted | No communication between entities | Isolated |

Communications improvements conduct the entities to a C2 approach closer to the edge. This more connected structure makes the awareness spread easier, because of the information sharing and collaboration among entities. Such a structure is more suitable for more complex scenarios. Alberts and Hayes [10] shows that C2 always refers to many individuals or entities, and C2, as a tool to coordinate multiple elements, is fundamental. Based on this, C2 applies to teams rather than individuals because of the management and coordination principles involved. In addition, these entities or individuals present behaviors or roles based on a collaboration environment to define the ability of mission accomplishment. Thus, we must define the members' roles or responsibilities to guarantee their interaction modeling.

We select these approaches according to the mission and a set of circumstances [6]. In addition, Alberts et al. [5] recommends following transition rules in case of circumstances complexity increasing. According to the state-of-the-art and state-of-the-practice [6], Table 2.3 shows the most suitable C2 approach corresponding to the endeavor space complexity. Based on this, there is no C2 approach that fits all situations and it depends on the circumstances and resources available.

**Table 2.3:** Endeavor complexity changes

| Endeavor Complexity | Appropriate C2 Approach |
|---|---|
| Low | De-conflicted |
| Medium | Coordinated |
| Medium-High | Collaborative |
| High - Very High | Edge |

### 2.1.4 C2 Agility

Gren and Lenberg [62] defines agility as the ability to deal with an uncertainty environment and subject to change. Agility is, in many scenarios, hard to be defined and many definitions can conduct us to a conflicting idea related simply to performance and fast action. Its definition can suffer slight changes according to the domain where it is being applied or measured [62][7]. Many scenarios use the agility definition as the simple idea related to performance and fast action in face of mission accomplishment [7, 62]. However, fast task execution does not assure proper results. Based on this, we relate the agility concept to a timely tasks' performance presented by Alberts et al. [6].

Based on the software context, Gren and Lenberg [62] defines agility as the ability to create and respond to change in order to succeed in a turbulent and uncertain environment. NATO Report *SAS-085* [6] defines agility as the capability to effect successfully, cope with and/or exploit changes in circumstances. In such a definition, successfully means operating within defined bounds, e.g., resources application, time consumption, and losses.

Both definitions above focus on changes in circumstance because of the uncertainty in the conditions related to the system. This uncertainty increases with the variables related to the environment (cf. Section 2.1. The capability to deal with these changes and uncertainty shows how agile is the entity, being a software or not. Based on this, Alberts et al. [6] extends the agility definition to apply it in C2 context because of circumstance changes and uncertainty naturally embedded in C2 definition. According to this extension, we define C2 agility as the capability of C2 to effect successfully, cope with, and/or exploit changes in circumstances [6]. Similar to agility definition, successfully involves mission requirements. Military domain emphasizes timely requirement in such a list. Alberts et al. [6] shows the relevance of this aspect in several realistic and simulated scenario analyzes.

To provide C2 agility, a fast *sensemaking* [9] is required, i.e., situational awareness, to get a right decision into the scenario to execute the right actions aiming to solve the problems. In case of an automated system, such a perception is obtained by sensors able to get information from the environment to adapt the system according to the circumstance changes. Based on this, awareness and self-monitoring permit a C2 approach to

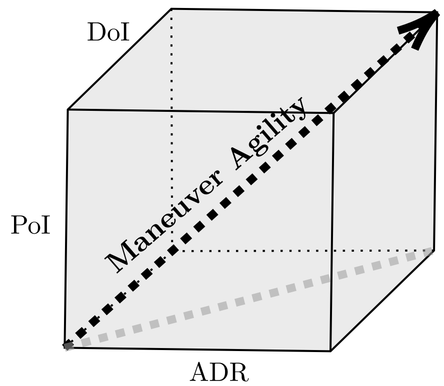



**Figure 2.3:** C2 Approach changes within C2 Approach Space (adapted from Alberts et al. [6]).

change according to the circumstances. All entities have the capability of choosing a C2 Approach during context changes to satisfy new requirements. Alberts and Hayes [9] defines this capacity as the ability to recognize the context changes and the incompatibility of the current C2 approach, identifying another proper C2 approach according to the new circumstance, and performing a timely change. Figure 2.3 shows the common movement inside the C2 approach space to deal with new circumstances more complex.

Alberts [8] proposes a C2 agility metric based on the endeavor space (cf. Section 2.1.2). Such a proposal uses the ratio between the areas in the endeavor space which entities can act and the total endeavor space. A high C2 Agility level conducts this ratio closer to one. A system with C2 agility can deal with more portions within the C2 approach space. However, the first step to perform a maneuver within the C2 approach space is to identify what is the current position in this space. Such an aspect defines the C2 maturity level described in SAS-065 Report [5].

Particularly, the C2 agility raises if the capacity to switch between different C2 approaches increases. The operated C2 Approach is related to the endeavor space complexity, being more connected, e.g., Edge, when such a complexity gets higher (see Table 2.3. According to Alberts and Hayes [10], the C2 agility is the combination of the following elements:

- **C2 Approach Agility**: members' and structure adaptation within the operated C2 Approach to deal with context changes;
- **C2 Maneuver Agility**: C2 Approach change to obtain a different awareness level to deal with context changes.

The entities can reach a suitable adaptation to the new circumstance through a fine tune in their internal configurations, with no C2 Approach change, to improve capabilities. In case of adaptation within the same C2 Approach, we can express this adaptation as an entity reconfiguration or tasks reallocation. However, the new circumstance can

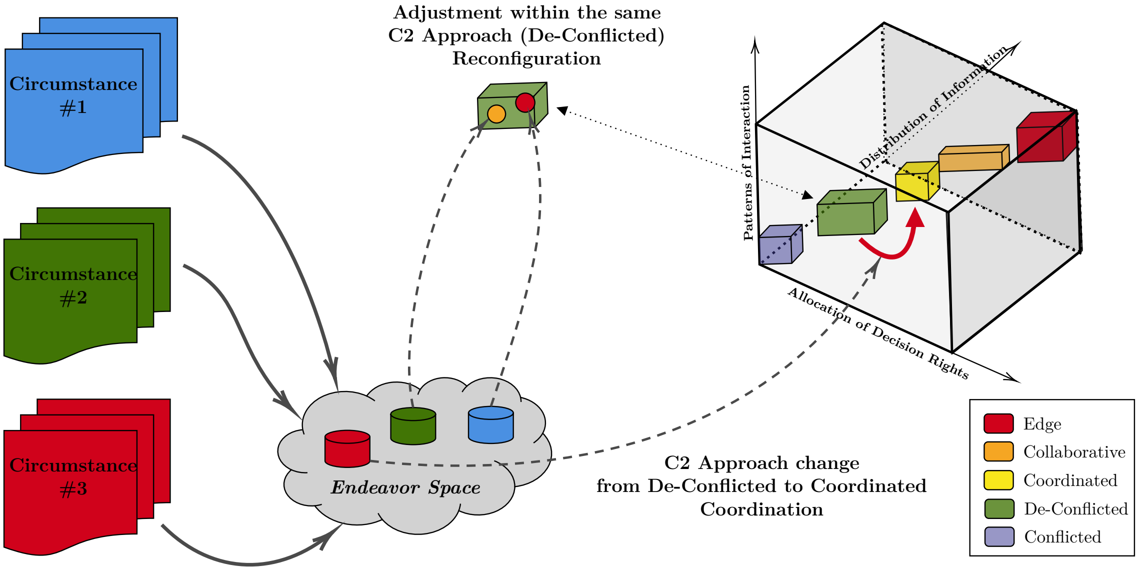



**Figure 2.4:** C2 Approach Agility satisfying new requirements with no C2 Approach change dealing with circumstances #1 and #2, and a C2 Approach change to deal with circumstance #3.

require a deeper adaptation and in this case, a C2 Approach change occurs to deal with new requirements. C2 Maneuver defines this change. Figure 2.4 shows a sequence of three different circumstances where we illustrate such situations of reconfiguration and C2 approach change.

A C2 maneuver implies changes in entities' coordination. Such adaptation provides a new awareness level and it can require some entity reconfiguration. Based on this, these components of C2 agility are orthogonal and affects in each other. Figure 2.5 shows the coordination and reconfiguration composition and effects. In addition, the C2 agility is linked with agile forces and operational concepts. The six components of agility that combine agile individuals, organizations and C2 Systems, so-called agility enablers, are listed below [9]:

- **Robustness:** it is the ability to maintain the effectiveness across a range of tasks, conditions and situations;
- **Resilience:** it is the ability to recover or adjust from a damage or perturbation in the environment;
- **Responsiveness:** it is the ability to react to a change in the environment in a suitable interval of time;
- **Flexibility:** it is the ability to employ multiple ways to succeed, applying different methods;

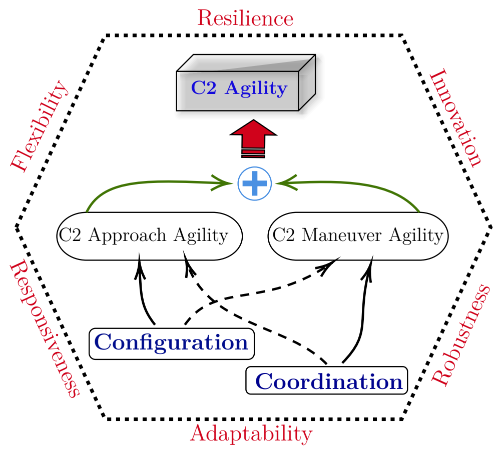



**Figure 2.5:** C2 agility composed of the combination of C2 approach agility and C2 maneuver agility, i.e., reconfiguration and coordination, surrounded by its enablers. Dotted arrows show the potential effects on each other, making a configuration change causes a coordination adapting.

- **Innovation:** it is the ability to do new things or in different ways; and
- **Adaptation:** it is the ability to change the organization or the processes.

The association of these enablers at different levels provides a specific agility capacity [6]. As previously described, C2 Agility relates the entities and the mission challenges with a suitable C2 Approach for each situation. Those enablers characterize required attributes in the C2 structure to recognize the context modifications that affect the mission performing. In addition, they timely select a new suitable C2 Approach to be operated.

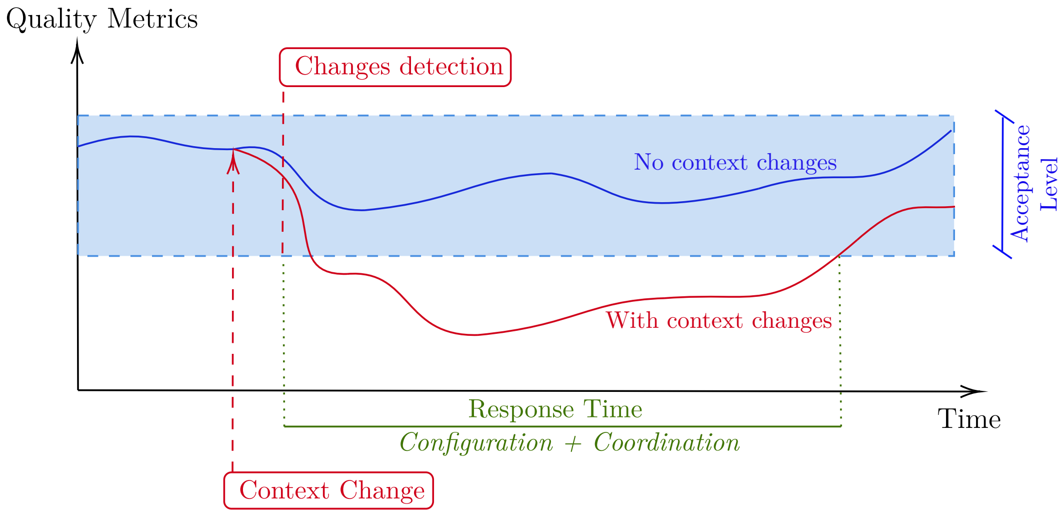



**Figure 2.6:** Lack of agility demanding a long time to react to an abnormality (Adapted from Alberts et al. [6])

Figures 2.6 and 2.7 depict situations where the identification of abnormalities was not effective and results were out of acceptance level. The blue area shows the expected

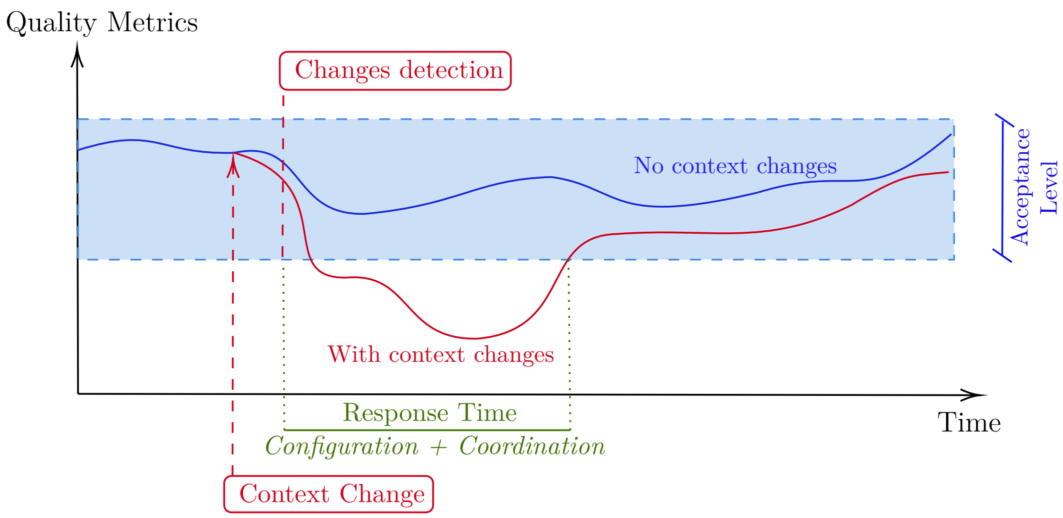



**Figure 2.7:** Agility characterized by a fast response under circumstance changes (Adapted from Alberts et al. [6])

quality limits throughout the mission execution. We notice that the blue continuous line, illustrating the results obtained in case of no context change, is all within tolerance boundaries. No context changes cause some perturbation in mission execution. However, under some context change, the system's behavior changes and the results become worst, as shown by the red continuous line.

After a context change, the system gets a time to detect it, and to adapt to deal with this new circumstance. An entity is agile when it can detect a changing event and to adopt some action to keep results within acceptable levels. Such capability shows an entity's responsiveness compatible with dynamic context. In Figures 2.6, the time elapsed to respond was longer than in Figure 2.7. Such an aspect characterizes more agility in this last figure because of the system's capability to restore the quality level timely within acceptable boundaries.

## 2.2 Software Product Line

Apel et al. [16] define a Software Product Line (SPL) as an approach that provides mass customization by constructing individual solutions based on a portfolio of reusable software components. According to the same authors, it introduces individualism into software production, but still keeps the benefits of mass production in that whole domain and it can serve market segments. It is a paradigm of product generation from a set of common artifacts, e.g., source code, models and documentation.

Based on this, SPL is a paradigm of product generation from a set of common artifacts, e.g., source code, models, and documentation. The base is the reusability concept, and it provides a way of mass customization, satisfying requirements and creating final products more specific to the user's needs. Through this process, the bind time, i.e., time of components combination, occurs before runtime. Besides the compile-time variability, we can generate multiple product working on the granularity. Coarse and fine-grained provides different most-common type of granularity to combine features through components of different types and levels, from method and functions, up to modules and classes.

In addition, resources of a specific language can provide the product-derivation process, or by a tool capable of managing the features combination. The first one shows the variability managing together with the features' implementation in source code. In such a case, the variability managing is in the source code. The second product-derivation process segregates the implementation and the variability management. Here, there is a tool to explore the variability of the feature implemented.

Figure 2.8 presents the SPL life-cycles. The first one, Domain Engineering, defines and implements the commonality and the variability of the product line based on the domain aspects, rules, and requirements. Besides, it defines the common and exchangeable parts of the system, ensuring the creation of reusable artifacts with traceability. The second, called Application Engineering, relates the software product to the reusable domain specific artifacts, exploring the capability to build products from a common base of components and/or artifacts.

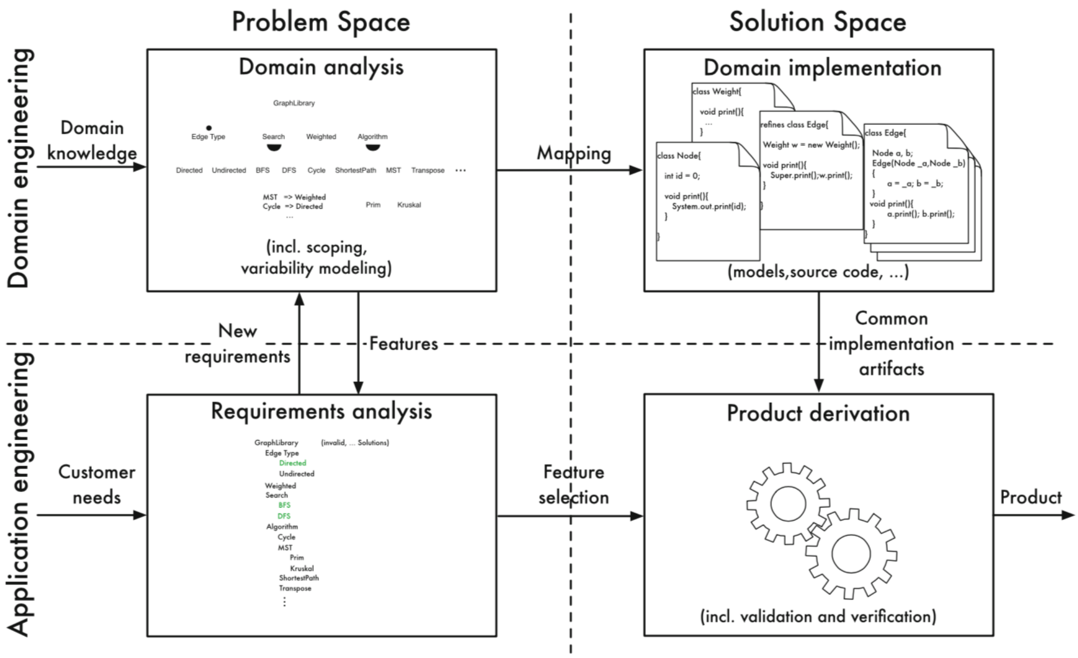



**Figure 2.8:** SPL lifecycle (from original Apel et al. [16])

With SPL, we can tailor a product to different individual customers, simply activating features and resources, providing mass customization. In addition, the reusability leveraged reduces the development costs, improves the quality of the final products, and provides agility for the process. Linux, an operating system with many features that can be activated according to the hardware, is an SPL example [15]. Here, it allows a single system to support different platforms and scenarios, being customized under user requirements.

Product specification using a SPL comprises a selection of features that satisfy the dependences between them. According to Apel et al. [15], a feature ($F$) is a characteristic or a visible behavior of a software system. A Feature Model ($FM$) represents the set of all products of a product line, which is given by a set of *features* $F = \{f_1, ..., f_k\}$ from which a valid set of configurations $[[FM]]$ is obtained, i.e., $[[FM]] : \mathcal{P}(F)$ [76, 107]. Figure 2.9 shows an FM example where the root node shows the system's name.

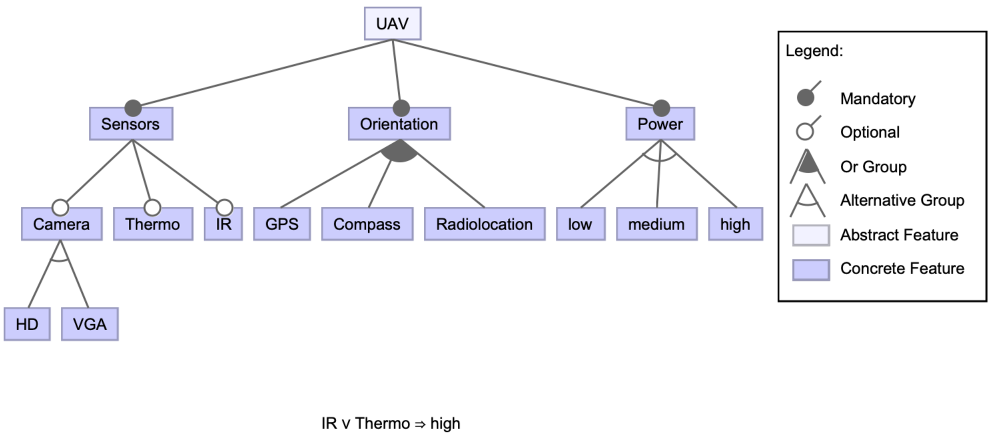



**Figure 2.9:** A Feature Model of the system called UAV with a constraint formula example

A FM shows the constraints between features and their configurability. Concrete features are those implemented and that characterize system functionality accessed by the user. Abstract features are used to organize concrete ones in groups, and they are not implemented. We point to a mandatory feature with a simple edge ending with a filled circle, and it represents a feature that is always present in all products generated from the feature model. The empty circle shows an optional feature that we can select if and only if its parent is selected in the product generated. A set of alternative features are pointed by edges connected to an arc, and in case of the parent feature is selected, only one feature or none is selected. A filled arc shows an *or-features*, and in such a case, at least one feature must be selected. Figure 2.9 shows such a definition in the set of features *GPS*, *Compass*, and *Radiolocation*, where at least one of them must be selected in case of selecting the parent feature *Orientation*. However, to the feature called *Power*, the nodes *high*, *medium*, and *low*, are not required and we can have none of them selected, even with the parent node selected. In addition, we can include constraints as the example where the *Thermo* or *IR* requires the feature *high* selected. i.e., $IR \wedge Thermo = high$.

Composed systems with multiple, interdependent, and decentralized managed SPLs define a Multi Software Product Line (MPL) [86]. Based on this, we can obtain complex products with the customization capability through each SPL configuration. Besides, there is an interdependence between product lines because of any compatibility issue among different configurations in different SPL.

Some requirement or constraint update may require a SPL product derivation at runtime. To deal with that, Bécan et al. [24] defines Dynamic Software Product Line (DSPL) as an SPL capable of configuring itself according to the new user requirements or constraints during system execution.

A DSPL execution in a dynamic scenario is a sequence of valid configurations adopted in runtime, aiming to maintain the system in operation [65]. Figure 2.10 shows an example sequence of states $S1, S2, ..., S8$ that represents a set of valid products from a DSPL following the rules and constraints of its FM. In this diagram, an arrow links each valid configuration presented to another configuration that shows which compatible adaptation can occur in runtime according to context modifications. Such a space of variability can be uncountable and there ate many strategies to analyze it [22].

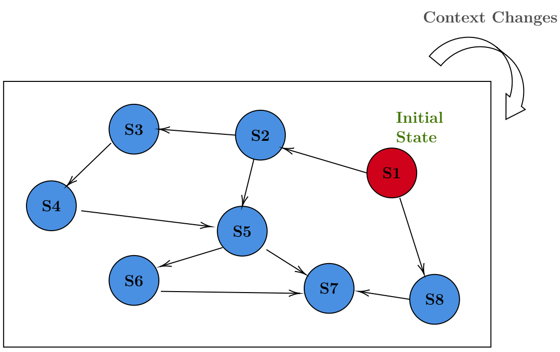



**Figure 2.10:** Set of products of a DSPL in runtime under context changes

Basically, DSPL is an approach to Self-Adaptive System (Self Adaptive System (SAS)) that can be incremented with a feedback loop to deal with constant environment and requirements change that were not previewed at design time, i.e., monitoring context changes that were not foreseen at design time [32, 35, 36]. These loops have states that collect surround and itself information and plan and execute a change. MAPE cycle is a classical loop [32] and its states are:

- **Monitor:** collect data from all managed elements and environment;
- **Analyze:** process all data collected applying metrics and constraints;

- **Plan:** defines necessary changes to get the best results or to fix some issue identified in the previous phase;
- **Execute:** executes the plan developed to adapt the system.

In summary, the monitoring mechanism feeds the system with all information required to perform the adaptation according to the new circumstances. It is the startup of the reconfiguration process, and it makes the system flexible and resilient to deal with dynamic context.

## 2.3 Channel System

Baier and Katoen [20] define channel systems (CS) as parallel systems with processes running independently that communicate via asynchronous or synchronous channels, which are first-in, first-out buffers that may contain messages. CS are assumed to be closed, i.e., inter-processes communication is only allowed within the system, but not outside it. Modern and complex systems do not have a sequential execution of multiple processes. In such a case, the parallelism is the base structure. Each parallel process is specified by a program graph (PG), which is a representation of the system's behavior, abstracting over its transition system semantics. Besides, the PG running in parallel must synchronize and exchange data through channels to provide a complete execution. Equation 2.1 defines the CS formed by a sequence of $n$ PGs running in parallel within the same system:

$$CS = [PG_1|PG_2|...|PG_n] \tag{2.1}$$

Figure 2.11 illustrates a program graph that models a simplified UAV's execution process that reads a set of tasks from a channel called *buffer* and tries to perform them, presenting a result message at the end. It is defined by the tuple
$PG = (Loc, Act, Effect, \hookrightarrow, Loc_0, g_0)$, where

- $Loc = \{wait, plan, move, run, report\}$ is the set of locations, that is, nodes that have a control function;
- $Act = \{start, nextTask, reset\}$ is the set of actions;
- *Effect* is the function modeling the effect of actions on the state of the PG, i.e., it's what the actions do;

- $\hookrightarrow$ is the conditional transition from a location $l$ to a location $l'$, that is labeled with $g : \alpha$, where an action $\alpha \in Act$ is performed when a condition, so-called guard, $g$ is satisfied;

- $Loc_0 = \{wait\}$ is the set of initial locations;

- $g_0$ is the initial condition on the variables of the system.

The location change occurs with conditional transitions labeled by the guard and actions. The guard condition validates the transition if it is *true*. In case of suppressing, i.e., a tautology $(: \alpha)$, the action $\alpha$ is always performed. Besides, the action is a function that changes the PG's state. For example, in Figure 2.11, if the guard *hasTask* holds, the PG changes from location *plan* to *move* and performs the action *nextTask*. Such action sets the variable that points to the following task to be performed. If any guard hold, the PG keeps on the same location waiting.

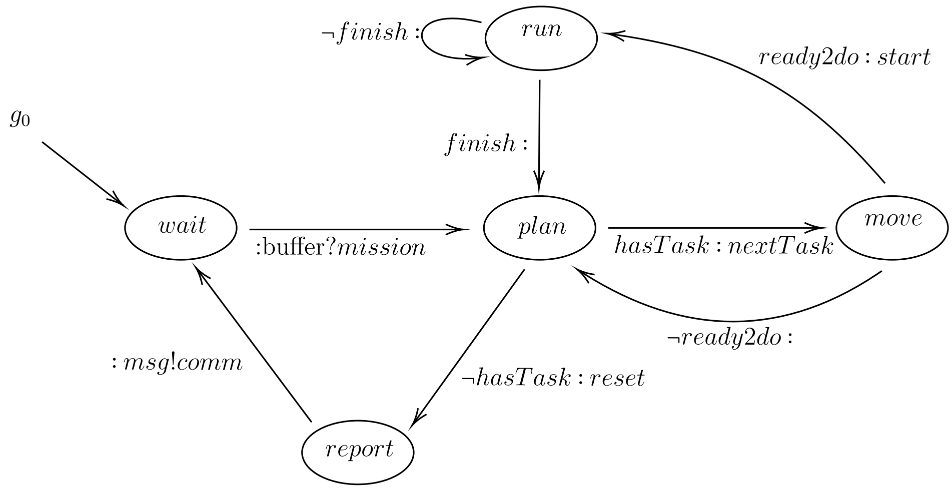



**Figure 2.11:** Program Graph example with 5 locations representing a simplified process played by an UAV

The conditional transition can define specific communication actions among program graphs. Action $ch!x$ writes message $x$ in channel $ch$, whereas action $ch?x$ reads the message from the channel and stores it in variable $x$. In either case, the variables and messages types must be compatible. For example, in Figure 2.11, from locations *wait* to *plan*, data is read from a channel called *buffer* and stored in variable *mission*. As there is no guard condition before the two points, and no alternative action from location *wait*, the operation always occurs when there is data on the channel. The write operation occurs in location *report*, where the variable *msg*'s content is written in channel *comm* and is available to be read by another PG within the same CS. As well as the previous situation, this operation always occurs because of there is no guard condition.

The channel capacity of storing messages, denoted by $cap(ch)$, is defined by its buffer size. As shown in Table 2.4, when $cap(ch) > 0$, communication occurs asynchronously. In such a case, we can store data up to the buffer's size and it works as a *first-in-first-out* queue. Otherwise, *handshaking* takes place, that is, read and write operations occur synchronously. In the previous example, we have that $cap(\mathit{buffer}) = \mathit{0}$ to make the PG stop and wait for information arrival at channel *buffer* written by another PG in the CS, that allows it goes from *wait* to *plan* location.

**Table 2.4:** Channels Capacity - cap(c)

| Capacity | Description |
|---|---|
| $cap(c) > 0$ | There is a delay between the messages transmission and receipt (Asynchronous channel) |
| $cap(c) = 0$ | The message transmission and receipt corresponds to a handshaking (Synchronous channel) |

In summary, in Figure 2.11, when the channel called *buffer* has at least a task, it is read and stored in the variable *mission*, going from location *wait* to *plan*. Next, the condition *hasTask* indicates if there are still tasks in the variable *mission* to be performed. If $hasTask = true$, the action *nextTask* returns the next task available and goes to the location *move*. Otherwise, it runs the action *reset* to put the system in sleep mode and goes to the location *report*, going back next to *wait* and writing the message *msg* in the channel *comm* with the result of the mission performed. From location *move*, if the UAV is ready to perform the task, i.e., $ready2do = true$, the *start* action is performed, and it goes to location *run*. In case of task completed, i.e., $finish = true$, it goes from *run* to *plan* location. When these guard conditions are not satisfied, i.e., $\neg ready2do$ and $\neg finish$, it goes back to the *plan* or keeps in *run* locations, respectively.

# Chapter 3

# Problem Statement

Many domains involve operations requiring coordination among distinct elements, forming a team engaged in the activities on the field, and an optimized use of the resources available. Static conditions, i.e., no changes during the operation are ideal. However, realistic C2 scenarios are under unpredictable changes, e.g., mission changes, entities damages, and environmental effects. Section 3.1 shows the exploratory study that analyzes the impact of dynamism in C2 scenarios, performing experiments to collect evidence of such study relevance. Section 3.2 shows the C2 approaches related to networks topology and their characteristics used by the present work. In addition, Section 3.3 defines the problem statement that motivates this work to propose an approach to deal with such a C2 application. Finally, Section 3.4 presents all assumptions considered by the present study.

## 3.1 Exploratory Study

By their nature, military operations require coordination among different elements, forming a team engaged in the activities on the field and an optimized use of the available resources [9]. For particularly dangerous missions, it is becoming usual the employment of Unmanned Aerial Vehicle (UAV) equipped with different sensors and presenting different levels of resources available [13, 96], as well as an efficient navigation and positioning system [2].

When a team of UAVs has a certain level of autonomy, it needs to plan the execution of the tasks necessary to accomplish a given mission, optimizing available resources to increase the probability of mission accomplishment. This planning can be addressed as a task allocation problem [84].

To deal with this decentralized task allocation problem, Swarm-GAP [51] is a heuristic based on swarm intelligence in which there is no central command unit that has global context knowledge. Thus, the planning decisions are taken and shared among the agents based only on local information[27, 80]. Despite Swarm-GAP's overall good results, its token passing mechanism does not scale when there are many tasks, even if the UAV has available resources. This issue can occur in some situations, e.g., the information exchange among UAVs can get into a loop.

Schwarzrock et al. [108] then proposed three variant algorithms of Swarm-GAP to mitigate the previously mentioned issue. The authors further raised the hypothesis that these algorithms may be useful in dynamic scenarios based on their architecture and logical structure. However, the design of their empirical evaluation relied only on a static scenario with no changes during runtime. Therefore, their claim on the possible usefulness of their approach to dynamic scenarios remains unsubstantiated.

On the other hand, realistic scenarios abound with the aforementioned dynamism, which is often present in the environment or in the system itself at runtime. Indeed, cooperative systems need to deal with this dynamism in order to keep the execution and results level [101]. This capability can be achieved through the form of coordination between the elements that compose the system, by the system reconfiguration or by the tasks reallocation that each element needs to perform. For example, Barbucha [21] explored these cooperative resources in vehicle routing as a multi-agent system to deal with the dynamic context in real scenarios. Additionally, Wu et al. [128] explored UAV application scenarios to emphasize the intrinsic dynamism involved. Further, in the military context, dynamism is pervasive across mission, environment, and the system itself and it has a high impact level on the results of the operation itself, as explained by Alberts and Hayes [10].

Based on this relevance of dynamic scenarios and on the previously mentioned lack of evidence on the usefulness of the proposed approach by Schwarzrock et al. [108] in such scenarios, we assessed that proposed approach in dynamic scenarios and further develop it for better suitability in these scenarios as presented in Amorim et al. [13]. These experiments were performed using that approach in dynamic scenarios, showing that it works, but with a severe level of degradation in the quality of the provided solution. Further, we advanced the state-of-the-art, by extending that original approach with a task redistribution mechanism, which now addresses dynamic scenarios while still keeping the quality level closer to the one achieved in static scenarios. As a result, the extended approach is more suitable for addressing the challenges of real scenarios [13].

The empirical assessment [13] relies on experimental replications complying with the rules presented in Carver et al. [33], Juristo and Gómez [75], Wohlin et al. [127]. All replications follow the same conditions of the original work [108] with the addition of the dynamism to the number of agents performing the mission. We extracted the outcomes to identify the differences in terms of quality and communication overhead. Basically, the key steps performed during this exploratory study were the following:

- The performance of an independent replication to evaluate the original algorithms by Schwarzrock et al. [108] in a dynamic scenario;
- Based on the results acquired with the replication of these original algorithms, improvements were proposed to better address the new dynamic scenario;
- An empirical assessment of the newly proposed algorithms along with the original ones in the dynamic scenario exploring and discussing trade-offs.

Finally, this exploratory study leverages a discussion about the entities coordination in order to optimize the tasks allocation and the mission performance. In addition, we identify opportunities for inserting C2 agility concept in the simulations structure to improve results for dealing with dynamic context.

### 3.1.1 Background and Scenario Formulation

The task allocation problem solution proposed by Schwarzrock et al. [108] was modeled as a Generalized Assignment Problem (GAP) [51]. The goal of that proposal is to maximize the total capability of the agents using a probabilistic approach [119]. Their solution relies on the combination of GAP and swarm intelligence [27], so-called swarm-GAP, using a token-based communication protocol. The proposal presented by Schwarzrock et al. [108] results in three variants based on swarm-GAP:

- *Allocation Loop (AL)*: a control list of visited agents is used and the token runs while there is available tasks. To avoid an infinite loop, the agent registers if it has any resource available to perform a new task before resending the token. Only an agent with free resources will receive the token;
- *Sorting and Allocation Loop (SAL)*: sorts the list of tasks based on the tendency of execution by the UAV, i.e., tasks with high probability and compatibility to be executed are in the first positions of this list. This prevents the first agent, during the token exchange, from being assigned all tasks, not considering other UAVs more suitable to perform some of these tasks and producing better results;
- *Limit and Allocation Loop (LAL)*: extends SAL algorithm and defines a task selection limit per UAV in each round to prevent a greedy strategy and idle agents.

To represent the agent's willingness to perform a task, the proposed algorithms define an attribute called *stimulus*$(st)$. This variable balances the weight among the distance to the task and sensor quality to perform it. Ferreira Jr et al. [51] provided evidence that the most suitable value to this variable, in most situations, is 0.6. Equation 3.1 shows the calculation of task execution tendency using the *stimulus* value and the threshold $\theta$. Higher *stimulus* indicates that less specialized agents will perform the tasks, whereas lower *stimulus* indicates the task will be done by a more specialized agent [27].

The agent threshold $\theta$ is defined by the Equation 3.2 and depends on the agent's capability $k_{ij}$ to perform a task. The capability of an agent $i$ to perform each task $j$ belonging to a set $\mathcal{J}$ of available tasks, is defined by Equation 3.3. The capability calculus uses the Euclidean distance $d(i,j)$ between the UAV and the task, the UAV's quality $Q(i,j)$ to perform the task and the weight $\alpha \in [0,1]$ given to the distance and quality factors.

$$\mathrm{T}_{\theta_{ij}}(st_j) = \frac{st_j^2}{st_j^2 + \theta_{ij}^2} \tag{3.1}$$

$$\theta_{ij} = 1 - k_{ij} \tag{3.2}$$

This capability represents the relation among the distance to the task and the sensor's quality to perform it. The sensors quality represents how suitable is the use of a sensor to execute a specific task.

$$\begin{aligned} k_{ij} = &\frac{\max_{g\in J}\{d(i,g)\} - d(i,j)}{\max_{g\in J}\{d(i,g)\}} \times \alpha + \\ &(1 - \frac{\max_{g\in J}\{Q(i,g)\} - Q(i,j)}{\max_{g\in J}\{Q(i,g)\}}) \times (1-\alpha) \end{aligned} \tag{3.3}$$

According to [108], for a constant UAV set $\mathcal{I} = \{i_1, ..., i_m\}$ with $m$ agents, the allocation matrix $X = (x_{ij})_{m\times n}$, with $x_{ij} = 1$ if the task $j$ is allocated to the agent $i$ and 0 otherwise, it is calculated based on the total capability of each agent $i$ to perform the tasks $j \in \mathcal{J} = \{j_1, ..., j_n\}$, as shown by Equation 3.4. The goal is to maximize the reward obtained by the agents with the tasks execution. This reward is based on the UAV capability $k_{ij}$.

$$X = (x_{ij})_{m\times n} = \arg\max_{X'} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} k_{ij} \times x'_{ij} \tag{3.4}$$

Equation 3.4 is subject to some constraints described as follows. Equation 3.5 ensures that each task is performed, at most, by only one UAV. Furthermore, the tasks allocation only occurs in case of the agent has a minimum quality to perform them, as described in

Equation 3.6. Finally, Equation 3.7 defines that the tasks allocation must to respect the UAV's resources limitation, where $C(i,j) = d(i,j) + c(j)$ is the total resources consumed to the UAV $i$ to perform the task $j$. The term $d(i,j)$ is the distance traveled by UAV $i$ up to the task $j$, and $c(j)$ is the time spent to perform the task $j$.

$$\forall j \in \mathcal{J} \sum_{i \in \mathcal{I}} x_{ij} \leq 1 \tag{3.5}$$

$$\forall x \in X \mid x_{ij} = 1, Q(i,j) > 0 \tag{3.6}$$

$$\forall i \in \mathcal{I} \sum_{j \in \mathcal{J}} x_{ij} \times C_{ij} \leq r_i \tag{3.7}$$

Algorithms 1 and 2 show the pseudo code for AL and SAL, respectively. Once an agent receives a token (line 1) with a list of available tasks, it calculates the tendency (line 5). To decide on task allocation, the agent analyzes its tendency and compares its available resources with the estimated cost to execute the task ($c_j$) (line 6). When the agent decides to get a task, this task is allocated (line 7) and the agent's resources are reduced accordingly (line 8). Lines 9 to 15 of Algorithm 1 are common to all three algorithms, and mark the agent as visited and send the token to the next not visited agent. The LAL algorithm was not listed due to its similarity with SAL, differing from the latter by an upper bound to the number of tasks selected by each agent per token pass (third condition in line 8).

**Algorithm 1:** Pseudo code - Allocation loop (AL) (from Schwarzrock et al. [108])

1: Receive Token
2: Compute available resources $r_i$
3: **for all** *available tasks* **do**
4:     Compute capability $k_{ij}$
5:     Compute tendency $T_{\theta_{ij}(st)}$
6:     **if** *roulette()* $< T_{\theta_{ij}(st)}$ *and* $r_i \geq c_j$ **then**
7:         Allocate task $j$ to agent $i$
8:         Decrease resource $r_i$
9: Mark agent as visited in the token
10: **if** *there are still available tasks* **then**
11:     Inform token if it has availability to perform any one of these tasks
12:     **if** *all agents already receive the token* **then**
13:         Clean the list of visited agents
14:         Fill list of visited agents with the unavailable agents
15:     Send the token to a not yet visited agent

**Algorithm 2:** Pseudo code - SAL (from Schwarzrock et al.[108])

Receive Token
Compute available resources $r_i$
**for all** *available tasks* **do**
  Compute capability $k_{ij}$
  Compute tendency $T_{\theta_{ij}}$
Sort tasks by descending tendency
**for all** *available tasks sorted by tendency* **do**
  **if** *roulette()* $< T_{\theta_{ij}(st)}$ *and* $r_i \geq c_j$ **then**
    Allocate task $j$ to agent $i$
    Decrease resource $r_i$
The same as lines 9 to 15 of the Algorithm 1

In the scenario adopted by Schwarzrock et al. [108], each UAV can perform more than one task, but each task can be performed only by one agent. The mission transmission, by a Central Command, is done using a token-based protocol that transmits all information about the tasks and agents association. With this protocol, when a specific UAV chooses a task and allocates it to itself, it sends this information to the next agent of the team. Thus, the next agent will know which tasks are available for execution. Additionally, *tick* is the unit used to measure the execution time of these algorithms.

However, the scope of that study is limited to static scenarios, and it does not reflect the possibility of changes as it occurs in real scenarios, e.g., a UAV can be shot down or new tasks added, changing the conditions of a mission. Thus, Equation 3.4 does not provide evidence on the suitability of the proposed solution for dynamic contexts. To explore this limitation and to extend the study reported by [108], Equation 3.4 was rewritten to address variation of the mission in relation to the time of its execution. Accordingly, Equation 3.8 now introduces the variable $t$, which represents time and thus the allocation matrix becomes a time-dependent function $X(t)$. This update makes the system more applicable to dynamic contexts.

$$X(t) = (x_{ij})_{m \times n} = \arg\max_{X'} \sum_{i \in \mathcal{I}(t)} \sum_{j \in \mathcal{J}(t)} k_{ij} \times x'_{ij} \tag{3.8}$$

Eventually, it is needed to recalculate the matrix $X$ at every time instant $t$ in which some change in the context of the mission occurs or is simulated. All constraints previously applied to Equation 3.4 still hold in this new approach.

### 3.1.2 Exploratory Research Method

To obtain empirical evidence on whether the original algorithms work in a dynamic scenario and to propose their eventual improvement to better address dynamism, this exploratory study relies on action research (AR) [43]. Based on this method, the research followed an iterative process comprising experiment replications and algorithms extension considering dynamic scenarios. This section first addresses the fundamental requirement of reproducibility of the original study (Section 3.1.2.1). Next, the dynamic scenario common to both replications is described, and finally a description of the extended algorithms is presented.

#### 3.1.2.1 Original Experiment Reproducibility

The analysis of the empirical replications in software engineering was performed by many researchers and extensively reported in the literature [33, 59, 75, 127]. They presented requirements to classify an experiment as reproducible and important guidelines to perform a reproduction, as well as the main experimental elements that must be observed during a replication. Madeyski and Kitchenham [89] presented the difference among replication and reproduction, highlighting aspects to guide an independent replication, which was the approach used in this present work.

According to Madeyski and Kitchenham [89], reproducible research (RR) is defined as a study that can be reproduced from all information and data registered about a given set of experiments.

Based on that definition, the original study presented by Schwarzrock et al. [108] can be classified as a RR since all experiments previously done were reproduced, obtaining results within the standard deviation margin of the original ones[1]. Table 3.1 shows all independent and dependent variables used by the experiment reproduction, in which the slight differences obtained were due to the distinct hardware configurations used in the original experiments. This procedure aimed at comparing the results to the original experiment in similar conditions of execution, including the number of thirty runs for each algorithm.

To explore different situations regarding mission complexity and available resources, the original and the replicated experimental scenarios were the following:

1. 3 UAVs; 4 tasks; 300 ticks as deadline; 100 x 80 px area size,
2. 3 UAVs; 8 tasks; 300 ticks as deadline; 100 x 80 px area size,
3. 3 UAVs; 16 tasks; 300 ticks as deadline; 100 x 80 px area size,

[1] Reproduction results available at https://github.com/junieramorim/replicationMaterial

**Table 3.1:** Variables used by the experiment reproduction

| Variable | Description |
|---|---|
| **Independent variables** | |
| Total UAVs | number of agents executing the tasks |
| Total tasks | number of tasks to be performed called mission |
| Stimulus | agent willness to perform a task (correlates task distance and sensor's quality) |
| Field size | size of the field where the tasks are spread |
| Total time | time limit to perform all tasks (mission deadline) |
| Sensor quality | number that represents how suitable the sensor is to a specific type of task |
| Task type | divides the tasks in different types |
| Number of Executions | number of times that each algorithm is executed to collect the mean |
| **Dependent variables** | |
| Capability | agent and tasks compatibility based on the *stimulus* factor |
| Reward | total of agent capability related to its completed tasks |
| Elapsed time result | time measurement to perform all tasks made in *ticks* |
| Completed tasks | number of tasks fully performed by any agent |
| Number of Messages | total number of messages (tokens) exchanged by all agents |

4. 3 UAVs; 32 tasks; 300 ticks as deadline; 100 x 80 px area size,
5. 6 UAVs; 64 tasks; 300 ticks as deadline; 200 x 160 px area size,
6. 9 UAVs; 96 tasks; 300 ticks as deadline; 300 x 240 px area size.

The platform used to perform all experiments presented by this work was the same of the original study [108], NetLogo, version 5.3.1, a multi-agent programmable modeling environment. This choice ensured to keep the previous code compatibility and allowed its complete reuse, besides allowing to implement changes in reproductions, which will be described later in this text. The original code was imported to the platform, the executions were done to generate results in text format through the built-in console. With a selection and copy feature, these results were exported to a file to be read by R Studio import tool, generating final graphs.

Since the study by Schwarzrock et al. [108] is reproducible, it is possible to perform independent replications from it. Replications allow exploring different contexts from the original study by changing variables, conditions, and controls to analyze the effects on the results [33].

#### 3.1.2.2 Dynamic Scenario Inclusion

The original work presented by Schwarzrock et al. [108] was based on a static context. In that scenario, important elements such as the tasks that compose the mission, or the number and the type of agents, are defined in the beginning and do not change during the mission execution.

However, as already mentioned, considering a military operation environment, this assumption is not realistic, limiting the usefulness of the solution in the real world. To explore a more realistic context, dynamism in the scenario must be considered. This work targets this aspect by considering a varying number of agents during the execution of the algorithms, randomly taking down of some agents. The agents removal follows a time slice $\mu$ defined by Eq. 3.9, i.e., one UAV will be taken down in each interval defined by the total mission time (measured in ticks) divided by the initial number of agents in the team. This proceeds until a single UAV remains in the team, as shown by Algorithm 3. This algorithm is implemented into the main loop of each original code variant proposed by Schwarzrock et al. [108], where the token exchange among agents occurs, as well as the task allocation, and it is executed in each pass of the main method.

$$\mu = \frac{total\_time}{total\_agents} \tag{3.9}$$

**Algorithm 3:** Pseudocode for taking down an UAV(agent) that is inserted after line 9 of Alghorithm 1 and used by the three variants proposed by the original study [108]

```
t = current time in ticks
if number of operational agents > 1 then
    if (t mod μ) = 0 then
        Choose an agent randomly from those with resources
        Remove all resources from the selected agent
        Remove unfinished tasks from the selected agent
        Operational agents counter is decreased by 1
        Update not visited agents list in the token
        Set agent color as "red"
    Increment time by 1 tick
```

A screen of the dynamic scenario running can be seen in Figure 3.1. It illustrates the simulation graphical interface, where the symbol for the tasks is represented by a $X$ shape and the symbol for a UAV has the shape of an airplane. The black airplane represents a functional airplane and the red one represents a UAV dropped and out of operation.

As the quantity of agents changes during the execution, this characteristic gives dynamism to the context. It grants the capacity to analyze the agents' response and to assess the original algorithms in this new dynamic scenario.

Furthermore, to reduce the standard deviation in the results, all experiments were executed 100 times instead of only 30, as it was done in the original study. It was noticed a decrease in the error using this increased number of runs.

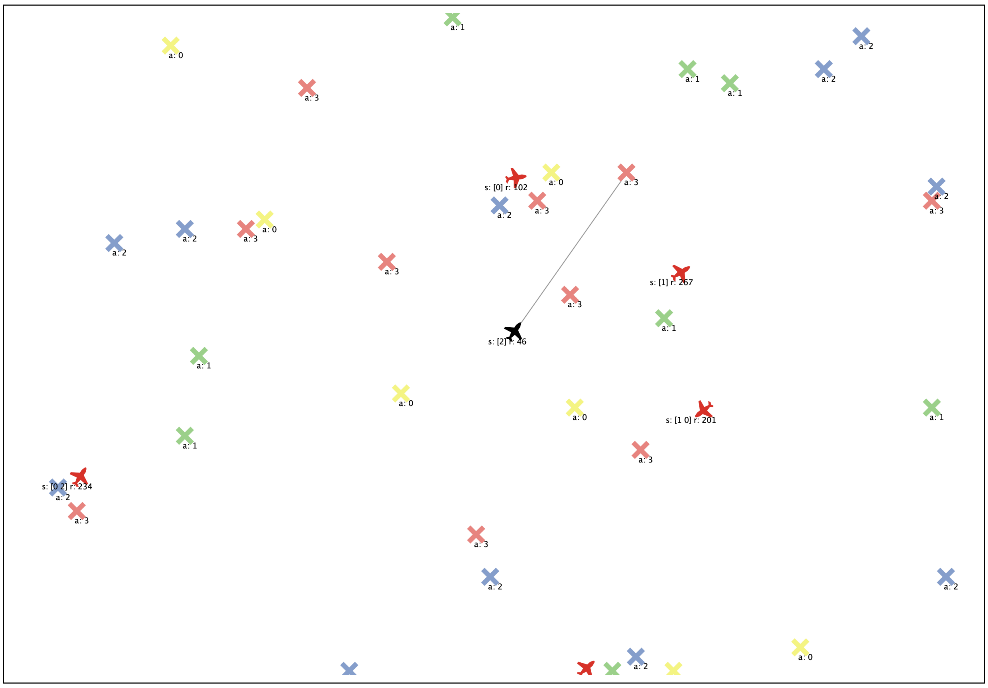


**Figure 3.1:** NetLogo 5.3.1 screen with dropped UAVs in red, in the dynamic scenario

#### 3.1.2.3 The Extended Algorithms

To address the dynamic scenario required by a realistic scenario, changes were implemented in the original algorithms [108] aiming to optimize task allocation. This optimization occurs with the best association of the sensor with the task to be executed, i.e., choosing the sensor that has the best quality attribute to that kind of task.

The new procedure is represented by Algorithm 4 and inserted after line 9 of Algorithm 1. Basically, it is done in two main steps: 1) Release all tasks not completed, and 2) Reset the token to reinitialize the task allocation.

The algorithm implementation uses a communication structure based on a token protocol with a ring architecture (token ring). This solution defines the network behavior and how the information is exchanged. With this strategy, the first agent visited by the token has more tasks available to choose, and this availability decreases on each visited element in the network.

The first step in the proposed change comes to minimize a limitation in the token-based protocol with a ring architecture (token ring). This limitation is the available tasks reduction when the token reaches each agent forming the ring, that allocates one or more tasks reducing the number of tasks to allocation. This change is represented by the block from line 1 to line 6 in Algorithm 4 and it releases all allocated tasks to guarantee another chance to reallocate not yet completed tasks.

This procedure aims at optimizing the task reallocation among the remaining UAVs to maximize the best usage of the on-board sensors. Thus, there is a new chance of an UAV being assigned to a task more suitable to its sensors, since the token, running its ring path, is refilled with the released tasks. This provides the possibility of another agent being assigned any of these newly available tasks.

**Algorithm 4:** Code inserted into the three variants proposed by Schwarzrock et al. [108] to better deal with dynamic context.

```
1:  if an agent was dropped then
2:      for all agent j alive (not dropped) do
3:          for all Task t not completed in agent.(TASK_TO_DO) list do
4:              add t to available tasks list in the Token
5:              update estimated costs (ticks)
6:              adjust stimulus st_j value
7:      if j in agent visited list then
8:          remove agent from the list visited
9:          add agent to the not visited list
10:         clear control parameter of the token
```

When the tasks are released, it is necessary to update the value of the estimated costs in terms of *ticks*, i.e., it is necessary to recalculate the time in *ticks* necessary to move to the task and to execute it. The tasks not executed should not consume agent resources or leave them locked. This explains the importance of the instruction executed in line 5 of Algorithm 4.

The second step, limited by the block of lines 7 to 10, in the Algorithm 4, is responsible for clearing the history of the visited agents, ensuring the token will pass again through all UAVs that are functionally active. This step then gives another chance to all agents to get another task or the same. It is always done maximizing the quality of sensors applied to the tasks. The parameter of line 10 of Algorithm 4 refers to the counter of visiting times used by LAL.

In the original algorithm, all three proposed variants (AL, SAL and LAL) share a common structure formed by some procedures implemented in NetLogo. The proposed extension was performed in a common part and it is shared by all these variants. Besides what is displayed in Algorithm 4, auxiliary structures were also created to control the number of dropped and remaining UAVs.

Each algorithm (AL, SAL or LAL) has a limit to the token's visitings quantity to agent. It is defined to avoid an infinite loop in the token passing and it needs to guarantee a visiting to all agents. Regardless of the algorithm, the token reset permits to pass by the UAVs, at least, one more time than what was established initially.

It is useful particularly in AL, which uses a greedy strategy to allocate the tasks. This way, it causes a task reallocation among the remaining UAVs. As the token has the opportunity to visit the agents more often, the number of exchanged messages increases.

Furthermore, the *stimulus* attribute value was updated to adjust the willingness of the agent to perform each task and to analyze its impact on a dynamic context (line 6 in the Algorithm 4). The initial value used by the original study was 0.6 as explained in Section 3.1.1, giving a good balance among distance and quality of the sensors at the choose task moment, but different values were tested to measure their impact on the results. The results of these simulations are discussed in Section 3.1.4.

### 3.1.3 Exploratory Replications

Following the overall process defined by action research, and using the new dynamic scenario (Section 3.1.2.2), the original algorithms and the extended ones (Section 3.1.2.3) were exercised with two replications to identify possible limitations, constraints, and opportunities for improvement. The first replication (Section 3.1.3.1) was conducted to assess how well the original algorithms work in the dynamic scenario, whereas the second replication (Section 3.1.3.2) assesses the extended algorithms in the same dynamic scenario.

All replications in this work use the same time limit of the original study, i.e., a total of 300 ticks. The displacement and task execution have a time consumption and the total time consumed to perform all tasks allocated has to be within this upper limit of 300 ticks.

To perform the replications, some definitions had to be done in terms of parameters applied during the execution of the algorithms. Three independent variables changed during experiments are listed in the following. All the other parameters received the same value used by Schwarzrock et al. [108].

- *stimulus*: parameter that defines a priority relation among the distance to the task and the quality of an on board sensors to perform the task. This value is considered during the agent capacity calculation and the study by Ferreira Jr et al. [52] suggested that the best value in most situations of Swarm-GAP strategy application in static scenarios is 0.6; However, in dynamic scenarios, there is a result impact with different values to *stimulus*. Table 3.2 shows the intervals of difference in results using different values for the *stimulus*. Since the differences are low, value of 0.6 was fixed as standard to *stimulus* for the final replications so that they have similar conditions to be compare to the results obtained by previous experiments.

**Table 3.2:** Impacts on the results of the experiments caused by different values of *stimulus* parameter compared with the using of the standard value (0.6)

| stimulus($st$) | results impact |
|---|---|
| $0.6 < st \leq 0.99$ | [1%,3%] |
| $0.1 \leq st < 0.6$ | [-3%,-1%] |

- *number of executions*: Each replication is repeated $n$ times to generate a set of results and, based on them, calculate the average and standard deviation of each attribute measured during the experiment. The number of executions was increased to 100 runs (instead of 30 used in the original study [108]) for each algorithm aiming at the reduction of the resulting standard deviation, thus getting a higher precision. This increase generated results closer to the normal distribution, minimizing the standard deviation (see a sample result set with 50 executions of the extended SAL algorithm in Figure 3.11).

- *number of UAVs*: Changes in the number of active agents force the algorithms to reconfigure the task allocation among the UAVs. This feature makes the scenario dynamic because the number of agent changes during the experiment run time. During the allocation tasks procedure, tasks not finished by the dropped UAVs are reallocated according to the extended algorithms proposed.

The focus of the analysis was on the broadest scenario with 9 UAVs. It was chosen because, with smaller scenarios (with lower number of UAVs), the differences in preliminary results were not statistically significant due to high standard deviation. Furthermore, the independent replication was done with the three algorithm variants (AL, SAL and LAL). The source code used during the experiments and the results obtained can be accessed through a link to GitHub repository[2].

Changes in the number of UAVs ($m$) occur following an interval $\mu$ to simulate the occurrence of changes of non-controlled variables in the environment, e.g., hazard, enemy treats and equipment issues, taking the system closer to the real scenario. In this study, these changes are implemented as agent removals from the team which occur at each time $t' = t + \mu$. Thus, the allocation procedure defined by Equation 3.8 is time-dependent and needs to be recalculated every interval $\mu$. Based on this model, the new agent set $\mathcal{I}(t') = \mathcal{I}(t) - i_g$, where $i_g(1 \leq g \leq m_t)$ is a random agent and $m_t = |\mathcal{I}(t)|$. Similarly, the set of tasks $\mathcal{J}$ at the moment $t'$ also changes with the exclusion of completed tasks and results in $\mathcal{J}(t') = \mathcal{J}(t) - \mathcal{J}_f(t')$, where $\mathcal{J}_f(t')$ is the set of completed tasks up to time $t'$.

[2]https://github.com/junieramorim/replicationMaterial

**Table 3.3:** Total reward, elapsed time, quantity and quality of the completed tasks and number of exchanged messages (tokens sent) for 100 runs of each algorithm with the following parameters: 9 UAVs and 96 tasks in area of 300x240 pixels with deadline of 300 ticks.

| | AL | SAL | LAL |
|---|---|---|---|
| | Mean (St.Dev.) | Mean (St.Dev.) | Mean (St.Dev.) |
| **Results of the original study in the same original static scenario** | | | |
| Total reward | 16.724 (±2.1908) | 37.9608 (±1.1119) | **44.733** (±1.5961) |
| Elapsed time (norm) | **0.9894** (±0.0064) | 0.9763 (±0.0092) | 0.9693 (±0.0124) |
| Comp. tasks (norm) | 0.2774 (±0.0276) | 0.4674 (±0.0170) | **0.5226** (±0.0162) |
| Quality (norm) | 0.7865 (±0.0641) | 0.9680 (±0.0202) | **0.9752** (±0.0159) |
| Sending token | 9.8667 (±1.1958) | 9.7000 (±1.2360) | **51.500** (±1.4797) |
| **Independent replication using the dynamic scenario (number of UAVs changes)** | | | |
| Total reward | 8.6995 (±1.9489) | 23.6496 (±2.6004) | **28.348** (±2.8464) |
| Elapsed time (norm) | **0.9767** (±0.0135) | 0.9652 (±0.0220) | 0.9513 (±0.0281) |
| Comp. tasks (norm) | 0.1418 (±0.0281) | 0.2852 (±0.0315) | **0.3278** (±0.0328) |
| Quality (norm) | 0.5753 (±0.1421) | 0.5751 (±0.1358) | **0.5858** (±0.1424) |
| Sending token | 10.0900 (±1.3416) | 9.9800 (±1.1805) | **49.4600** (±1.9562) |

In the presented replications, the context change occurs at every interval $\mu$ until there is only one UAV. Based on this, the matrix $X$ is recalculated $(m_0 - 1)$ times, where $m_0 = |\mathcal{I}(0)|$, i.e., the initial number of agents.

#### 3.1.3.1 First Replication

The first independent replication to collect empirical results was done with the algorithms proposed in the original work by Schwarzrock et al. [108] using the dynamic scenario presented in Section 3.1.2.2. It aims to find evidence to either support or refute the conjecture made in the referenced study that it would be fully functional in dynamic scenarios.

Table 3.3 shows the results from all algorithms proposed by the original study [108] applied to dynamic context, with the highest values highlighted. These results show the total reward obtained with the completed tasks, the total tasks that were finished, the portion of total time ($[0\%, 1\%]$) used to perform the maximum possible tasks, the total quality that relates the sensors and the performed tasks, and the number of tokens sent during the execution of the experiments. Some results are normalized to make the comparisons easier.

The analysis concentrates in all results of the experiments that used more agents (9 UAVs) and tasks (96) since they represent the highest value difference between the dependent variables of the original study and those obtained in the dynamic scenario. All other results are available in the repository[3]. Overall, these results suggest that the original algorithms work in a dynamic scenario due to the average level obtained with no inconsistent value to all variables, e.g., zero or negative values. Nevertheless, those results were significantly different from the values obtained by the original work, indicating up to 50% decrease in reward and quality. Graphs comparing the results obtained with this

[3]https://github.com/junieramorim/replicationMaterial

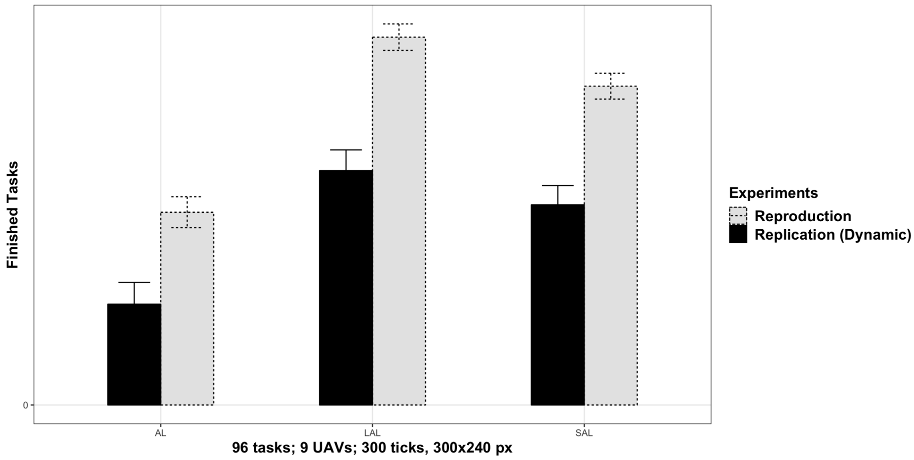



**Figure 3.2:** Finished Tasks (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

first replication in the dynamic scenario and the reproduction of original study in the static context are presented in the following, along with a related discussion.

Graphs comparing the results obtained with this first replication in the dynamic scenario (Section 3.1.2.2) and the reproduction of original study in the static context are presented in the following, along with a related discussion.

The metric concerning the number of finished tasks (Figure 3.2) presented a reduction greater than 40% due to the fact that there are fewer agents performing the mission. Indeed, as the dynamic scenario makes the number of agents decreases at runtime, not all tasks allocated are completed, which explains this difference.

Another metric that suffered a significant degradation (greater than 30%) was the quality, as seen in Figure 3.3. This quality is calculated as a sum of the sensors compatibility with the tasks, i.e., each sensor has a number between 0 and 1 that defines how suitable this sensor is to be used in a specific task, correlating the sensors characteristics and the type of task. This number ($Q_{ij}$) defines the agent $i$ quality to perform the task $j$ and it is used to calculate the agent capability ($k_{ij}$) (Equation 3.3).

The reduction in this attribute was in consequence of the algorithms characteristics that simply discard the allocated tasks, from the available tasks list, when the agent is shut down. These tasks are not reallocated among the remaining UAVs, thus reducing the quality of the sensors related to the finished tasks.

On another hand, other metrics presented very similar results to the original experiment, as the number of messages exchanged, and the total reward and execution time.

The ring network relies on passing a token to each element. Even reducing the number of elements, the number of messages did not decrease because the token runs while it has tasks available or there is a UAV not visited by the token.

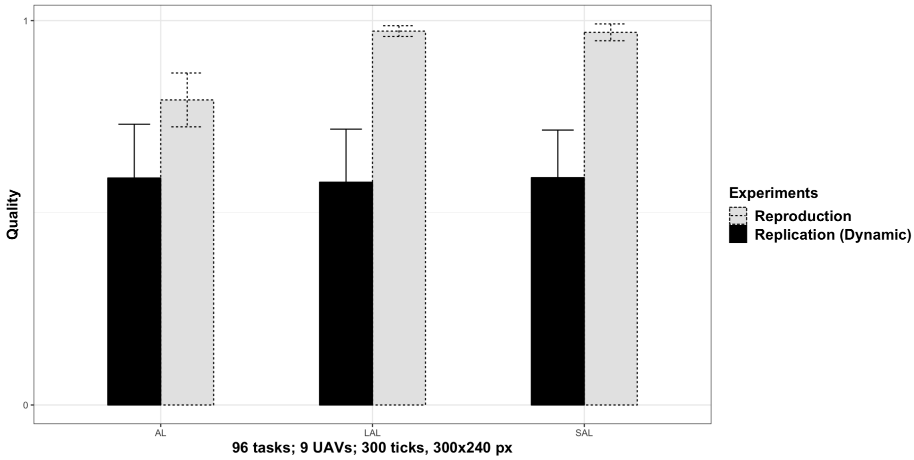



**Figure 3.3:** Quality (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

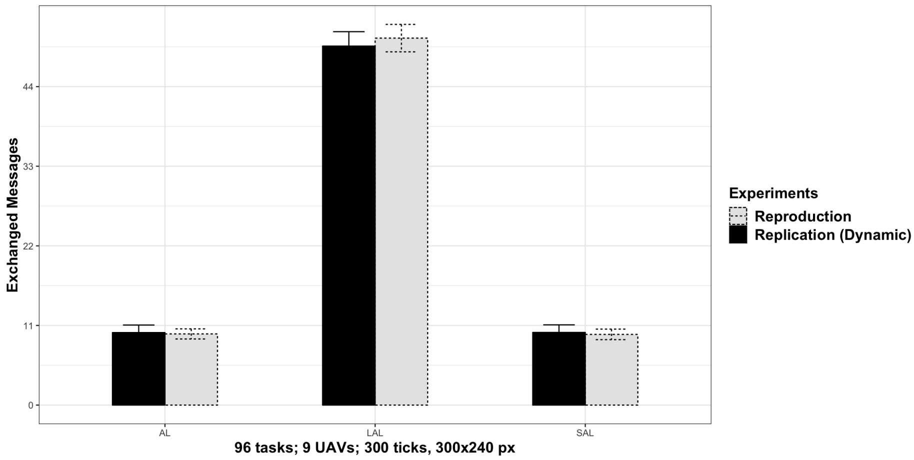



**Figure 3.4:** Exchanged Messages (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

As the total reward is the sum of the UAV capability $k_{ij}$ related to the finished tasks, and the capability, defined in Schwarzrock et al. [108], is a function of distance to the task and sensor quality, the reduction of quality causes a proportional decrease in the capability, as confirmed by Figure 3.5.

The time measurement was done in percentage of total mission time (in ticks) utilization. The result obtained was equivalent to the original study experiments, considering the standard deviation (see Figure 3.6). It is explained because the dynamic scenario ensures at least one UAV and this agent spends the most of the available time (in ticks) to execute the tasks allocated to it.

According to what was presented in Schwarzrock et al. [108] and Ferreira Jr et al. [51], the more suitable value for *stimulus* parameter is 0.6. However, during the replications, different values were investigated. The resulting differences were not significant compared

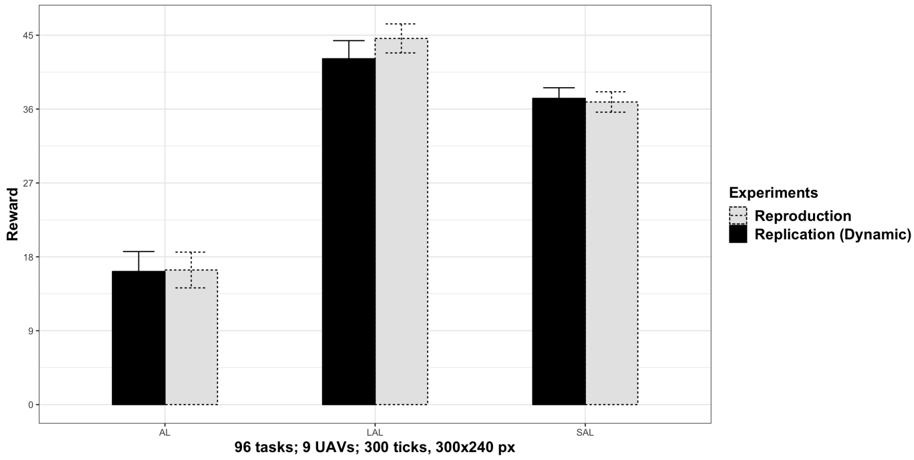



**Figure 3.5:** Total Reward (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

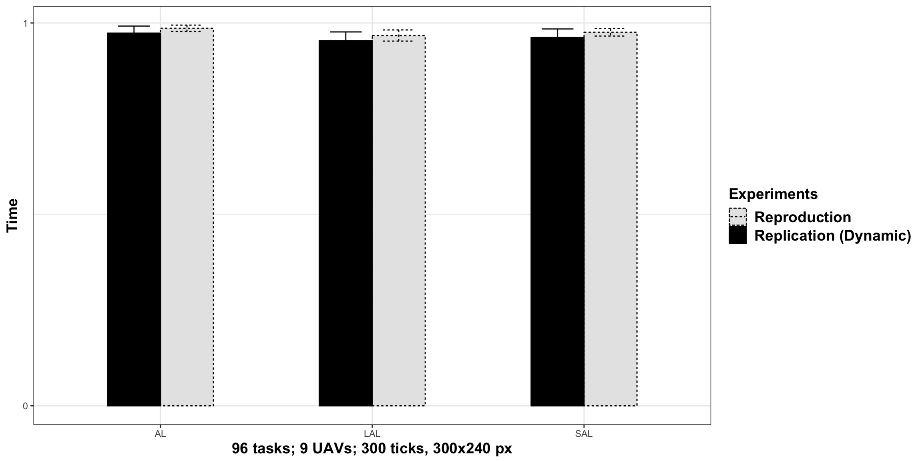



**Figure 3.6:** Elapsed Time (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

**Table 3.4:** Total reward, elapsed time, quantity and quality of the completed tasks and number of exchanged messages for 100 runs of each algorithm after modification to deal with dynamic context.

| | AL | SAL | LAL |
|---|---|---|---|
| | Mean (St.Dev.) | Mean (St.Dev.) | Mean (St.Dev.) |
| **⟶ 3 UAVs and 4 tasks in the area of 100x80 pixels with deadline of 300 ticks** | | | |
| Total reward | 2.0608 (±0.4417) | 2.1512 (±0.3328) | **2.2684** (±0.2372) |
| Elapsed time (norm) | **0.4147** (±0.1358) | 0.4085 (±0.1422) | 0.3506 (±0.1159) |
| Comp. tasks (norm) | 0.9875 (±0.0548) | **0.9975** (±0.0250) | **0.9975** (±0.0250) |
| Quality (norm) | 0.8218 (±0.1747) | 0.8437 (±0.1434) | **0.9215** (±0.1115) |
| Sending token | 5.0800 (±2.3471) | 5.3000 (±2.8972) | **7.0300** (±2.3632) |
| **⟶ 3 UAVs and 8 tasks in the area of 100x80 pixels with deadline of 300 ticks** | | | |
| Total reward | 3.2771 (±0.5984) | 3.4573 (±0.5436) | **3.7628** (±0.4796) |
| Elapsed time (norm) | 0.6927 (±0.0762) | **0.7016** (±0.0895) | 0.6714 (±0.0642) |
| Comp. tasks (norm) | 0.8538 (±0.1006) | 0.8475 (±0.1060) | **0.8888** (±0.0886) |
| Quality (norm) | 0.7151 (±0.1218) | 0.7494 (±0.1399) | **0.8047** (±0.0978) |
| Sending token | 10.6100 (±2.5222) | 10.1900 (±2.9566) | **14.7600** (±2.8467) |
| **⟶ 3 UAVs and 16 tasks in the area of 100x80 pixels with deadline of 300 ticks** | | | |
| Total reward | 4.6678 (±0.9381) | 6.8921 (±0.9166) | **7.8121** (±0.7524) |
| Elapsed time (norm) | 0.7393 (±0.0720) | **0.7575** (±0.0658) | 0.7510 (±0.0720) |
| Comp. tasks (norm) | 0.5131 (±0.0723) | 0.6113 (±0.0600) | **0.6500** (±0.0526) |
| Quality (norm) | 0.7505 (±0.1290) | 0.8813 (±0.0902) | **0.9532** (±0.0505) |
| Sending token | 9.7600 (±2.4000) | 10.1700 (±2.2699) | **25.7400** (±3.4160) |
| **⟶ 3 UAVs and 32 tasks in the area of 100x80 pixels with deadline of 300 ticks** | | | |
| Total reward | 4.7568 (±1.1019) | 10.0601 (±0.9066) | **11.0194** (±0.7653) |
| Elapsed time (norm) | **0.7512** (±0.0692) | 0.7343 (±0.0558) | 0.7271 (±0.0538) |
| Comp. tasks (norm) | 0.2563 (±0.0433) | 0.3906 (±0.0340) | **0.4063** (±0.0291) |
| Quality (norm) | 0.7134 (±0.1094) | 0.9502 (±0.0479) | **0.9742** (±0.0269) |
| Sending token | 6.8900 (±1.1712) | 7.0600 (±1.2045) | **32.1500** (±1.8607) |
| **⟶ 6 UAVs and 64 tasks in the area of 200x160 pixels with deadline of 300 ticks** | | | |
| Total reward | 5.1439 (±1.2455) | **16.3631** (±1.2697) | 13.5601 (±0.9898) |
| Elapsed time (norm) | **0.8836** (±0.0836) | 0.8331 (±0.0669) | 0.8100 (±0.0733) |
| Comp. tasks (norm) | 0.1231 (±0.0245) | 0.2763 (±0.0202) | **0.2993** (±0.0213) |
| Quality (norm) | 0.7834 (±0.1374) | 0.9919 (±0.0185) | **0.9968** (±0.0127) |
| Sending token | 22.4000 (±1.3559) | 22.1300 (±1.3154) | **95.2100** (±3.5142) |
| **⟶ 9 UAVs and 96 tasks in the area of 300x240 pixels with deadline of 300 ticks** | | | |
| Total reward | 4.9713 (±1.4175) | 15.6611 (±1.7195) | **18.2397** (±1.8068) |
| Elapsed time (norm) | **0.9355** (±0.0683) | 0.9143 (±0.0534) | 0.9067 (±0.0643) |
| Comp. tasks (norm) | 0.0773 (±0.0175) | 0.1741 (±0.0182) | **0.1993** (±0.0193) |
| Quality (norm) | 0.7932 (±0.1053) | 0.9980 (±0.0083) | **0.9993** (±0.0040) |
| Sending token | 47.9600 (±1.9012) | 48.0200 (±2.0100) | **148.6200**(±4.8550) |

to those using the original *stimulus* value in a static context reproduction. Therefore, all final results obtained in this section were based on the *stimulus* = 0.6

#### 3.1.3.2 Second Replication

The second independent replication was done with the extended algorithms (AL, SAL and LAL) previously presented in Section 3.1.2.3. The assessment was done in the same dynamic context used by the first replication (Section 3.1.3.1), following similar original algorithms conditions and variables values.

Table 3.4 shows the total reward obtained with the completed tasks, the elapsed time to perform the tasks, the number of completed tasks, the quality obtained with the completed tasks, and the total number of tokens sent with 100 executions and the same *stimulus* and sensors quality used by the original experiments in all scenarios listed in Section 3.1.2.1.

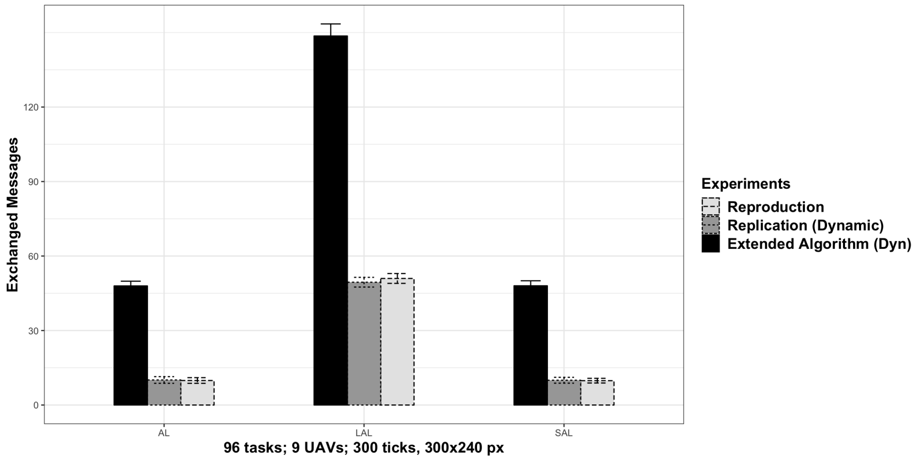



**Figure 3.7:** Exchanged messages of the original algorithm in the static scenario (reproduction), in the dynamic scenario (replication) and the modified algorithms in dynamic scenario (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

The results obtained by these modified algorithms, with small number of UAVs(3) and tasks (4, 8 and 16), presented less difference from the original algorithms in the dynamic scenario (Section 3.1.3.1). As the number of elements is low, the differences are within the standard deviation. Therefore, in the following, the largest group (9 UAVs and 96 tasks) was used to explore the differences in the results.

Overall, the data from Table 3.4 show that after the modifications described in Section 3.1.2.3, resulting in extended algorithms, different results were obtained compared with the original ones in the dynamic scenario. From all assessed metrics, quality and number of exchanged messages were those with more significant variation.

In particular, the number of messages increases over 100% with the modified algorithms, indicating a communication overhead. The exchanged messages in the original and the modified algorithms are depicted in Figure 3.7. This overhead can be explained by the token reset made in the modified algorithms (see Line 7 in Algorithm 4). Resetting the token allows it to run more often among the UAVs, generating more messages exchanged among these elements.

Figure 3.8 shows that the extended algorithms presented an increase of around 40% in the quality attribute compared with what was obtained in the first replication (Section 3.1.3.1). This recovering was due to the task redistribution with the token reset after a context change with a UAV removed from the team. The new token round through the UAVs ensures reallocating tasks to maximize the quality results, relating the task with the agent that has the best sensor to perform it.

Figure 3.9 shows the elapsed time to perform all tasks allocated to the UAVs. It indicates a slight decrease of around 5%, if compared with the original algorithms replication

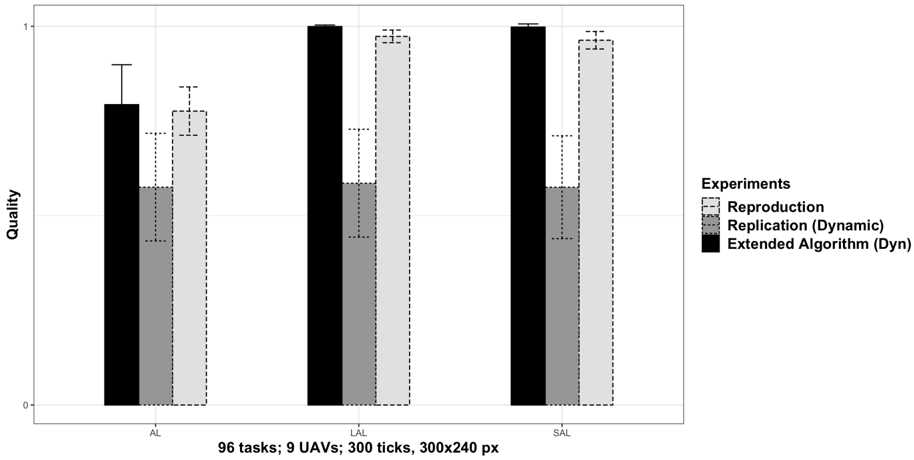



**Figure 3.8:** Quality of the original algorithm in the static scenario (reproduction), in the dynamic scenario (replication) and the modified algorithms in dynamic scenario (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

in dynamic scenario, because of the agents spend more time communicating and adjusting the tasks allocation in order to optimize the quality. It results in less time available to execute the tasks due to the reallocation. However, the difference between the three algorithms (AL, SAL and LAL) in the dynamic context is within the standard deviation as seen in Figure 3.9.

Analyzing the number of finished tasks, depicted in Figure 3.10, it is visible a decrease comparing the algorithms. The lowest one is for the extended algorithms and it occurs because the team spends more time communicating and adjusting the tasks allocation in order to optimize quality thus with less available time for task execution. However, the difference runs into the standard deviation as seen in Figure 3.10.

The confirmation of the statistical significance of the difference of such values was obtained by a t-Test, which was employed because the results distribution is close to a normal one, as show in Figure 3.11. Thus, it is confirmed that the average quality and messages results obtained by the original algorithms and the modified ones are different. The test confirmed the difference with a confidence interval of 0.99 and $p - val < 0.05$.

For this replication, the applied *stimulus* value was of 0.6. It is the same value applied by the original work and the first reproduction done. Such a value balances the decision making for performing the task by making a trade-off between the distance to the task and the sensors' compatibility with it. Schwarzrock et al. [108] and Ferreira Jr et al. [51] justifies this value with several experiments to evaluate results.

As it was done in Section 3.1.3.1, different values to the *stimulus* attribute were tested. However, when the results obtained with different *stimulus* values were compared, the variation stayed into the standard deviation. To allow a faithful final comparison in this

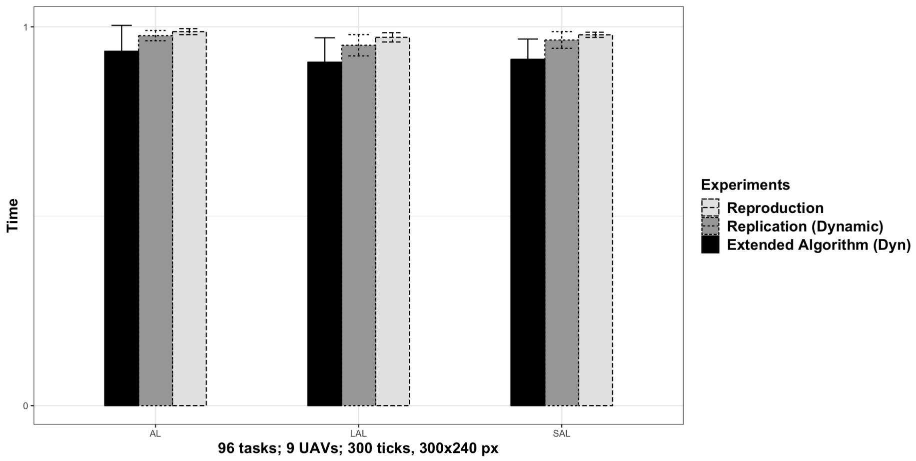



**Figure 3.9:** Elapsed Time in the dynamic scenario (replication) to the original algorithm in static scenario (reproduction), in the dynamic scenario (replication) and the modified ones (96 tasks; 9 UAVs; 300 x 240; 300 ticks)

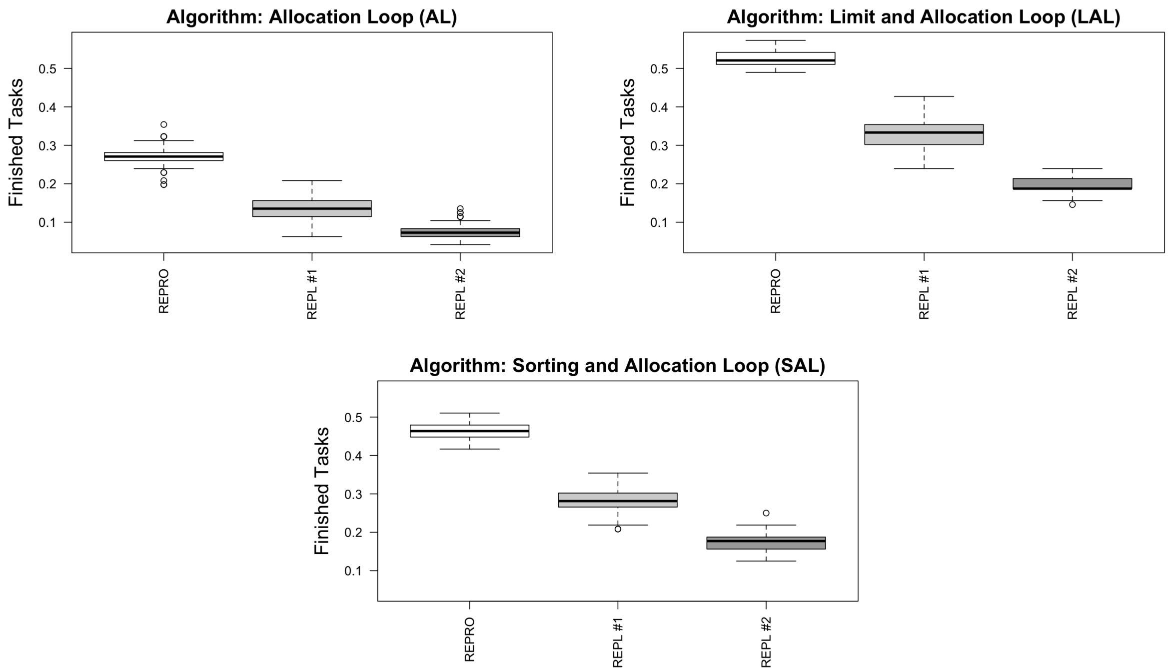



**Figure 3.10:** Finished tasks to the original algorithm in static scenario (reproduction - REPRO), in the dynamic scenario (first replication - REPL #1) and the extended algorithm in dynamic context (second replication - REPL #2) with the attributes: 96 tasks; 9 UAVs; 300 x 240; 300 ticks and 100 executions

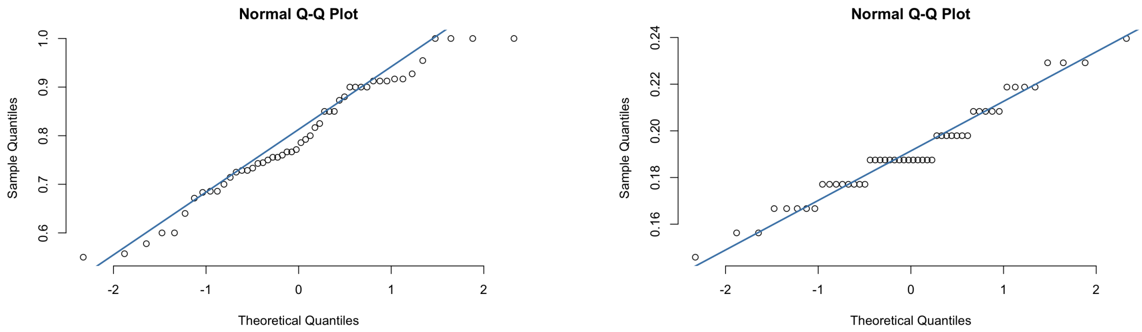



**Figure 3.11:** Distribution of finished tasks (left) and quality (right) results compared with the Normal Graph (Extended SAL algorithm; 96 tasks; 9 UAVs; 300 x 240; 300 ticks; 50 executions)

work, the value used by the original work [108] of 0.6 was also used to this independent variable (Section 3.1.4).

### 3.1.4 Exploratory Discussion

This section aims to identify the main aspects found during the replications done and to analyze all obtained results (Section 3.1.4.1), and identifying its implications for practice (Section 3.1.4.2).

#### 3.1.4.1 Findings

The replications performed with the original and the extended algorithms, in the dynamic scenario, suggested higher results difference in quality and number of messages sent metrics. These highlighted differences justify choosing these two variables to analyze the algorithms behavior and the emerging trade-off, in that new dynamic scenario. The messages exchanged represent the token being sent to the next agent in the communication network.

The quality of the sensors related to the tasks guarantees that the agent is the most suitable one to perform it. Figure 3.3 shows a significant quality results decrease with the original algorithms in the dynamic scenario. On the other hand, Figure 3.8 shows that extended algorithms [13] increased this variable results, even with a reduced number of performed tasks, bringing the level to the same original baseline.

As the first result of this preliminary action research, it provides evidence that the original algorithms proposed in Schwarzrock et al. [108], and based on Swarm-GAP intelligence, work in dynamic scenarios. However, the results obtained for all variables with the proposed extended algorithms [13] were not in the same level of what were obtained by the original study [108] in a static context. All dependent variables presented similar values in both replications, except the following for the original algorithms [108]:

- **Quality**: when the UAVs are disabled (representing a shut down aircraft), the tasks allocated to it are not performed and the quality sum of completed tasks decreased ($\approx 40\%$) as there are fewer agents;

- **Capability**: as the agent capability is calculated using the sensors quality, and it decreases as explained above, this variable decreases by $\approx 40\%$;

- **Number of finished tasks**: having fewer UAVs to perform the tasks, it is natural that the number of finished tasks decreases. This variation was $\approx 50\%$ in most cases.

- **Elapsed Time**: the time spent to perform the mission was shorter due to the fewer number of UAVs. The remaining UAVs always use the maximum possible available resource to perform the tasks. However, in general such resource exhausted before the total available time. Thereby, the elapsed time reduced in $\approx 15\%$, particularly in LAL and SAL algorithms. The results in those algorithms fall off the standard deviation compared with the original experimental results.

Even with the variation above, the algorithms presented by Schwarzrock et al. [108] solution can be applied in case in which there is no other strategy available to deal with a dynamic scenario. However, there is evidence indicating possibility of algorithms improvement.

In this vein, the second contribution is the improvement proposed to those algorithms to deal better with the proposed dynamic scenario. The extended algorithms presented in Section 3.1.2.3 was submitted to an independent replication and the results were similar to the original algorithms applied in dynamic scenarios except for following metrics:

- **Quality**: this metric presented a significant result recovery ($\approx +40\%$) showing a level equal to the original algorithms [108], which was carried out in a static scenario;

- **Number of exchanged messages**: the extended algorithms require more exchanged messages due to the token reset. This characteristic causes an increase of 100% in the number of exchanged messages compared to the original algorithms [108] in the dynamic scenario;

- **Number of finished tasks**: the number of finished tasks reduced up to 50%, compared to the original algorithms [108] in the static scenario, due to the required time to resend tokens when something in the dynamic context occurs, i.e., when there is a an UAV removal. With this, there is a fewer number of available UAVs in the experiments.

It is possible to notice that, even with a reduced number of finished tasks, the total quality increased due to the tasks reallocation that occurs when a UAV is shut down. This operation always occurs getting the maximum possible quality level based on the available resources, associating the most suitable sensors to the tasks.

Thus, it is possible to increase quality results if the network aspects are not an issue and the communication structure can support the demand required to execute the extended algorithms. Indeed, it was also possible to identify a message traffic three times higher than the original one due to the communication need among the agents using the extended algorithms, which provides evidence for the network capability requirement. Besides, this investigation [13] provides evidence that there is a trade-off among the quality and the number of exchanged messages when the extended algorithms here proposed are used in dynamic scenarios.

In summary, this exploratory study, characterized as an empirical work, provides evidence that there is a trade-off among the quality and the number of exchanged messages when the extended algorithms here proposed are used in dynamic scenarios as described in Section 3.1.2.2. The scope of this study is relevant because the dynamic created scenario use assumptions closer to the real world operation scenario, presenting a more realistic (and useful) way to perform tasks allocation.

#### 3.1.4.2 Implications for Practice

As explained in previous sections, the existing trade-off between the communication overhead and quality results is an important aspect to be considered when choosing which algorithm to use. The right choice depends on what is the priority. If the allocation needs to be the best according to the sensors available, it is necessary to know that there will be an overhead in communications among the agents.

In that way, it is necessary to know the network capacity in order to support the messages exchange. Thus, the simulation shows that this aspect is a mandatory requirement to explore the advantages of the modified algorithms. If the communication structure is not good enough or there is no evidence that it supports high volume of messages, it is appropriate reduce the quality result, but to keep a certain level of mission accomplishment.

Based on the performed replications, a real scenario described by a set of targets to be photographed can be an example of these algorithms application. Thus, it is necessary to choose if it is more important to have best pictures with the most suitable cameras according to the type of targets, or it is necessary to keep the communication structure not overloaded with high quantity of messages (tokens). This example represents a real possible situation where this trade-off needs to be properly analyzed to obtain the best results according to the demand.

### 3.1.5 Partial Evidences

Since the original work reported in Schwarzrock et al. [108] is classified as a reproducible research [89] and confirmed by the reproduction presented in Section 3.1.1, we performed two independent replications, using NetLogo 5.3.1 environment, to collect results and obtain evidence about limitations and improvements opportunities proposed by the present study.

Closer to real world, a dynamic scenario was defined in place of the original static one used in Schwarzrock et al. [108]. Moreover, a first replication was performed to validate that the original algorithms proposed by Schwarzrock et al. [108] are fully functional in this new dynamic scenario. This experiment showed an improvement opportunity due to a decrease in some dependent variables, e.g., capability and quality.

Proceeding an action research and based on results obtained by the original work, extended algorithms were proposed to better address the new created dynamic scenario. The second replication assessed these extended algorithms and allowed the identification and discussion of emerging trade-offs. The metrics of quality and exchanged messages were the focus of these analysis due to the significant difference presented by the extended algorithms in these measurements compared to the original ones.

The proposal presented in this exploratory study has, as the main idea, that all tasks releasing after the team looses an agent, and the token resetting to allow a new turn of allocation execution steps. Thus, the tasks not completed can be reallocated always looking for the quality maximization based on the relation among the task distance and the sensor suitability to its performance. There is evidence that this new procedure requires more communication since the number of exchanged messages increased significantly face the original proposal.

The main discussion is the collateral effect caused to obtain an increased quality. This effect is an increase in the exchanged messages that suggested the idea about the need of a network requirement level to apply the proposed algorithms. However, more detailed measurements of network level requirements are left as suggestion of future work. Furthermore, the communication structure used, initially a token ring network, can impact in results and different topologies can be tested in the future.

As this study considered only a perfect communication channel, it is an opportunity for future work to improve system's resilience via a mechanism capable of dealing with communication issues. Another improvement opportunity is to model and to implement a scenario with more dynamic elements, e.g., number of tasks and sensors status, which would make the model even closer to the real-world scenarios. In this work, only the number of agents was changed, but in a real military operation scenario it could have mission/tasks or agents capabilities changing during runtime. With this new context, additional impact may be observed and require variations of the proposed algorithms or even in the original Swarm GAP strategy. In addition, a promising work is to combine the allocation solution here proposed with other expert system proposals, e.g., robots application or navigation. This way, it is possible to apply a better task allocation strategy and to improve the autonomy of such systems.

Further empirical assessment has to be performed to evaluate trade-offs among the applied task allocation algorithm and the network structure, e.g., ring in the Swarm-GAP and its extensions. Finally, all current and future directions of research rely on extensive usage of empirical studies involving simulation. The correct setup, execution, and analysis of such studies is non-trivial, error-prone, and time consuming [113], thus requiring further development of supporting tools.

The results obtained during this exploratory study leverage the opportunity to present a more efficient approach to handling dynamic scenarios. Thus, we use the allocation algorithm proposed in this work within a structure that incorporates the C2 agility concepts. Such a structure resulted in the models presented in the next chapters of the present thesis.

## 3.2 C2 Approach as Network Topology

We use the C2 approach dimensions for exploring C2 agility concepts in our proposal. According to Section 2.1, we relate C2 approach with data exchanging and decision making. Basically, such elements provide coordination and we address them with the network topology, protocol, and message content exchanged to get awareness. To represent these elements in a simulated environment, we define the C2 Approach with the network topology adopted by the team.

To define which topology is related with which operated C2 Approach, we analyze the characteristics and properties of each C2 approach to identify similarity with the properties of network topologies. Figure 3.12 shows key elements for the dimensions of each C2 Approach.

Analyzing the *Conflicted C2*, we observe that the only element is related to the distribution of information, where in such a case, the entities work only with organic information. This characteristic shows no data coming from outside. Based on this, the topology that can be used in such a representation is the isolated structure, i.e., no connections among the entities.

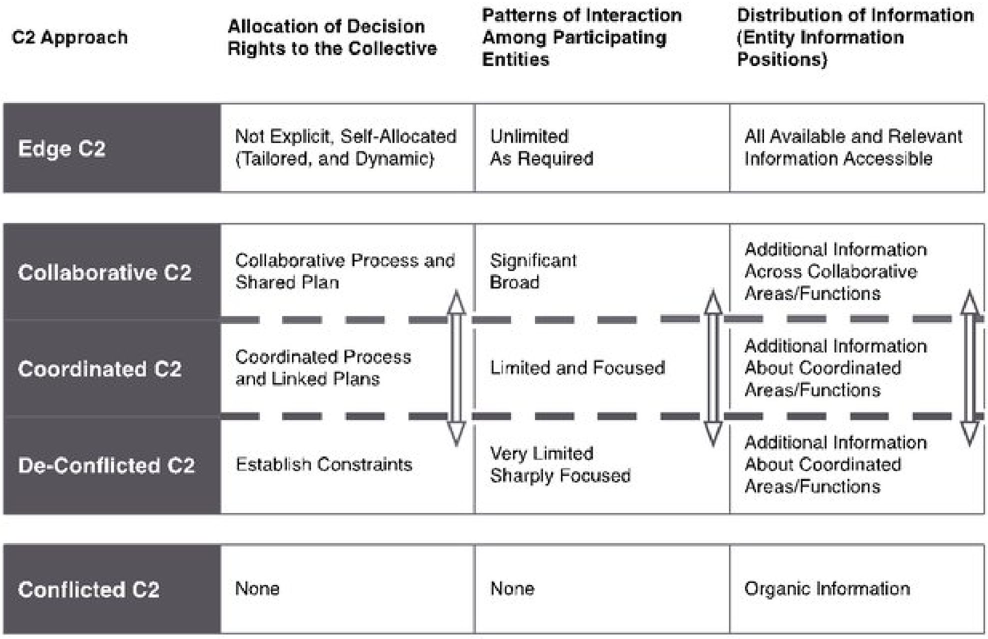



**Figure 3.12:** C2 Approaches and their relation with C2 Approach Space dimensions (from Moffat et al. [94])

The first element of this table is the *Edge C2*, which works with the maximum volume of information available. Besides, the interaction among the entities is unlimited and according to the requirements. The entity itself defines the decision rights and according to the accessible information. A network with all elements connected to each other can represent such a structure. This fully connected network topology is suitable to guarantee data exchanging required to the entities operate the C2 approach edge.

*Coordinated C2* works on coordinated areas of communication, with one or more coordinator. Such elements are the center of a star structure from which all entities around have a link to the central node. This coordinator has the responsibility of decision rights distribution and action planning. This C2 approach reminds us of a topology with a central node, forming a structure so-called star.

We perform a similar analysis with the *Collaborative C2*. In such a case, we have small groups under coordination sharing information and perception. Scale-free networks can represent enormous and complex structures with such a strategy. An extreme distribution of interactions among their nodes characterizes these structures, with a few nodes having a very large number of interactions. These nodes represent the coordinators of each group of nodes. This network does not preview any node with no connection. We can address this C2 approach with a network topology where we have a set of coordinated sub-networks, and we link each central node to all other coordinators.

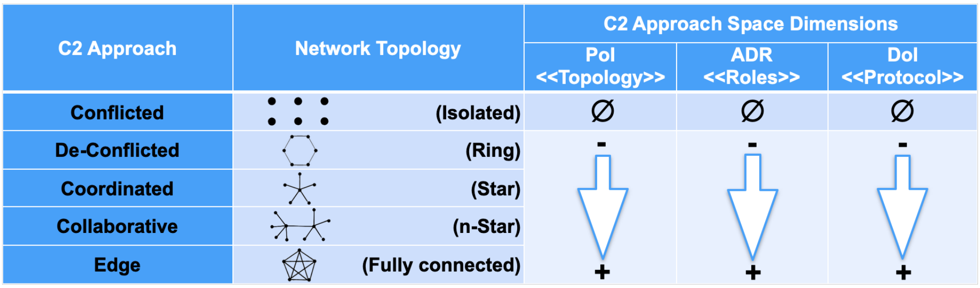


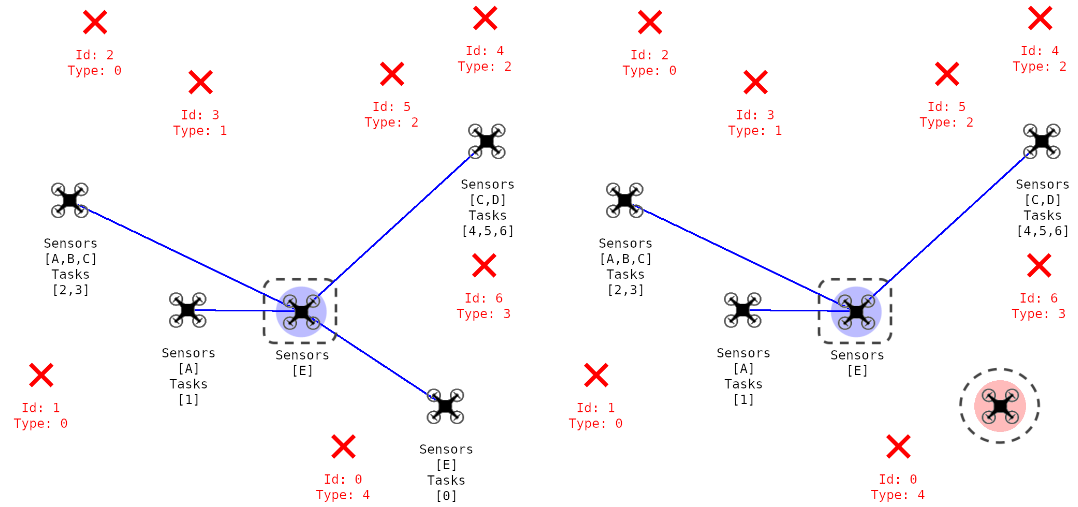

**Figure 3.13:** Network topology applied to each C2 Approach and its impacts on each dimension of the C2 Approach Space

Finally, the *De-Conflicted C2* approach works under a set of constraints that limits their capacity of information and data sharing. The pattern of interaction in such an approach is limited and defines a few connections among the entities. In such a case, we are considering each entity having only two connections, creating a ring structure. Based on this, we address the *De-Conflicted C2* approach with a ring network topology. Figure 3.13 shows the topology for each C2 approach. In summary, we can observe an increase of the C2 dimensions values in a C2 approach more connected and, a better awareness.

## 3.3 Problem Description

To motivate the addressed research problem, we present a simplified example of a mission execution by entities. Figure 3.14 illustrates a reconnaissance mission where a team of four UAVs is interacting in a network configuration with a central coordinator (shown in Figure 3.14(a)), which only provides instructions to its subordinates. These interactions and network topology establish the Coordinated C2 approach [6]. The coordinator, depicted with a dashed square around it, guides the other team members to complete mission tasks. Each task has a characteristic, and it requires a specific type of sensor for obtaining aerial images of particular points of interest, represented by red crosses.

The mission has some inherent risks that may lead to a change in the conditions of execution. In this example, one member of the team, marked with a dashed circle (Figure 3.14(b)), dropped because of some environmental change, such as an intense storm, causing damages to its motors. Losing one entity, depicted in Figure 3.14(b), might decrease the effectiveness and quality of the mission execution. So, a new plan is required, considering task *id 0* originally assigned to the dropped drone. Otherwise, there would be a lack of C2 agility. The absence of a system strategy to maintain or increase C2 agility may compromise the amount of completed tasks at the end of execution.




**(a)** Coordinator (dashed square) **(b)** UAV dropped (dashed circle)

**Figure 3.14:** Team of entities related to a central coordinator (dashed rectangle) performing tasks at predefined locations (red crosses). A context change results in an UAV dropped (on the right, marked with a dashed circle).

In general, context changes occur in the entities themselves (self) and in the environment, e.g., UAV failure, sensor damage, or weather change, and they must be considered during the mission planning.

Real scenarios, where there are entities applying the resources available to accomplish a mission or to reach a goal, naturally have embedded dynamism [3, 10]. Any adaptation to the new context may decrease quality results because of an incompatibility between entities and mission, or insufficient resources to complete the mission, or even the inability to meet minimum quality acceptance level. Therefore, the C2 Agility concept introduced in Section 2.1.4, composed of C2 Approach Agility and C2 Maneuver Agility, must explore the configuration and coordination for suitably responding to deal with dynamic context. Based on this, this work considers a refined definition for C2 Agility as the following capability of a team in a mission:

> C2 Agility: For all tasks $t$ of a mission, it is possible to find a C2 approach $\omega$, such that a team $E$, i.e., set of members, becomes constrained to operate under $\omega$, in a way that $t$ is allocated to a member $e \in E$, and $e$ adopts an internal configuration $c$ among its valid configurations, such that this configuration makes $e$ capable of dealing with $t$.

In other words, a limited C2 agility might compromise the mission accomplishment in a dynamic context. Nevertheless the state-of-art and the state-of-the-practice do not explore methodologies or strategies to provide C2 agility, especially considering context changes [10]. Indeed, to provide C2 agility in dynamic scenarios in which members are performing a mission, it is necessary to guarantee efficient management of resources available to deal with context changes. Extremely dependent on the application domain, exploring terms and specific definitions of the application scenario, last studies show low maturity of C2 agility.

We derive such problem considering the C2 agility concept presented in 2.1, where C2 agility is a combination of C2 approach agility and C2 maneuver agility. Based on this, we explore the C2 approach agility and C2 maneuver agility separately, making the association with entities' configuration and coordination, respectively. Specifically, we consider this problem in the simulation environment scope. This scope, which is often used to study C2 in its common application areas [6], is relevant to explore many other scenarios such as Network Centric Warfare and telemedicine [6, 9, 112].

Based on the state-of-the art and the practice on C2 agility, considering the simulation context of *in silico* and *in virtuo* experiments [122] applied in the military domain and reported in last works, our study derives the problem shown in Section 1.1 in the main research questions as follows:

- (RQ1) How to provide C2 Approach Agility ?
- (RQ2) How to provide C2 Maneuver Agility ?

## 3.4 Assumptions

Realistic scenarios involving C2 application are extremely complex because of an enormous number of elements and variables that compose it and they may change drastically in runtime. The unpredictability becomes constant. The circumstances change and it becomes the model's complexity high and is mandatory to limit some possibilities and to consider some assumptions. In addition, all C2 concepts have a wide application in many domains, e.g., financial, military, and natural disaster relief (cf. Section 2.1). Indeed, all these domains present similarities, but some particularities could conduct our study to biases. Based on this, we have chosen the military domain as a reference to validate our proposal and to define some terms and processes. We based this choice on the domain knowledge of this work's researchers and collaborators, becoming easier the access for some information and staff. In addition, a documented and well-defined process follow by these domain experts helps us the information collecting.

The fundamental assumption to be considered is that the system is always operating a C2 approach. Even when a critical issue occurs, the team adopts at least the *Conflicted C2* approach in runtime to keep mission execution. A manager role, i.e., the entity playing the manager role in the C2 approach structure, performs this selection. The communication protocol adopted makes the members look for a manager in order to receive the information about which C2 approach must be operated. The entity with such a role can decide by itself or can receive orders and definitions from a central command. Both information sources are compatible with the military domain rules theory described by Alberts et al. [5, 6]. Exploring the C2 dimensions, we can have different ways of interaction between the members and the autonomy level of each one. Since the principles of C2 is to adopt different portions within the C2 Approach Space in order to deal better with the context changes, the entity responsible to define the C2 Approach to be operated makes this choice based on the information collected from the surrounds and from another team's entities.

Even if we represent entities' behavior in a dynamic context, modeling time constraints is outside of the present study scope and we consider it as a future work. In addition, last works related to C2 do not explore such a temporal aspect because of the natural complexity involved and the limitations of this field. However, such works that analyze entities' behavior in simulated and realistic scenarios, not considering time constraints, have proven to be relevant in several domains, e.g., military [1, 6, 124].

The environment is the most complex element in the C2 scenario, because it concentrates all effects outside from the entities, e.g., hazard caused by enemies' presence, weather, terrain, visibility, etc. These effects are from an infinity set, we face with a high level of uncertainty. To simplify such complexity, our study translates all environment effects as entities' perturbation, e.g., onboard sensors capacity reduction. In summary, we consider the following assumptions related to the C2 Approach selection in this work:

- We omit the C2 approach selection process because of the possibility of connecting any selection strategy and technology, e.g., artificial intelligence, big data, or a doctrine;
- We call C2 System as the structure composed by all elements operating a C2 approach, i.e., entities, mission, and environment;
- We cross the C2 approach space in its diagonal, starting from a total isolation to a complete connectivity and high autonomy in entities' decision making;
- The C2 Approach choice depends on the performance requirements because this process is time-consuming during the possibilities analyzing;
- The C2 approach operated defines the logical communication structure and the information exchange chain, i.e., the network protocol and topology;

- All elements distinct from mission and entities compose the environment;
- We consider no limitations to establish the network connections, providing all required conditions to select any C2 approach selected by the manager; and
- We consider the C2 approach space discrete with the five positions described in Section 2.1.

In addition, we abstract the system status monitoring mechanism, i.e., the process to check entities' status during execution. The MAPE loop [18] implemented within the members as DSPLs (cf. Section 2.2), has in its monitoring phase the engine responsible for performing such a process using a standard and simple solution not focused on performance. We fed this monitoring mechanism by the information coming from the entities' communication. There is evidence that it is easy a replacement for an improved solution to get better results in such a phase.

Based on the exploratory study described in Section 3.1 , we use the token-based communication protocol in the simulations. Besides, the token carries the information about the entities status, helping the monitoring mechanism implementation. This process, compared to a ping command, checks entities' status during the execution, sharing the communication package with monitoring information. Such information runs in the system, respecting the connectivity rules and topology defined by C2 approach. The information within the token is used to define which approach is the most suitable to deal with current circumstance. In a realistic scenario, we can use a dedicated channel to exchange all required information to the system operation.

Finally, as we base the proposed models on roles, an entity can play multiple roles simultaneously within the team. In such a case, distinct entities have the same allocation decision rights and, to avoid orders conflict, they exchange information among them to get a suitable awareness. The present work omit such a secondary communication step from the models because of it does not cause the proposed model state change. However, the simulator used to assess the models in this work uses the token structure to perform such a data exchanging. Similar to the previous assumptions, we can adopt different strategies to implement this requirement, looking for a better result in the mission execution.

# Chapter 4

# C2 Agility Models

To provide C2 agility, we present two complementary computational models capable of representing the scenario members' behaviors based on the roles that such elements have. These models are based on computational environment and evaluated through software simulation. In terms of software domain, we consider the agility as the ability to create and to respond context changes in order to succeed in an uncertain and unstable environment, so-called responsiveness according to Gren and Lenberg [62]. Software engineer and adaptive systems architecture use such a definition in their theory.

On a coarse-grained level, configuration refers to a new system structure with no C2 Approach change. Differently, *coordination* represents how these elements exchange awareness, coordinate their operations, or collaborate with each other, i.e., different C2 Approaches application. In such a level, we use an extension of the Multi Product Line (MPL) definition to create a proper structure of monitoring, configuring, and acting. On a fine-grained level, *configuration* refers to the ability of system members to adapt themselves according to new requirements, conditions, or circumstances. Therefore, configuration and coordination both contribute to C2 Approach Agility and C2 Maneuver Agility (cf. Figure 2.5).

Based on this, we describe the rational to relate software to C2 agility definitions in Section 4.1. The Section 4.2 uses such definitions to describe a static model that supports the representation of the main C2 system's elements and their relation and dependence. The elements behavior using channel system theory is defined by the dynamic model shown in Section 4.3.

## 4.1 Software Approach to C2 Agility

The combination of C2 and software domains is innovative, and it requires a suitable approach because of the broad concepts that C2 involves. Humans aspects and uncertainty are aspects existing in C2 context. We deal with such aspects in software domain, focusing on the ability of perceiving a context modification and acting to provide a suitable change to deal with dynamic scenarios. These abilities define C2 agility in C2 domain (cf. Section 2.1.4).

To satisfy the C2 agility requirements and in line with Self Adaptive System (SAS) [12], our proposal relies on the software capacity of adapting to deal with requirements change in runtime. Relying on different valid products generation from a common artifact set, just activating or deactivating features, to provide context dealing, the Software Product Line (SPL) approach shows itself is suitable for dealing with such a variability. Such a capability and usability make the SPL as an efficient strategy to deal with varying requirements and constraints [64]. However, the system's configuration capability must be in runtime. Based on this, we leverage Dynamic Software Product Line (DSPL) as a strategy for implementing an SAS. Such an approach provides the self configuration capability to the system. This strategy addresses the C2 approach agility described in Section 2.1.4. However, C2 is based on collaboration. Based on this, we propose the use of a set of collaborative DSPLs to represent the team of entities.

We extend the Multi Product Line (MPL) [67] concept, originally defined to be used with static Software Product Line (SPL) components, to insert coordination capability among the DSPLs. We call this structure of *Multi Dynamic Software Product Line* (MDSPL), which is represented by two *quasi-orthogonal* [102] dimensions so-called coordination and configuration. Figure 4.1 shows such a structure combining C2 elements with Dynamic Software Product Line (DSPL) [26] definition.

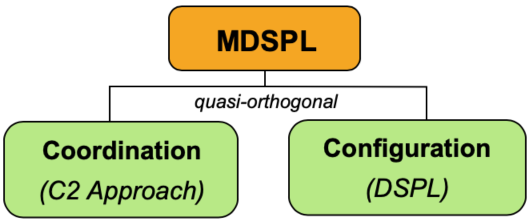



**Figure 4.1:** MDSPL composed by the configuration and coordination dimensions that can complement each other

The term *quasi-orthogonal* expresses a partial dependence among these two dimensions, where the changes in any of them may cause reactions in the other. The configuration capability obtained with the DSPL approach can require a coordination change through a suitable C2 approach adoption. Similarly, a different coordination can require some configuration change. However, this collateral effect is not mandatory, and it depends on the requirements. In addition, we use the principles presented by Kephart and Chess [78] applying the definition of orthogonality in self-adaptive systems.

Figure 4.2 shows the structural view of the MDSPL conceptual model. A MDSPL is goal-oriented, guided by Quality Attributes (QAs) [90]. Two combined MAPE [17] loops comprise this structure. Each one takes care of C2 agility dimensions, i.e., coordination and configuration. The proposal of Bencomo et al. [26] inspired such an approach, with a layer related to domain evolution and another one related to product evolution. Both levels have the results improved with the knowledge step ($K$), providing information from previous scenarios, doctrine, big data or any IA component connected. To simplify our representation, we use only one MAPE loop to represent the application and domain engineering in a DSPL representing our configuration level.

At a high and structural level, our MDSPL model of the C2 system enables coordination of capable members to accomplish a mission. This level represents the coordinator behavior, responsible for organizing the team's entities and for defining the C2 approach to be operated. Such a coordination provides the team's awareness and the most suitable entities' configuration according to the mission. It can be within multiple entities. The lowest level played within the entities looks for providing self adaptation to deal with context changes. It represents the DSPL by itself and it receives information from the highest level through channels. The information exchanged between these layers is part of the awareness procurement process of the system.

We focus on the ability of perceiving context modification and acting to provide a suitable change to deal with such a scenario. Each team's member is a DSPL with a reconfigurable behavior addressing a subset of the mission tasks with prescribed quality levels. The members organize themselves in a way to share information and to perform a decision making about the task allocation. The DSPLs are goal-oriented [26], looking for satisfying the mission requirements and reconfiguring for keeping running and compatible with the tasks to be performed. These abilities define C2 agility (cf. Section 2.1.4).

In summary, to provide C2 Agility to this model, the MDSPL must first be able to identify changes in circumstances. For example, members' onboard sensors perceive new environment conditions. Members are also aware of their own status, e.g., UAV's battery level. At the global level, there is a list of tasks to be performed that characterize the mission. By leveraging the reconfiguration capability of each member and the team's coordination capability, eventually performing task allocation and/or changing the C2 Approach, in all cases guided by QoS constraints, the MDSPL of the C2 System provides agility.

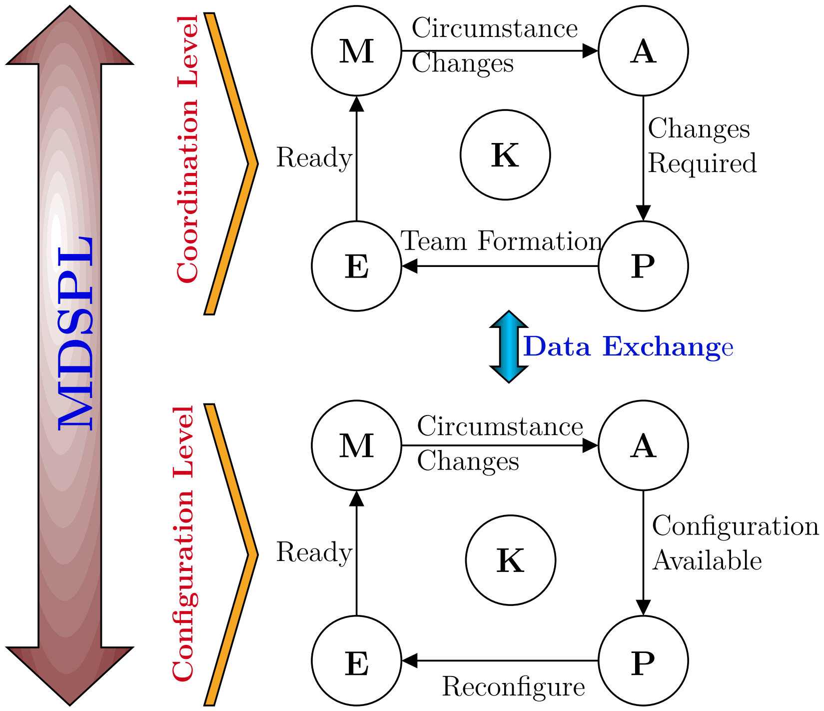



**Figure 4.2:** The quasi-orthogonal dimensions of MDSPL represented by two combined MAPE loops

## 4.2 Static Model

We base the proposed C2 system on four key concepts: *entity*, *role*, *mission*, and *C2 Approach*. The C2 Approach coordinates the entities and provides awareness to them. We use a meta-model as a static model to complement scenario information according to domain requirements [44]. Figure 4.3 shows such the C2 Meta-Model (C2MM) to capture C2 system elements and their dependence focused on goal fulfillment. We use Obeo Designer Community modeling tool [98] to draw the model and to create the instances from it that represent a snapshot taken during the team's performance on a mission and their configuration.

The C2 approach selected defines the connections set and the roles available to the team members. Such connections can be between members, i.e., intra, or between teams, i.e., inter. Such connections define the network topology under which the agents and teams exchange information and obtain awareness. The protocol applied to perform this communication can be any satisfying the roles based on organization and C2 Approach dimensions.

Each *entity* can play one up to three roles simultaneously according to the roles associated with the C2 Approach operated by the team. The role assignment occurs in all levels, i.e., teams and members, and in case of teams the role assigned represents an abstraction which is effectively decomposed by a collaboration of finer-grained roles carried

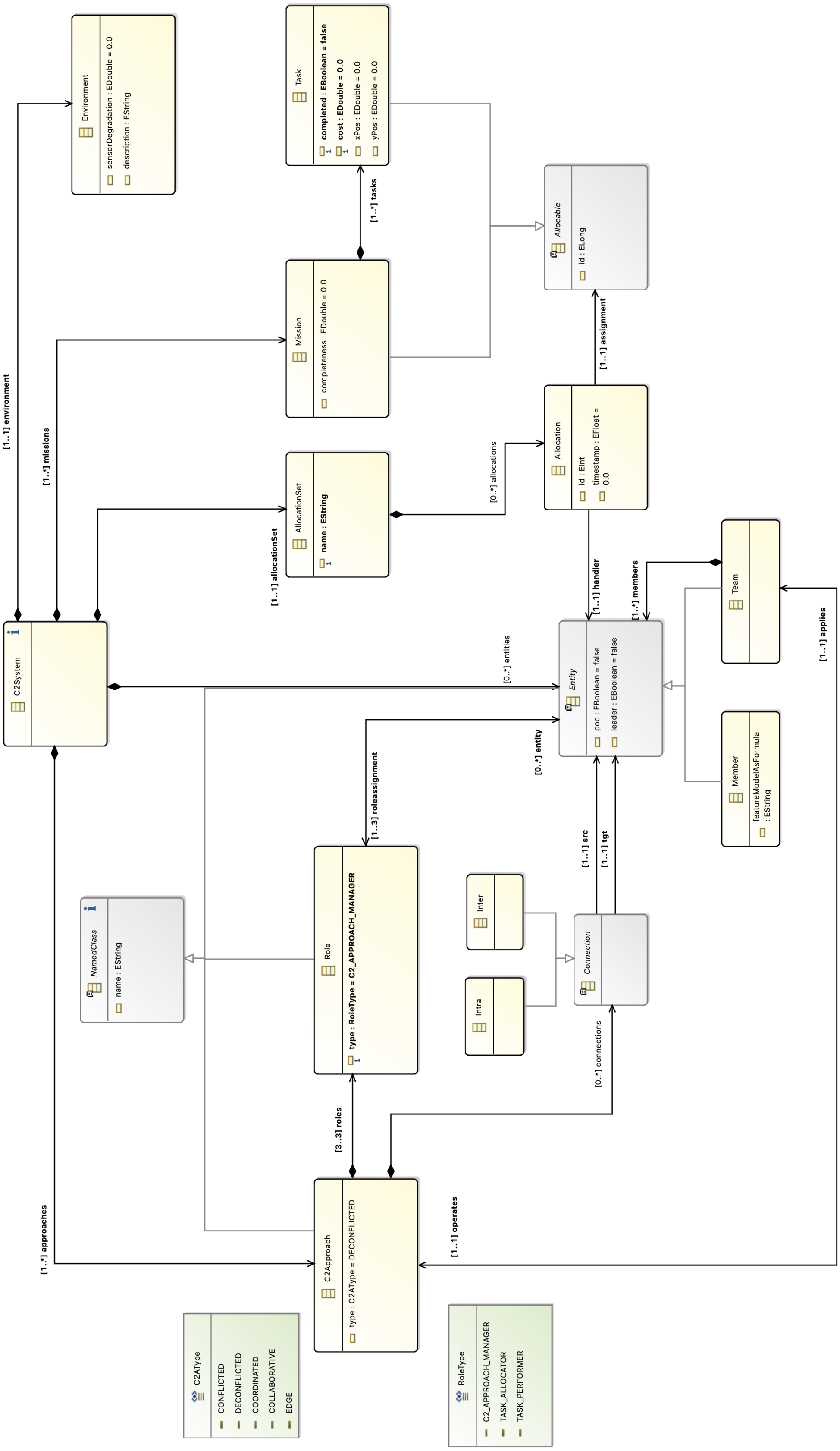



**Figure 4.3:** C2 System meta-model(C2MM) with the roles that compose the C2CS

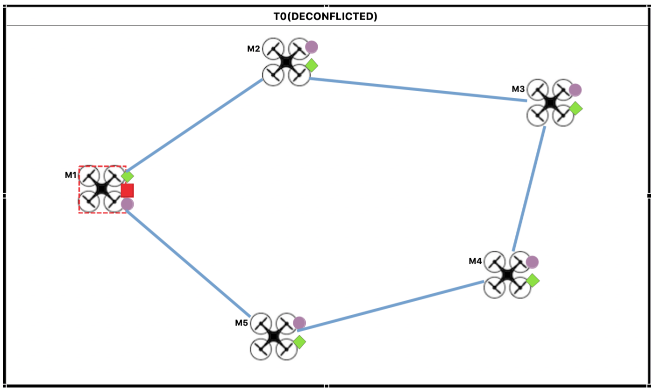



**Figure 4.4:** A C2MM instance representing a simple team with five members and their roles

out by the members making up the team, recursively looped through all levels.

To provide the right connections and responsibilities to establish them, defining link's source and destination, there are two connectivity roles defined as an entity attribute, so-called point of contact ($PoC$) and leader. $PoC$ member defines the connection point between teams. The *leader* member concentrates and organizes the internal communication among the team's member. We can have multiple $PoCs$ and *leaders* in a team, depending on the C2 Approach and the network topology requirements.

The metamodel provides richer information about the C2 system to leverage the simulation implemented. Figure 4.4 shows an instance of the C2MM containing five members with their respective roles. The blue line shows the communication topology that describer the C2 Approach operated, i.e., De-Conflicted C2 Approach. The icons on the side of each *entity* represent the roles. A red square represents the C2AM role, TA by a purple circle, and TP by a green diamond.

As a simplification, this team has only the *leader* because of no communication performed with another team. The member M1 surrounded by a red dotted square show in Figure 4.4 is the *leader* that starts the communication and defines the package shipping order. Normally, *leader* comprises members with the C2AM role. The entities with this attribute activated organize the internal communication and the protocol used. An example to show the leader's acting is in the collaborative C2 approach, where each group of entities has a leader to concentrate the information and to create small cells.

We generated such instance from the C2MM providing complementary information to define the C2 system structure in operation. However, the metamodel aims to model more complex structures, specially when we have teams of teams. In such case, the $PoC$ *entity* connects these teams. Figure 4.5 shows a complex structure where we have a team made up of three internal teams, one of which is also made up of three other teams. The $PoCs$ and *leaders* are surrounded by a red dotted square and, to simplify the instance, put in the same *entity*. In such an example, the team T3.2 is the *PoC* and *leader* of the T3, but the actual execution of these functions is performed by the member at the finest-grained level, i.e., member A3.2.5.

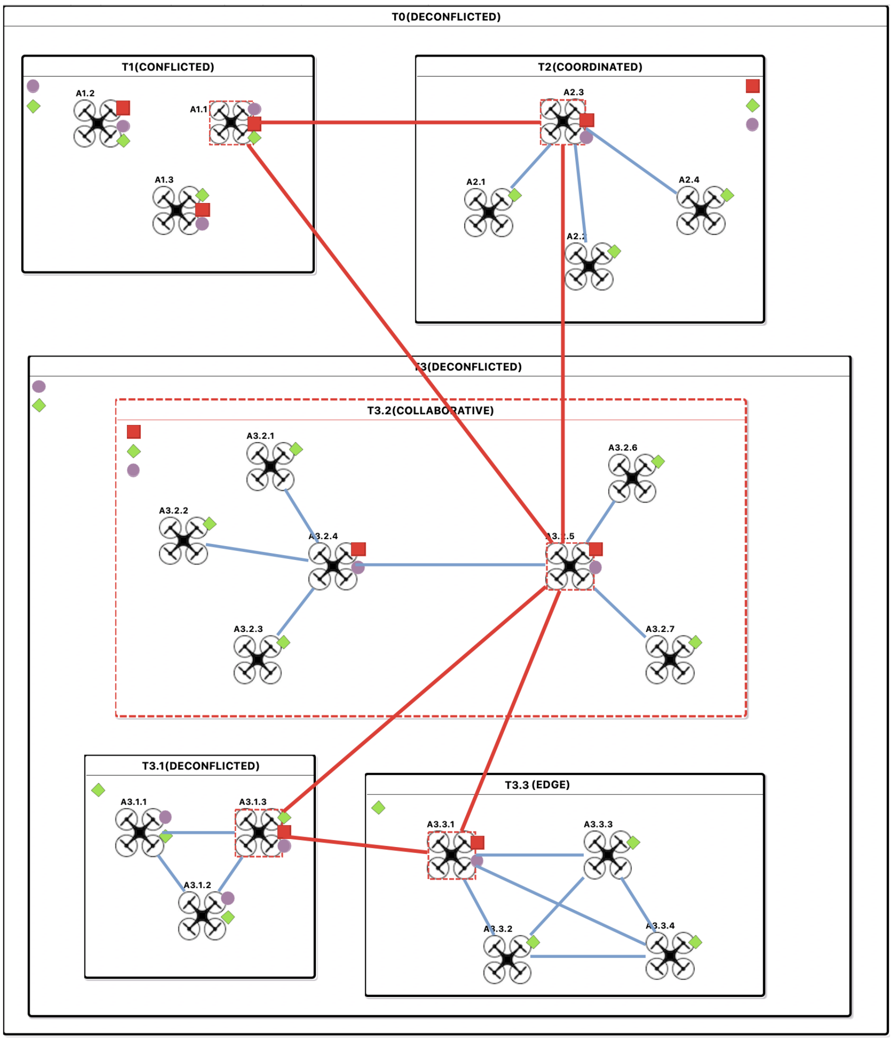



**Figure 4.5:** A C2MM instance representing a hierarchical scenario.

The entities with TA role perform the tasks allocation, and this operation updates the list of tasks associated with an *entity*. The C2 System has an allocation set, i.e., class *AllocationSet* in the metamodel, which groups the set of allocations that link the teams and the agents to the tasks. We can allocate one task or a set of tasks, so-called mission. We are using the name mission to define a set of tasks. The C2 System can have one or more missions composed of one or more tasks. The global mission is the sum of all these partial missions as packs of tasks. Such an aspect guarantees the ability to allocate single tasks, a set of tasks, or an entire mission to an *entity*. This allocation changes at runtime according to the resources available and the algorithm executed by the TA role, and changes the list of tasks associated with each *entity*. The *timestamp* attribute in the *Allocation* class shows the allocations order to track such changes executed by the TA *entity*.

Basically, the metamodel proposed can represent different scenarios through instances generated with its elements. These different scenarios result from context changes that cause perturbations and modifications in one or more C2 System elements. Figure 4.6 shows another instance from the metamodel with the previous composition, where the C2 Approach of the team T2 did not change, but the responsibilities of some members have changed. Some problem with the member A2.3, e.g., onboard resources limitation, may have caused this scenario. In such a case, coordinator of external and internal communications, i.e., *PoC* and *leader*, is switched from the defective member to the A2.1. The same occurs with the C2AM and TA roles that need to be assumed by another element and, in this case, it concentrates all in the A2.1 member.

In addition, Figure 4.7 shows an extended scenario from the previous one where member A2.3 has its battery drained. Such an issue completely removes the member from the team's composition. However, as it was not responsible for the connections with other teams and agent within its own team, and it has not a role that needs to be transferred to another member, it is simply disabled and excluded from the scenario. In such a case, no C2 Approach change is required.

However, some context changes require C2 Approach change, with a resulting migration of the connectivity's responsibility or tasks reallocation. Figure 4.8 shows another instance from the evolution of the previous scenario where a member loss in the team T3.3 was identified and a C2 Approach change was required for the awareness reestablishment. This change requires a new *PoC* and *leader* to reorganize the communication links. They redistributed the roles according to the new C2 Approach selected, i.e., de-Conflicted, and creating a ring topology with the team's members.

It is important to note that the C2MM supports two complex features: the representation of hybrid teams through entity detailing and the possibility of representing hierarchical teams. The hybrid structure contains distinct *entities* with their particular structure. Finally, the hierarchical teams permit having teams of teams.

## 4.3 Dynamic Model

To provide C2 Agility, we also present a computational model capable of representing the scenario members' behaviors based on the roles that such elements have. To cope with context changes, these members may reconfigure themselves or adjust their coordination according to the communication structure and information exchange of a C2 strategy.

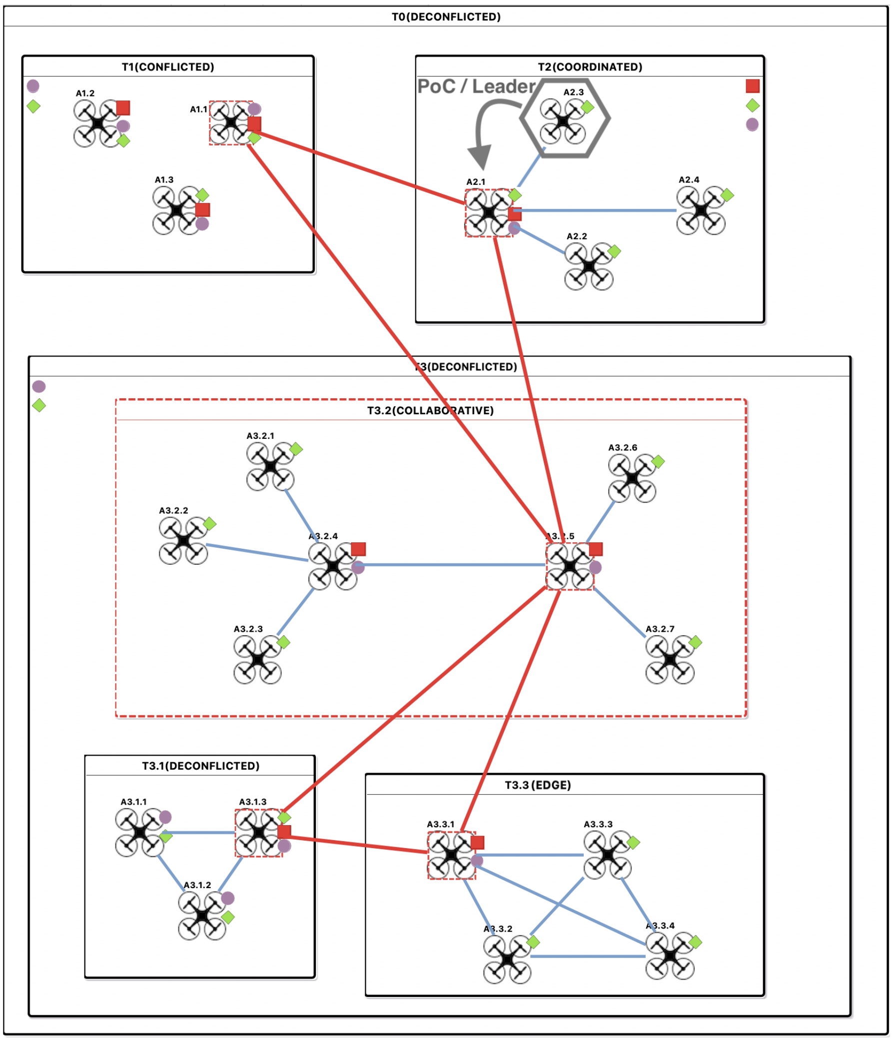



**Figure 4.6:** Instance generated after a context change due to an issue with the agent A2.3 forcing to change PoC and leader functions to the agent A2.1

This capacity is enabled by a model that allows coordination and data exchange via so-called channels, and a member's description composed of features that can be dynamically enabled according to the circumstances, generating different configurations.

Particularly, the proposed computational model is a typed-parameterized extension of the Channel System (CS) presented in Section 2.3, henceforth referred to as C2 channel system (C2CS), which is described in next sections. As a suitable and popular model for describing communication protocols and parallel systems, including the usage as the base for a model checker's input language, it makes CS a proper base to represent the roles within our proposed model. Thus, we base our proposal on the CS concept because of its capability of parallel process modeling, that makes it a proper support to represent the roles in the C2 system.

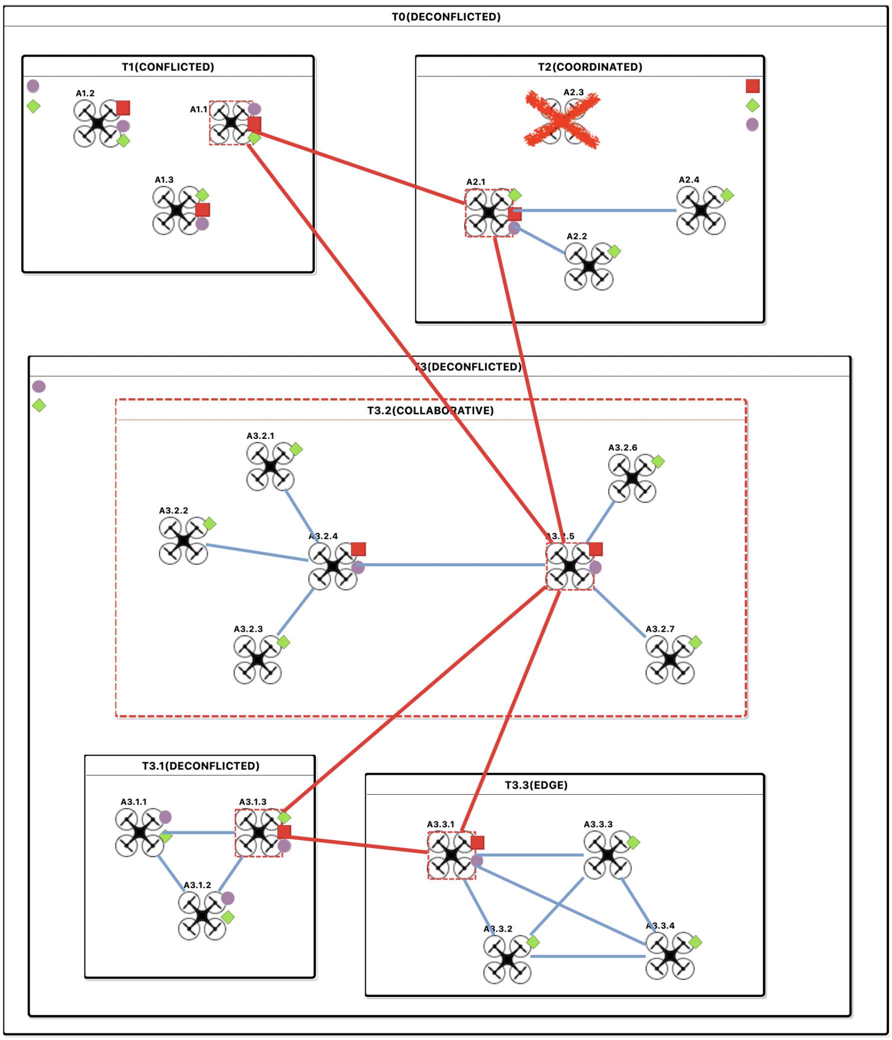



**Figure 4.7:** Instance generated after a context change due to a member loss with no structure changes required

### 4.3.1 Typed-Parameterized Channel System

Essentially, C2CS defines coordination among entities and their reconfiguration to cope with context changes eventually arising during a mission. These behaviors are described by three different program graphs (PGs) making up C2CS and representing role types played by entities listed in Table 4.1. TP establishes the tasks' execution and entity's reconfiguration behavior. TA defines task allocation responsibilities among entities, and C2AM specifies a C2 Approach change protocol. Types are used to describe the structure of the PG's locations as well as of the message content exchanged in synchronous and asynchronous channels.

Figure 4.9 depicts C2CS, whereas Equation 4.1 defines it. Using channel system concepts allows the execution of parallel processes with communication capability among

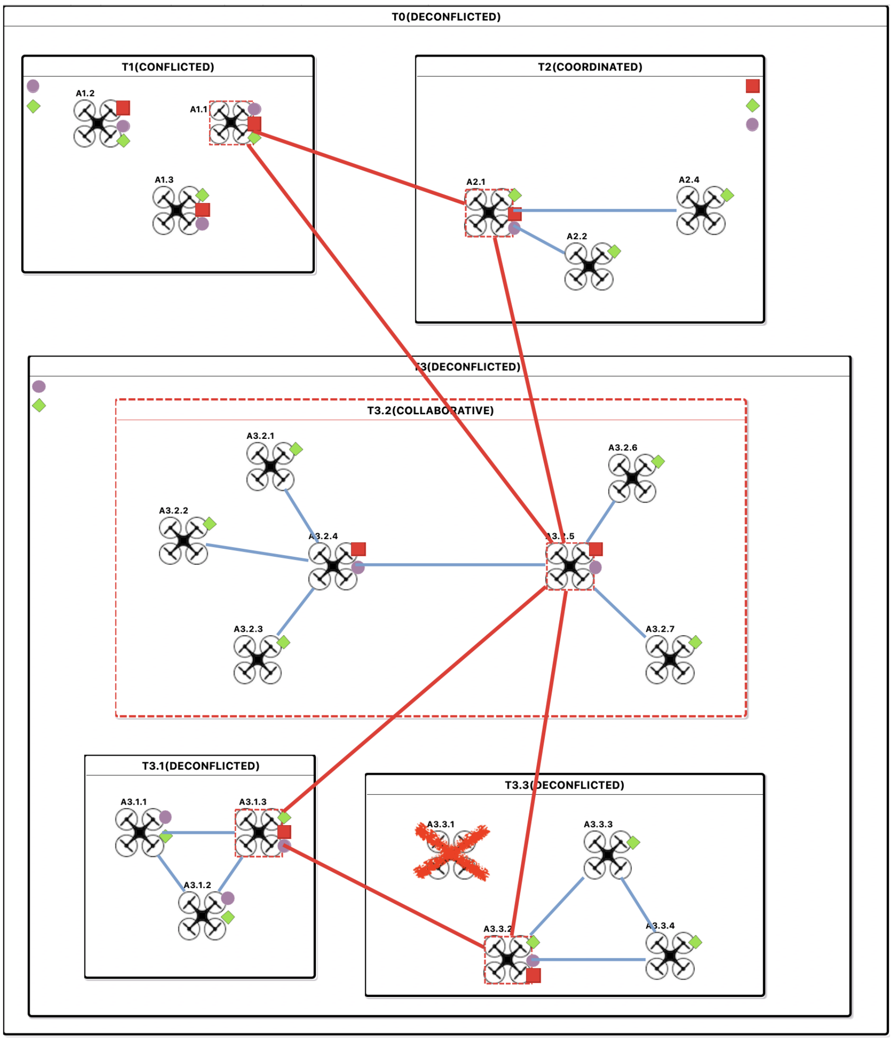



**Figure 4.8:** Instance generated after a context change due to a member loss with high impact on the C2 Approach and PoC/leader functions

them. Such a structure allows us to model the identified roles in a C2 system running in parallel and performing information exchange. This exchange leverages channels, i.e., FIFO buffers, that carry messages. Such a mechanism may use a Pub/Sub messaging [69] service extended with a decentralized message broker for abstracting CS channel structure. Thus, mainly used to model communication protocols [20] where data exchanging is defined, we use channels to model the roles' interaction. C2CS employs both synchronous and asynchronous channels. The former, e.g., $ch_2$ and $ch_3$, synchronizes C2AM's and TA's PG when they perform tasks allocation and C2 Approach change. The latter, e.g., $m_k$ and $ch_1$, enables the transmission of allocated tasks to the TPs and the return of not viable tasks to the TA.

**Table 4.1:** C2 System roles and responsibilities

| Role | Abbrev. | Responsibility |
|---|---|---|
| Task Performer | TP | Perform mission tasks |
| Task Allocator | TA | Allocate the tasks among mission performers |
| C2 Approach Manager | C2AM | Select an appropriate C2 Approach based on the resources available |

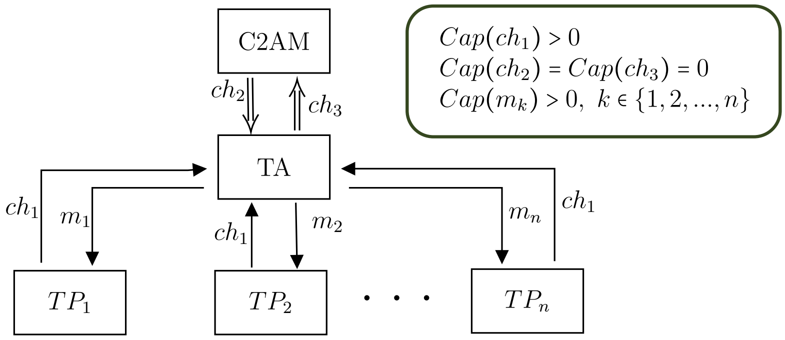


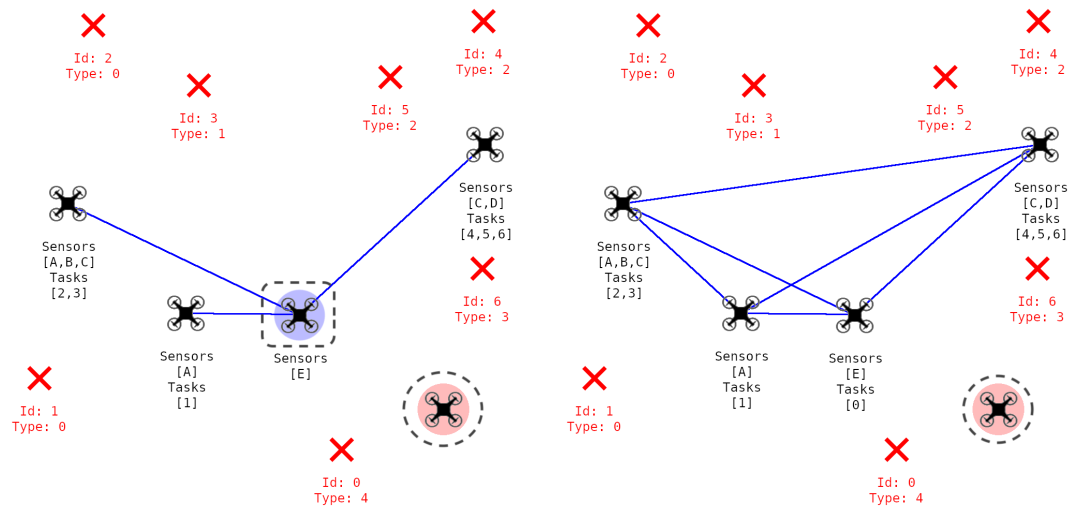

**Figure 4.9:** C2 Channel System for a team composed by $n$ members with the TP role asynchronous and synchronous channels represented by single and double line arrows, respectively.

$$C2CS([FM]\ E, \mathcal{P}(Task)\ M,\ C2_{ap}\ \omega_0) = [C2AM(E, M, \omega_0) \mid TA \mid \\ TP(fm_1, \omega_0) \mid ... \mid TP(fm_n, \omega_0)] \tag{4.1}$$

In Equation 4.1, the parameters have the following meaning and types. The set $M = \{t_1, t_2,\ ...,\ t_m\}$ is the designated mission, consisting of tasks that the members will try to perform. $M$ belongs to $\mathcal{P}(Task)$, which is the power set of $Task$, where $Task = \{t_a, t_b,\ ...,\ t_z\}$ is the set of all possible tasks. The list $E = [fm_1, fm_2,\ ...,\ fm_n]$ of team members is such that each team member has a capability of type Feature Model ($FM$). Additionally, such members always operate a C2 Approach $\omega \in \Omega$, where $\Omega$ = *{Edge, Collaborative, Coordinated, De-Conflicted, Conflicted}*. Lastly, the vertical bars represent parallel composition of the PGs representing the roles played by the team members.

C2CS leverages parameters to abstract from specific mission, entities, and initial conditions. It models roles that will be performed by the agents.

Each role played by the entities is represented by a PG that receives parameters that define some aspects of its internal structure. Such PGs are parallel processes that compose the C2CS and are instantiated according to the C2 Approach selected. Instances of different PGs can coexist in the same member according to the C2 Approach operated. In particular, the representation of multiple TPs shown in Figure 4.9 is to emphasize the existence of a specific asynchronous channel $m_k$ for each member $k$ owning a TP role instance, from where they receive the tasks to be performed. All members within $E$ with such a role instance are responsible for the mission's tasks accomplishment. Based on this,




**(a)** UAV dropped (dashed circle) **(b)** C2 Approach change

**Figure 4.10:** The team performing tasks after C2 Approach change (lines indicating communication links) due to problems with one of the members (dropped UAV marked with a dashed circle)

the member-to-role-instance mapping is such that each team member may eventually play one instance of the TP, TA, and C2AM roles simultaneously, and more details related to the implementation of this mapping are available in a public repository[1].

In a typical C2CS scenario, when the mission starts, TA allocates tasks to team members, which work on accomplishing them. Eventually, to handle context changes, members may reconfigure themselves, e.g., due to sensor failure, or the TA can reallocate tasks among members, e.g., due to member failure. Alternatively, even a task allocation might not suffice, in which case the C2AM might change the C2 approach, prompting new task allocation among members. In the worst-case scenario, when changing the C2 approach does not suffice, the task fails.

Recalling the motivating example (Section 3.3), in that specific C2 approach the coordinator plays the C2AM and TA roles. Each one of the other members plays the TP role. In that particular case, the loss of one member will first prompt the leader to reallocate tasks to the other team members, using the channels depicted in Figure 4.10a. Since the only member capable of executing the task that was previously allocated to the lost member is the leader, the C2AM changes the current C2 approach to Edge [6], prompting this modification using the communication network among members, i.e., the C2 Approach. This way, the former leader now plays the TP role and thus can execute the aforementioned tasks. This resulting situation is illustrated in Figure 4.10b.

[1]https://github.com/junieramorim/C2Agility

Overall, reconfiguration, performing new task allocation, and changing the C2 approach under certain context changes are explicitly represented in the PGs, thereby enabling the C2CS's strategy to achieve agility (cf. Section 3.3). The PGs are detailed in the following sections, and their implementation is publicly available elsewhere.[2]

### 4.3.2 Task Performer

The Task Performer's PG defines task execution and reconfiguration behavior for members playing this role. To enable this behavior, each member $e \in E$ is modeled as a Dynamic Software Product Line (DSPL) [63], whose Feature Model ($FM$) is given by a set of *features* $F = \{f_1, ..., f_k\}$ from which a valid set of configurations $[\![FM]\!]$ is obtained, i.e., $[\![FM]\!] : \mathcal{P}(F)$ [76, 107]. The $features$ represent the members' onboard resources, which when enabled indicate that the corresponding sensors are operational.

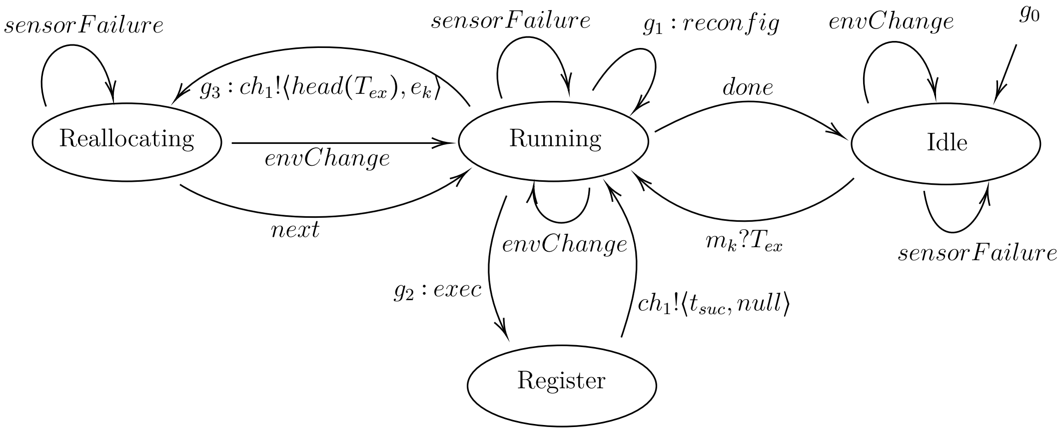
To work as a self-adaptive system, a set of actions composes the PG, e.g., *sensorFailure* and *envChange*, that perceives what is happening with the sensors and the environment, respectively. All perturbations from the system or outside of it call the action *reconfig*, that tries to perform a self-configuration. If it is not possible, the task returns to the *TaskAllocator* in order to be reallocated.

The member's configurations described by its feature model provide task completion capability. The member's reconfiguration behavior described in the PG is a response to deal with context changes and it is characterized by choosing a configuration compatible with the tasks. This aspect allows the members to reconfigure themselves to become compatible with the tasks to be performed.

Based on this, a member $e$ is capable of executing a task $t$ when it reconfigures itself to adopt a configuration $c \in [\![FM]\!]$ such that this configuration makes the member capable of dealing with the task $t$. This compatibility is known at the start and is denoted by *compatible*$(c, t)$. This configuration is characterized by sensors activated. There is a score, between 0 and 1, which indicates the compatibility level between each sensor $i$ onboard and the type of a task $j$, written as $Q_{ij}$ and so named quality. In summary, a member can receive a task $j$ when it has a sensor $i$ enabled whose $Q_{ij}$ meets the threshold, i.e., the acceptance level defined.

Figure 4.11 defines TP's behavior formally, including the effect of the actions $\alpha$ on the variables evaluation $\eta$. Accordingly, upon mission start (guard $g_0$), TP has an initial configuration $c_0$, i.e., initial state defined by member's characteristics or even by the domain requirements, and an initial C2 strategy $w_0$ received through channel $ch_2$ from $C2AM$ (location *Idle*). An example of initial configuration is an energy safe mode, i.e., all sensors off, operated to increase members' autonomy up to the target.

[2]https://github.com/junieramorim/C2Agility/tree/main/source




$g_0 := (\omega = \omega_0) \wedge (c = c_0 \in [\![e_k]\!])$

$g_1 := (T_{ex} \neq []) \wedge (\exists c' \in [\![e_k]\!]) \mid (c' \neq c) \wedge (\,!\, compatible(c,\ head(T_{ex}))) \wedge (compatible(c', head(T_{ex})))$

$g_2 := (T_{ex} \neq []) \wedge (compatible(c,\ head(T_{ex}))))$

$g_3 := (\nexists c \in [\![e_k]\!] \mid (compatible(c,\ head(T_{ex}))))$

$Var = \{e_k : FM;\ T_{ex} : [Task];\ \omega : C2_{ap};\ c : [\![FM]\!];\ t_{suc} : Task;$
$find\_configuration :\ Task \to [\![FM]\!];\ f\_env : FM \to FM\}$
$Effect(reconfig,\ \eta) =\ \eta[c := find\_configuration\ (head(T_{ex}))]$
$Effect(exec,\ \eta) :\ \eta[t_{suc} := head(T_{ex});\ T_{ex} := tail(T_{ex})]$
$Effect(done,\ \eta) :\ \eta$
$Effect(sensorFailure,\ \eta) : \eta[\ e_k\ :=\ e_k - f_{sensor}]$
$Effect(envChange,\ \eta) : \eta[\ e_k\ :=\ f\_env(e_k)]$
$Effect(next,\ \eta) = \eta[T_{ex} := tail(T_{ex})]$

**Figure 4.11:** Program Graph defining the Task Performer $k$ (TP) role

After the initial guard condition $g_0$ is satisfied and the first location is reached, as shown in Figure 4.11, TP can eventually be allocated tasks $T_{ex}$ that arrive over its dedicated asynchronous channel $m_k$ with the TA, at which point it will start addressing these in sequence (location *Running*). If TP is able to execute the first allocated task (guard $g_2$), it indicates successful task execution with the *exec* action, moving to location *Register*, and reports such task to the TA over the shared asynchronous channel $ch_1$, then moving back to location *Running.* If the member still has a configuration capable of addressing the task (guard $g_1$), it will reconfigure to it. Otherwise (guard $g_3$), it will notify this problematic task to the TA (location *Reallocating*) and continue execution (location *Running*).

Alternatively, the member may non-deterministically experience sensor failure (action *sensorFailure*), whose effect is described by removing configurations with the corresponding feature from the member's feature model, i.e., a self-reconfiguration that can be improved by a task reallocation. This action together with members loss (action *memberFailure*) composes changes in the self. In such a case, the tasks can be reallocated or a new C2 Approach can be operated to improve the system's awareness about the situation.

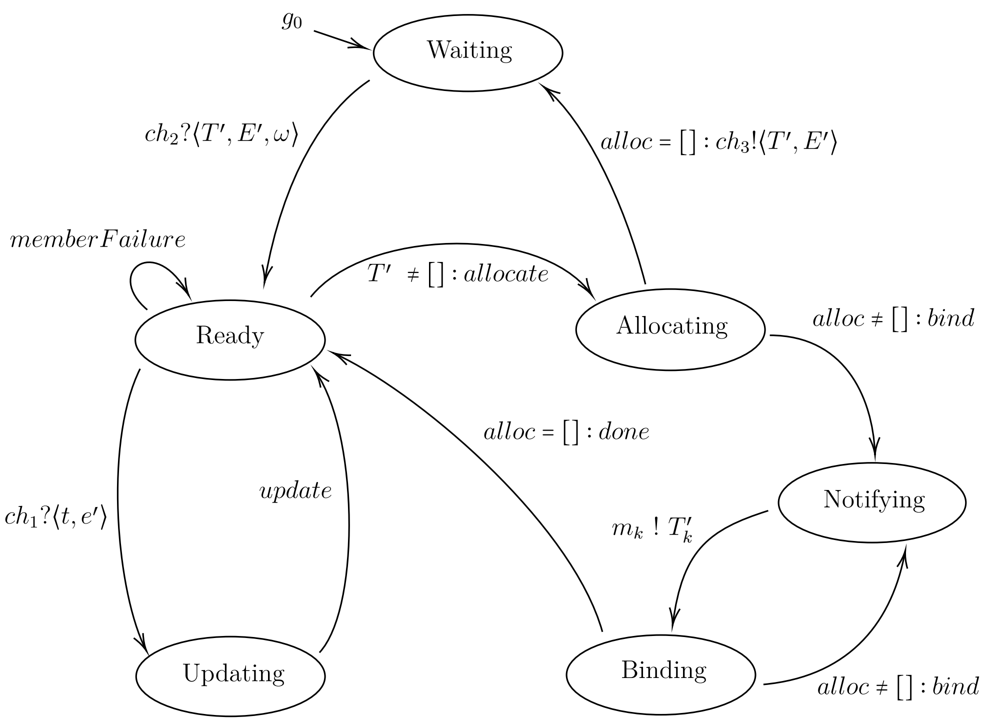
### 4.3.3 Task Allocator

In contrast to role TP, which focuses on task execution and member reconfiguration, TA defines coordination behavior among members, establishing task allocation responsibilities, as shown formally in Figure 4.12. When the mission starts, TA is in the *Waiting* location, staying there until it receives mission tasks $T'$ and members information $E'$ from the C2AM over the synchronous channel $ch_2$, then moving to location *Ready*. TA then performs task allocation (action *allocate*), notifies each member $k$ (locations *Notifying* and *Binding*) of its assigned task $T'_k$ via an specific asynchronous channel $m_k$, returning to the *Ready* location. Eventually, TA will be notified, over a shared asynchronous channel $ch_1$, of successfully completed tasks by the members and of failed tasks, in which case it will perform a new task allocation. In case of members' failure (action $memberFailure$), the TA tries to reallocate tasks previously allocated to those members. If unsuccessful, the TA reports such tasks along with members' status to the C2AM over the synchronous channel $ch_3$.

### 4.3.4 C2 Approach Manager

At the coarsest-grained level of coordination, C2AM specifies a C2 approach change protocol, as defined formally in Figure 4.13. C2AM starts operation by receiving mission tasks, member information, and initial C2 approach from the C2CS call ($g_0$ at location *Notifying*), i.e., mission startup. From this initial location, it goes to the location *Operating* with the initial C2 approach $\omega_0$ and sends the set of tasks $T$, the team $E$ and the $\omega_0$ over the synchronous channel $ch_2$. The C2AM remains in the *Operating* location until eventually receiving, from TA, the pair formed by the set of unallocated tasks and the updated team members. Such information comes over the synchronous channel $ch_3$ and takes C2AM to location *Maneuvering*.

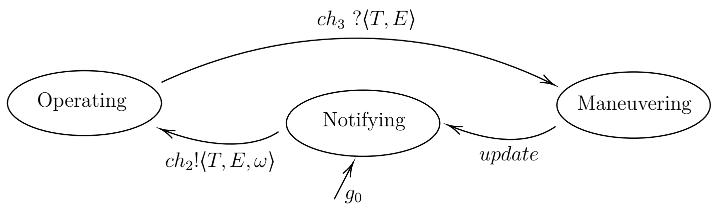
The tasks received and the last information about the members' status are analyzed with the action *update*. In case the function *find_maneuver* finds a C2 Approach to be applied in order to perform the tasks with the available team, it is defined and C2AM sends mission tasks, members' information and C2 Approach selected to the TA over the synchronous channel $ch_2$, then moving to the location *Operating*. Otherwise, those tasks are registered as failed ($T_{fail}$) and the TA is notified with an empty set of tasks. The C2AM then goes to location *Operating* passing through *Notifying* and adopts a domain defined C2 Approach $w_f$ (cf. Section 2.1).




$$g_0 := (T' = []) \ \wedge (f\_alloc \ = \ f_x([t_1,\ ...,t_n]\ p_1,\ [e_1,...,e_n]\ p_2,\ C2_{ap}\ p_3)) \\ \wedge (T_{success} = []) \ \wedge \ (\omega = \omega_0)$$

$$\begin{aligned}
&Var = \{\ alloc : [\langle\ \mathbb{N}, [Task]\ \rangle];\ f_alloc : ([Task], [FM], C2_{ap}) \ \rightarrow [\langle\ \mathbb{N}, [Task]\ \rangle]; \\
&\qquad teamStatus : [FM] \ \rightarrow \ \langle [Task], [FM]\rangle;\ T' : [Task];\ T_{success} : [Task]; \\
&\qquad E' : [FM];\ t : Task;\ e' : FM;\ \omega : C2_{ap};\ T_{add} : [Task]\ \} \\
&Effect(allocate,\ \eta) = \ \eta[alloc := \ f_alloc(T',\ E',\ \omega)] \\
&Effect(bind,\ \eta) = \ \eta[\ \langle k, T'_k\rangle := head(alloc);\ alloc := tail(alloc)\ ] \\
&Effect(update,\ \eta) = \begin{cases} \eta[\ T_{success} := T_{success} \ + + [t]\ ] & if\ (e' = null) \\ \eta[\ T' := T' + + [t]\ ;\ E' := E'(e/e')] & otherwise \end{cases} \\
&Effect(done, \eta) = \eta[\ T' := []\ ] \\
&Effect(memberFailure, \eta) = \eta[\ \langle T', E'\rangle := teamStatus(E')\ ]
\end{aligned}$$

**Figure 4.12:** Program Graph defining the Task Allocator (TA) role with the effect of the actions $\alpha$ on the variables evaluation $\eta$




$$g_0 = (\omega = \omega_0) \wedge (find\_maneuver = f_1(C2_{ap}, [e_1, ..., e_n], [t_1, ..., t_n]) \wedge (E = \{e_1, e_2, ..., e_n\}) \wedge (T_{fail} = []) \wedge (T = [t_1, t_2, ..., t_f])$$

$$Var = \{find\_maneuver : (C2_{ap},\ [FM],\ [Task]) \to C2_{ap};\ T_{fail} : [Task];\ \omega : C2_{ap};\ T : [Task];\ E : [FM];\ \omega_f : C2_{ap}\ \}$$

$$Effect(update,\ \eta) = \begin{cases} \eta[T_{fail} \coloneqq T_{fail} + +T;\ T \coloneqq [];\ \omega \coloneqq \omega_f\ ] & if\ (find\_maneuver(\omega, E, T) = NULL) \\ \eta[\ \omega \coloneqq\ find\_maneuver(\omega,\ E,\ T)] & otherwise \end{cases}$$

**Figure 4.13:** Program Graph defining the C2 Approach Manager (C2AM) role with the effect of the actions $\alpha$ on the variables evaluation $\eta$

# Chapter 5

# Assessment

This chapter presents the assessment of the models proposed in Sections 4.2 and 4.3. First, we use the static model as a resource of entities' information and properties during the implementation. In addition, we generate instances of realistic scenarios to validate the model compatibility and usability through a survey applied to military domain experts.

Second, we submit the dynamic model under different scenarios to measure the C2 system agility obtained. We perform this assessment through an *in silico* experiment [122] simulating UAVs dispatched on a reconnaissance mission to take images from a terrain. Besides, the results obtained are submitted, as well as the static model, to domain experts to validate the entities' behavior suitability.

## 5.1 Simulation-Based Assessment

This section presents the assessment of the dynamic model proposed in Section 4.3.1 under different scenarios to measure the obtained C2 System agility. We perform this assessment through an *in silico* experiment [122] simulating UAVs dispatched on a reconnaissance mission to take images from a terrain. Such a method provides a way to analyze different situations, and it allows us to work with different scenarios that would otherwise be unfeasible to test given incurred cost and resource availability. The simulated environment considers that the circumstances can change during the mission execution, allowing to test the effectiveness of the C2 System under such conditions. The replication package comprising the artifacts related to this empirical evaluation is available in a public repository [1].

The remain of this section presents the goal definition, and the metrics applied in the evaluation (Section 5.1.1), the study's hypotheses description (Section 5.1.2), the details and elements of the simulation scenarios (Section 5.1.3), the experimental design and

[1]https://github.com/junieramorim/C2Agility/wiki

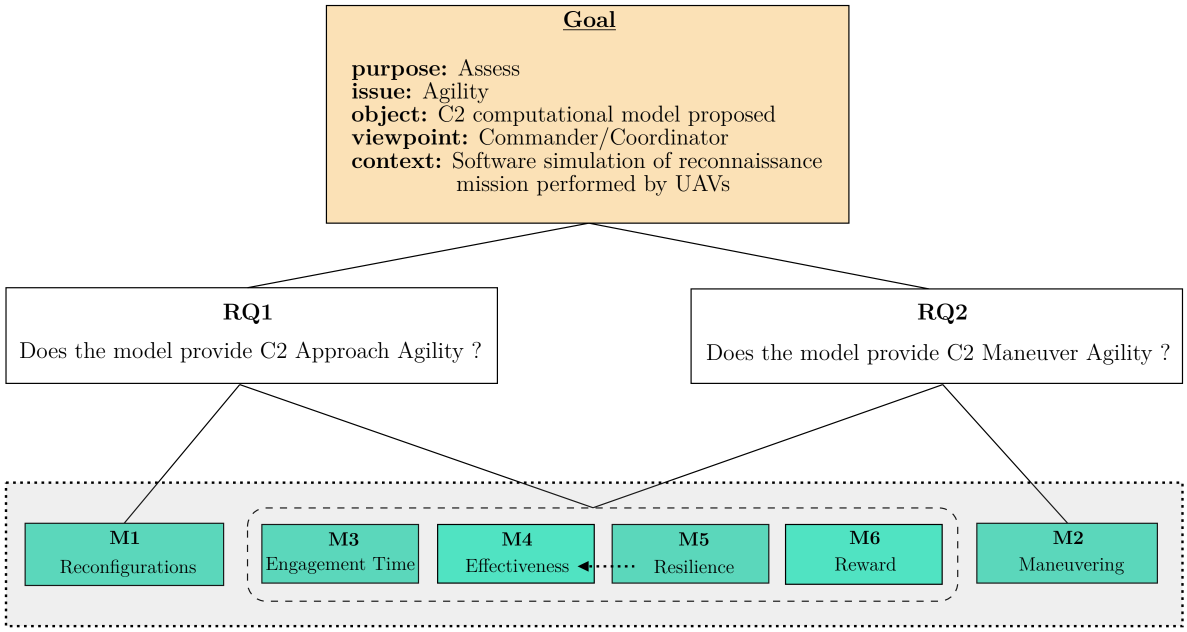



**Figure 5.1:** GQM Diagram with the Goal, the Research Questions RQ1 and RQ2, and the metrics (M1, M2, M3, M4, M5 and M6)

analysis procedure (Section 5.1.4), the simulation's implementation details and environment (Section 5.1.4.1), the experiments executions (Section 5.1.5), and last, the analysis of the obtained results and discussion about the findings (Section 5.1.6).

### 5.1.1 Definition

The Goal-Question-Metric (GQM) [125] defines the empirical study performed. Figure 5.1 shows the goal as the assessment of C2 Agility in our proposed computational model. The assessment made is useful to the entities' commanders. Such entities may have internal entities, recursively, up to the level of individual members who also have commanders. From the goal, we derived two research questions, which consider the members within the same C2 Approach (*Q1*) and C2 Approach changes (*Q2*). Such questions address the level of C2 Agility existing in the system. Table 5.1 defines the metrics applied in the evaluation to answer the proposed questions and corresponding agility enablers. The metrics are (*M1 - Reconfigurations; M2 - Maneuvering; M3 - Engagement Time; M4 - Effectiveness; M5 - Resilience; M6 - Reward*) related to the agility enablers described by Gren and Lenberg [62], Tran et al. [121] and Alberts [7].

Context perturbations, e.g., sudden changes in the environment, or onboard component damage, require system adaptation to keep running. To identify the system's adaptation ability, we count the number of member reconfigurations (M1) and the C2 approach changes (M2). The Maneuvering (M2) metric assesses the adaptability level

**Table 5.1:** Metrics used to evaluate the proposal

| ID | Metric | Description |
|---|---|---|
| M1 | Reconfigurations | Number of internal reconfigurations performed by the members to accomplish the mission within a given timeout. |
| M2 | Maneuvering | Number of C2 Approach changes performed by the members to accomplish the mission within a given timeout. |
| M3 | Engagement Time | System time, in ticks, during which the entities were engaged in the execution of the tasks. |
| M4 | Effectiveness | Percentage of successful tasks completed by the task performers. |
| M5 | Resilience | System's capacity to obtain the same effectiveness of scenarios without context changes when dealing with new contexts. |
| M6 | Reward | Total quality of all sensors used to perform the tasks |

of the system because of its capability of changing its organization. Response time is measured by the Engagement Time (M3) metric, which measures how long the system takes to execute the mission's tasks. Such an aspect highlights the responsiveness of the system to changing circumstances. The ratio of the mission's tasks that the members have already accomplished is evaluated by the Effectiveness (M4) metric.

The difference between the effectiveness (M4) metric obtained from scenarios with and without context changes shows the system capacity to recover the highest effectiveness level obtained from the scenario without context changes. Such recovery capacity defines the Resilience (M5) metric. Finally, Reward (M6) defines the compatibility between the members and the mission's tasks. Equation 5.1 defines this metric in terms of the compatibility score $Q_{ij}$ sum.

$$M6 = \sum Q_{ij} \ , where\ t_j \in T_{success}\ and\ compatible(c_i, t_j) \tag{5.1}$$

The $Q_{ij}$ defines the compatibility level of a task $j$, which has a type, with a sensor $i$ used to perform it. Such a sensor corresponds to a feature in the configuration $c_i$ run by the member, and all tasks successfully performed are stored in $T_{success}$ for results analysis. Besides, all tasks with the same type have the same compatibility level with the sensors.

### 5.1.2 Hypotheses Formulation

Figure 5.2 shows the factors and the dependent variables related to the GQM (Figure 5.1). The scenario and the action method are identified as factors. A scenario comprises an initial context and a sequence of events that occur during the mission. Section 5.1.3 details potential scenarios.

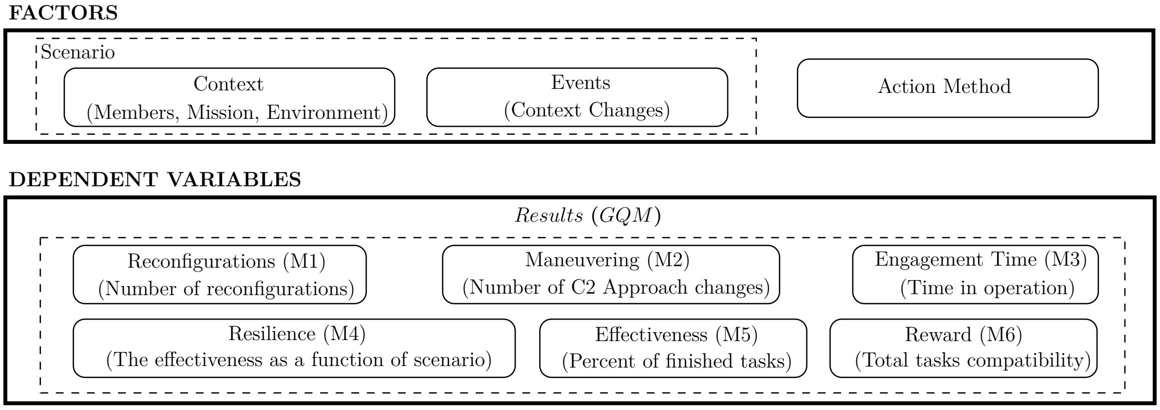



**Figure 5.2:** Factors and dependent variables used by the proposed model

The second factor is called the action method, and it represents the system's response to deal with a context change or perturbation. Consider two feasible treatments, namely A1 and A2, which identify the baseline and the proposed model to respond to context changes, respectively. With treatment A1, the system starts the execution with an initial context, performs a task allocation of mission tasks among team members, and in face of context changes, the system just keeps running as long as it can, but it does not perform any kind of adaptation to deal with new circumstances. Such a treatment represents the low maturity of the state-of-the-art in C2 agility to deal with context changes. Indeed, especially in the military domain, e.g., Alberts et al. [6], Leal et al. [83], the proposed solutions perform only a historical analysis of success or an aleatory strategy choosing to deal with the new circumstances based on a fast and simplified analysis of some scenario variables. In contrast, treatment A2 applies the C2 computational model proposed in Section 4.3.1.

Also, in all considered scenarios of both treatments, each task can be executed by at least one team member. It is relevant because of the baseline, where there are no members' reconfiguration.

Considering the research questions and the metrics identified in GQM, we define the corresponding null and alternative hypotheses as shown in Table 5.2.

The null hypothesis states there will be no statistically significant differences in metrics' average for different action methods in the same scenario. The alternative hypothesis states otherwise.

### 5.1.3 Simulation Scenarios

A simulation scenario consists of an initial context and a sequence of events that provide dynamism (cf. Figure 5.3). A regular expression was used to write the sequence of events representing the possibility of multiple occurrences of each event $\alpha_i$, which can be

**Table 5.2:** Experimental hypotheses, where $\overline{m}$ is the average of each metric $m$ in $Metric = \{M_1, M_2, M_3, M_4, M_5, M_6\}$ and *Scenario* is the set of scenarios composed by context and events (see Figure 5.2)

| **Hypothesis** | **Definition** |
|---|---|
| $H_0$ | $\forall m \in Metric, \forall s \in Scenario \cdot (\overline{m}(s \times A1) = \overline{m}(s \times A2))$ |
| $H_1$ | $\forall m \in Metric, \forall s \in Scenario \cdot (\overline{m}(s \times A1) \neq \overline{m}(s \times A2))$ |

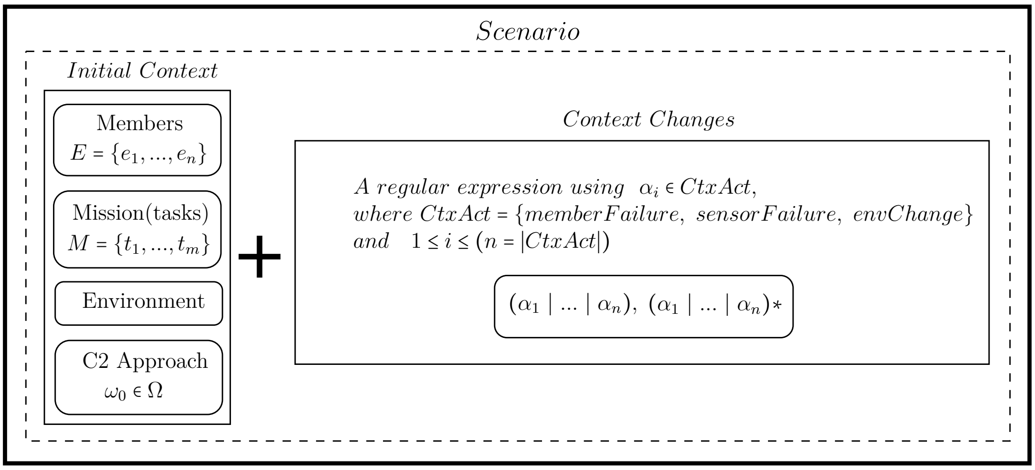



**Figure 5.3:** Scenario comprising the initial context variables and a sequence of events that characterizes a dynamic context

none or several, making up the sequence of context changes. The initial context comprises the set $E$ of members operating an initial C2 Approach $\omega$, the mission $M$ composed of a set of tasks, and the environment. Each event in the set $CtxAct$ (see Equation 5.2), so-called Context Actions, represents an action that causes a member or environment changes at runtime, i.e., changes in the context.

$$CtxAct = \{memberFailure, sensorFailure, envChange\} \tag{5.2}$$

The environment represents all the conditions of the place where the members act, e.g., weather conditions, hazard due to enemy's activity, and communication restrictions. They are modeled as the state of a specific kind of onboard sensor, i.e., a foggy day can turn a VGA sensor useless.

Table 5.3 lists the possible actions combined with the initial context used by the simulation. These actions occur during the simulation to create a dynamic scenario, resembling realistic settings. Environmental changes are perceived by onboard sensors in all members and cause a reduction of their capability to be employed in a mission.

**Table 5.3:** Context changes handled by the simulator with related actions in the CS and their impact level in the system

| Change | Action | Impact | Description |
|---|---|---|---|
| Environment | *envChange* | low | Weather conditions (e.g., luminosity, cloudiness) impacting the sensor's quality; Hazard level requiring changes in the UAV's behavior |
| Self | *sensorFailure* | medium | Sensor onboard damage caused by any internal issue (e.g., electronic circuit damaged) |
| | *memberFailure* | high | UAV out of operation due to serious damage (e.g., no fuel or battery, or taken down) |

With military domain experts' collaboration, we classify these context change events according to their impact on the system. Such classification is based on consolidated doctrines of a situation analysis and planning in military operations with application in many contexts [5, 13, 29, 50, 55, 83]. It is used to create the sequences of events in an increasing level of complexity, requiring more resilience from the system.

Table 5.4 shows the scenarios considered in the simulation.

The related event sequences were selected to have an increasing level of complexity and commonality, based on the impact classification presented in Table 5.3, obtained from military domain experts. Therefore, environment changes are more common to occur in the operation, followed by sensor issues and member failures because of enemy engagement or mechanical problems.

Besides the sequences of events presented, the initial contexts for all scenarios are formed by a team $E$ with 5 members, a mission $M$ with 30 tasks, and De-Conflicted as the initial C2 Approach $w_0$, i.e., a ring communication structure. During the simulation, these events occur in the sequence presented but at random times within the mission timeout.

The simulation operates a scenario with 5 possible types of tasks (0 to 4) and 5 types of sensors (A, B, C, D, and E). The tasks and onboard sensors are randomly chosen before the round executes. When the task allocation process starts, the algorithm applies a function that returns the quality $Q_{ij}$, obtained from a table, that correlates a sensor $i$ to the task $j$. When $Q_{ij} = 0$ it means the sensor $i$ is not able to perform the task $j$.

### 5.1.4 Experimental Design and Analysis Procedure

Figure 5.4 illustrates the factorial experimental design applied to assess the formulated hypotheses, where each action method is exercised with each scenario in a trial

**Table 5.4:** List of events (*EC-envChange; SF-sensorFailure; MF-memberFailure*) that characterizes the context changes within the scenarios tested. The initial C2 Approach, the set of members $E$ and the mission $M$ remain unchanged.

| Scenario | Context Changes |
|---|---|
| 1 | EC, EC, EC, EC, EC |
| 2 | EC, EC, EC, EC, EC, EC, EC, EC, EC, EC, EC, EC, EC, EC, EC |
| 3 | EC, EC, EC, EC, EC, MF, EC, EC, EC, EC, EC, MF, EC, EC, EC, EC, EC, MF, EC, EC, EC, EC, EC |
| 4 | EC, EC, EC, EC, EC, SF, EC, EC, EC, EC, EC, SF, EC, EC, EC, EC, EC, SF, EC, EC, EC, EC, EC |
| 5 | EC, EC, EC, SF, EC, EC, EC, MF, EC, EC, EC, SF, EC, EC, EC, MF, EC, EC, EC, SF, EC, EC, EC, MF |
| 6 | EC, SF, MF, EC, SF, EC, SF, EC, SF, MF, EC, SF, EC, SF, EC, SF |
| 7 | MF, SF, EC, EC, EC, EC, EC, EC, EC, EC, EC, EC, MF, SF, EC, EC, EC, EC, EC, MF, SF, EC, EC, EC, MF, SF, EC |
| 8 | MF, SF, EC, MF, SF, EC, MF, SF, EC, MF, SF, EC |
| 9 | SF, SF, SF, MF, SF, SF, SF, MF, SF, SF, SF, MF, SF, SF, SF |
| 10 | SF, EC, MF, SF, EC, MF, SF, EC, MF, SF, EC, MF, EC, EC, EC, EC, EC, EC, EC, EC, EC, EC |

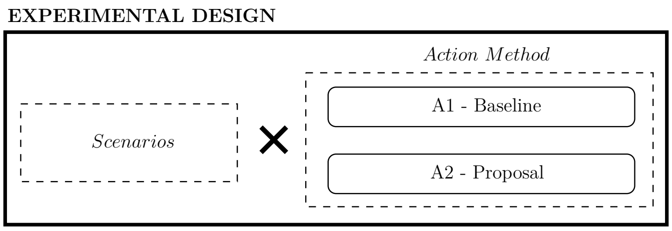



**Figure 5.4:** Factorial experimental design, where each action method is exercised with each scenario.

(cf. Section 5.1.3). Combinations of the factors are assessed with the metrics shown in Figure 5.1.

The simulation uses a set of random variables to define some elements during execution, such as UAVs' and tasks' positions, and sensors that the context event will affect. Simulator engine generates these random numbers based on seeds created in runtime. For a consistent comparison between the action methods A1 and A2, a fixed set of seeds within the same trial were used in order to have the same value of such variables during experimentation round.

To proceed with data analysis and sample size definition, a confidence level of 95% in all statistical calculations was used. Such confidence level brings a significant population mean estimation and an acceptable sampling error A *p-value* of 0.05 was used to perform statistical hypothesis testing. The choice of these parameters is based on the standard cutoff used in previous works, e.g., Cochran W.G. [37] and Bruce [31].

#### 5.1.4.1 Instrumentation

Following related research on C2 simulation [6, 50, 116, 121] and on the use of roles as components of agents' implementation [45, 70], we performed simulations using an agent-based framework, in this case *Repast Simphony* [1, 97, 127]. This choice was based on the necessity of flexibility and well-formed structure. After the exploratory study results (cf. Section 3.1), that showed us a risk for using *Netlogo* because of its integration and documentation limit, and its monolithic code.

Using JAVA programming language, we defined the structure with the root *C2Drones*, as shown in Figure 5.5. It has two main packages, named *controller* and *entities*. The first one has the classes responsible for using all features provided by the *Repast* Suite and making the simulator to work as a multi-agent system. The second one concentrates the classes that represents the entities and its communication structure, i.e., UAVs and token mechanism, and it has a dependence with *domain* package, with the classes related to the C2 approach, and with *roles*, that implements the roles defined by the C2CS (cf. Section 4.3.1). Finally, the package *utils* concentrates classes that provide common functionalities used by all the simulator, i.e., general converter, parameters, log and report generators.

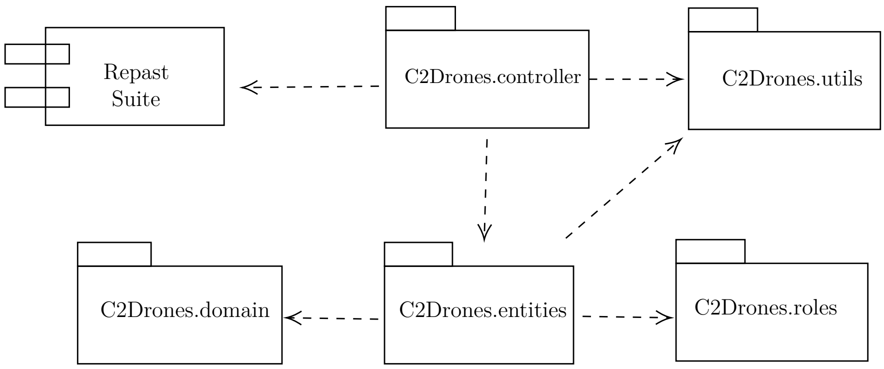



**Figure 5.5:** Simulator structure with the packages and component interaction.

The simulation implements the three roles modeled by the PGs shown in Figures 4.11, 4.12, and 4.13 to define UAV behavior with autonomy of reconfiguration, tasks allocation and execution, employed in a military recognition mission [29]. Figure 5.6 shows the UAV implementation through the class *Drone* and its roles defined by the abstract class *Role* and its heritages. The UAVs have a list of onboard sensors and the methods to send and to receive communication tokens, i.e., *sendToken()* and *receiveToken(token)*.

Each role played by the UAV has in its specialized class, all methods for moving to the task and for performing it, i.e., the methods *move(drone)* and *executeCurrentTask(drone)* within the class *Executor*. In addition, the class *TaskAllocator* implements the task allocator role with its method to allocate existing tasks, i.e., method *allocateBestPair(tasks,*

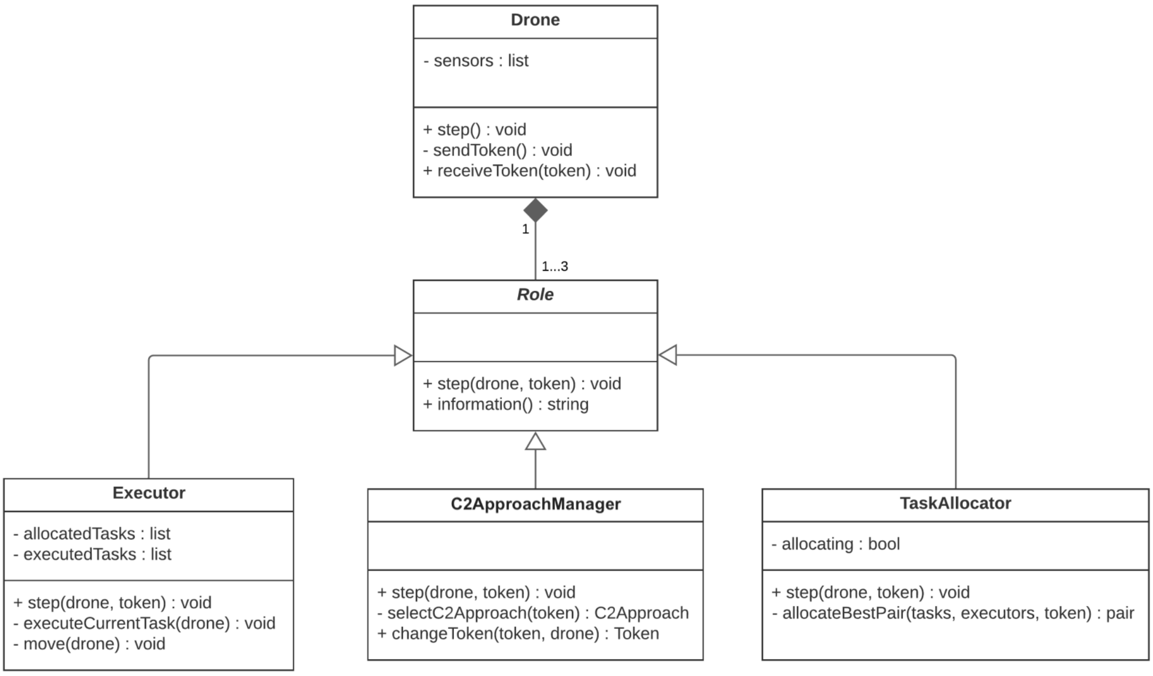



**Figure 5.6:** Class diagram representing an entity and its roles

*executors, token)*. Finally, class *C2ApproachManager* implements the methods to select a suitable C2 approach, i.e., *selectC2Approach(token)*. In such a case, it may use the method *changeToken* to adjust the communication structure according to the C2 approach selected. As the selected communication protocol is token based, the *token* object is used as a parameter by many methods because of its function of data exchanging to provide system awareness.

We implement the behavior for each role within the method *step()*. In the class *Executor*, this method implements the MAPE loop that characterizes the entities' self-configuration, i.e., the first MDSPL layer (cf. Section 4.1). Besides, this same method in the class *C2ApproachManager* implements the second MDSPL layer responsible for coordinating. Its implementation in the class *TaskAllocator* receives the allocation requirement coming from the C2AM role.

Inspired by the solution applied in Schwarzrock et al. [108] and Amorim et al. [13] to allocate tasks to a set of agents, we used a similar principle in this work based on swarm strategy. Depending on the C2 Approach operating, the members are notified about the allocation performed and unfinished tasks. This guarantees a better awareness level among entities and can improve the allocation result because of more context information shared by the members. Such allocation is feasible when a member has a configuration that activates a task-compatible sensor. Implementing all required procedures to perform this allocation is within the class *TaskAllocator*.

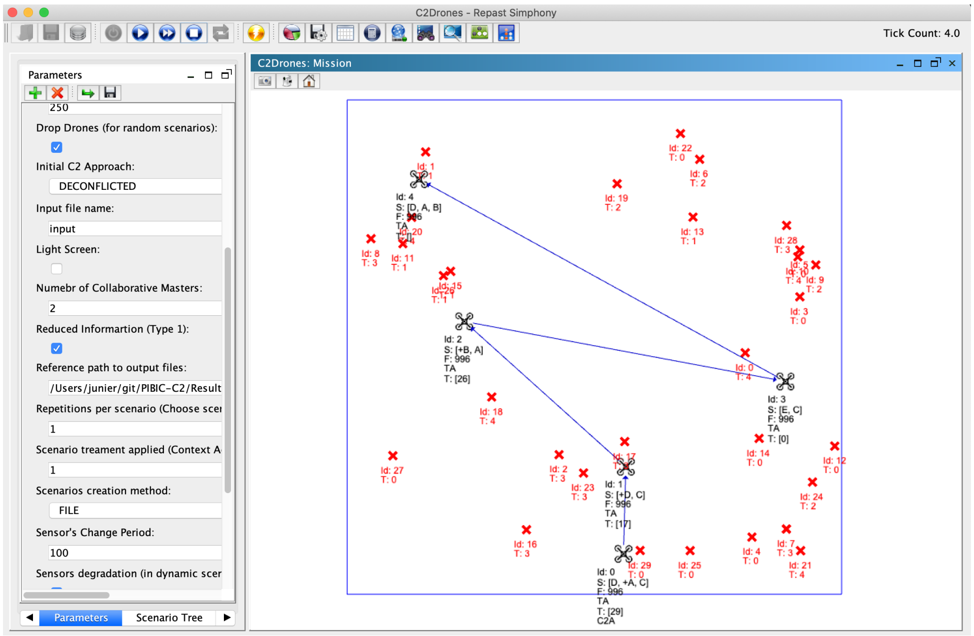



**Figure 5.7:** Simulator implementation with Repast Simphony

Furthermore, perturbations and factors modifications were inserted in runtime to provide dynamism to the system and make it closer to a real scenario. A context change simulated by the system reduces a specific type of sensor, e.g., an environment change simulating a luminosity decreasing reduces by 50% the quality of the sensor type 2 that represents a VGA camera. Sensor failure is modeled by setting its quality to zero. Thus, a member can be lost with all its onboard sensors.

The C2 Approach Maneuvering follows the enumerated type described in Section 2.1. From the initial C2 Approach, which is De-Conflicted (cf. Section 5.1.3), the maneuver follows incrementally over this list and going back to Conflicted after Edge. The Conflicted is the final C2 Approach adopted by the system before discarding the unfeasible tasks. Such a sequence is based on real scenarios of the military domain where a disconnected structure is adopted in extreme situations where there are no conditions to hold another communication and interaction strategy [5].

Precisely, the C2 elements within the simulator follow the behavior defined by our C2CS (cf. Section 4.3.1). Figures 5.7 and 5.8 show an execution screen of the implementation with parameters that can be changed to create different experimental setups and the panel with the agents' execution field, and its state diagram showing the *Executor* role behavior. We modeled all the roles following the steps defined in the C2CS. We played the simulations in a notebook with a 2.3 GHz Quad-Core Intel Core i7 processor, 16 MB DDR3 of memory, graphics by NVIDIA GeForce GT 750M, and OS X as the operating system.

The trials are performed feeding the simulator with the scenario data (cf. Section 5.1.3)

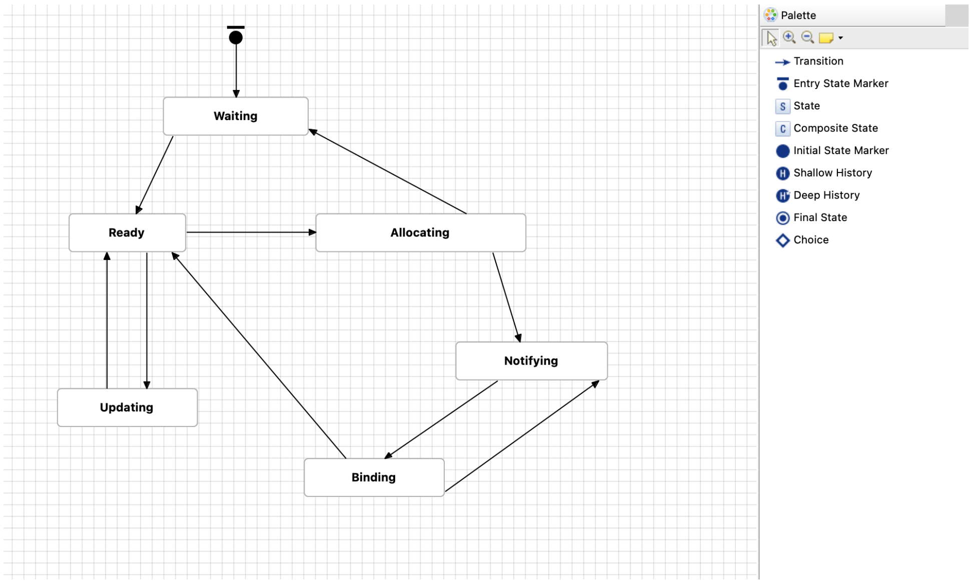



**Figure 5.8:** State diagram created in the Repast Simphony, corresponding to the role Executor in C2CS, to define the entities' behavior playing such a role.

combined with the two action methods shown in Figure 5.4. Such scenarios are loaded through an input text file (CSV pattern). With all these data, the simulator runs each scenario with each action method, according to the experimental design (cf. Section 5.1.4). The results are exported as text files and processed with R tool [77, 118] to calculate the mean and the standard deviation, draw graphs, and perform the statistical tests and validation.

### 5.1.5 Executions

After some initial single executions to validate the entities' behavior according to the C2CS, following the state diagram (see Figure 5.8), we perform a set of executions in a batch way to reduce the effect of system startup on the standard deviation. An initial number of runs ($n_i = 50$) was chosen to analyze the amount of dispersion in the results obtained by the experiments. Based on this, it was possible to observe the need to adjust the sample size to meet the acceptable confidence level (cf. Section 5.1.4). According to Cochran W.G. [37], Equation 5.3 gives the sample size $n$ that satisfies a given confidence level, a population standard deviation $\sigma$, and the difference $d$ between the population mean $(\overline{X})$ and sample mean $(\mu)$.

$$n = \left( \frac{Z_{\alpha/2} \cdot \sigma}{d} \right)^2 \tag{5.3}$$

According to the analysis procedure (cf. Section 5.1.4), the Equation 5.3 is applied with 95% confidence, i.e., a $Z_{\alpha/2} = 1.96$, and $d = |\overline{X} - \mu| \leq 0.1\mu_i$, where $\mu_i$ is the mean obtained with the initial set of runs $n_i$. This value of $d$ is less than 10% of the mean result obtained by the initial sample. With such parameters, an average sample size of 500 was obtained. Based on this, the factorial experiment with scenario and action method factors (see Figure 5.4) was executed 500 times for all combinations of treatments. Table 5.5 shows the results obtained for the metrics listed in GQM (Figure 5.1) for each scenario, which was performed 500 times.

**Table 5.5:** Metrics results (Reconfigurations - M1, Maneuvering - M2, Engagement Time - M3, Effectiveness - M4, Resilience - M5 and Reward - M6) after 500 executions of each scenario listed in Table 5.4 with the initial context of 5 members, 30 tasks, De-Conflicted as initial C2 Approach and a deadline of 1000 time ticks.

| | M1 | M2 | M3 | M4 | M5 | M6 |
|---|---|---|---|---|---|---|
| | Mean(St.Dev.) | Mean(St.Dev.) | Mean(St.Dev.) | Mean(St.Dev.) | Mean(St.Dev.) | Mean(St.Dev.) |
| ⟶ **Scenario 1** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 890.0 (±66.8) | 82.0 (±5.9) | 97.5 (±1.0) | 20.7 (±1.7) |
| **A2** | 17.5 (±1.7) | 1.7 (±0.3) | 945.5 (±33.1) | 97.7 (±2.9) | 99.9 (±0.1) | 26.5 (±1.2) |
| ⟶ **Scenario 2** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 866.5 (±72.3) | 77.5 (±6.3) | 92.2 (±1.7) | 20.0 (±1.7) |
| **A2** | 18.1 (±2.1) | 2.6 (±0.6) | 934.0 (±33.0) | 95.2 (±4.2) | 97.3 (±1.3) | 25.1 (±1.2) |
| ⟶ **Scenario 3** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 782.1 (±78.3) | 63.7 (±5.5) | 75.7 (±1.9) | 16.1 (±1.3) |
| **A2** | 15.4 (±1.7) | 2.8 (±0.3) | 883.0 (±49.2) | 69.1 (±4.5) | 70.7 (±2.3) | 17.2 (±1.5) |
| ⟶ **Scenario 4** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 799.1 (±77.0) | 63.6 (±5.3) | 75.6 (±1.6) | 16.4 (±1.3) |
| **A2** | 18.2 (±2.3) | 3.0 (±0.1) | 913.4 (±35.9) | 84.5 (±5.1) | 86.4 (±2.4) | 21.9 (±1.4) |
| ⟶ **Scenario 5** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 733.7 (±75.5) | 54.6 (±4.6) | 64.9 (±1.4) | 13.9 (±1.2) |
| **A2** | 16.1 (±2.0) | 2.9 (±0.2) | 887.7 (±48.4) | 70.7 (±5.1) | 72.3 (±2.9) | 17.9 (±1.5) |
| ⟶ **Scenario 6** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 700.6 (±79.4) | 42.0 (±4.5) | 49.9 (±2.2) | 10.7 (±1.1) |
| **A2** | 15.0 (±2.0) | 2.9 (±0.2) | 910.5 (±59.3) | 65.7 (±5.9) | 67.2 (±3.9) | 16.6 (±1.9) |
| ⟶ **Scenario 7** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 686.2 (±58.4) | 41.1 (±3.5) | 48.9 (±1.1) | 10.9 (±1.0) |
| **A2** | 15.0 (±2.0) | 2.8 (±0.3) | 820.5 (±61.4) | 60.1 (±5.6) | 61.5 (±3.7) | 16.2 (±1.6) |
| ⟶ **Scenario 8** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 629.7 (±68.6) | 36.3 (±3.8) | 43.2 (±1.7) | 9.1 (±0.9) |
| **A2** | 13.0 (±1.5) | 2.8 (±0.3) | 785.1 (±58.3) | 52.1 (±4.7) | 53.3 (±3.1) | 13.6 (±1.7) |
| ⟶ **Scenario 9** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 491.0 (±74.2) | 26.4 (±3.7) | 31.4 (±2.4) | 7.0 (±1.0) |
| **A2** | 15.5 (±1.6) | 3.0 (±0.1) | 743.2 (±66.1) | 53.2 (±5.1) | 54.4 (±3.4) | 14.2 (±1.5) |
| ⟶ **Scenario 10** | | | | | | |
| **A1** | 0.0 (±0.0) | 0.0 (±0.0) | 393.4 (±81.2) | 24.3 (±2.6) | 28.9 (±1.3) | 6.2 (±0.9) |
| **A2** | 11.4 (±1.4) | 2.9 (±0.2) | 650.7 (±114.8) | 38.8 (±4.7) | 39.7 (±3.4) | 9.9 (±1.5) |

### 5.1.6 Analysis and Discussion

Table 5.5 and Figures 5.9, 5.10, 5.11, 5.12, 5.13 and 5.14 show the results obtained by the simulation. Positive values for Reconfigurations (M1) (cf. Figure 5.9) occur only in the proposed method (A2) and shows team's adaptation to deal with context changes in all simulated scenarios. By tracing into the source code of the simulation, we observed that the tasks' reallocation activity follows such adaptations in case the reconfiguration does not make the member compatible with the task. A similar analysis can apply maneuvering (M2) shown in Figure 5.10. Positive values are obtained only in the A2 method, which can perform a C2 Approach change to try a new awareness level. The values obtained show an upper bound because of the sequence of C2 Approach adopted starting from De-Conflicted up to Edge and finishing on the Conflicted one (cf. Section 2.1).

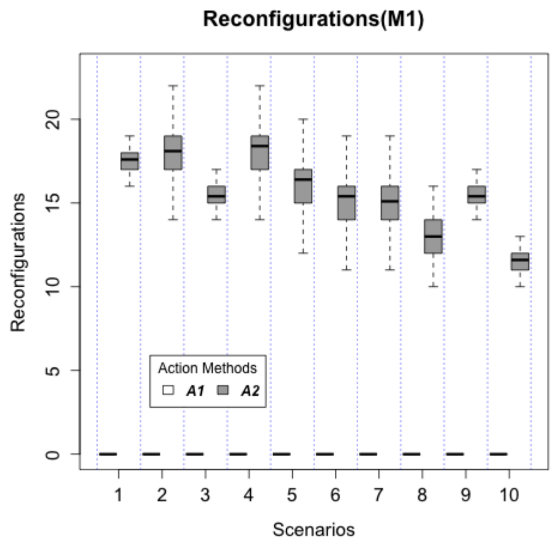



**Figure 5.9:** Reconfigurations (M1)

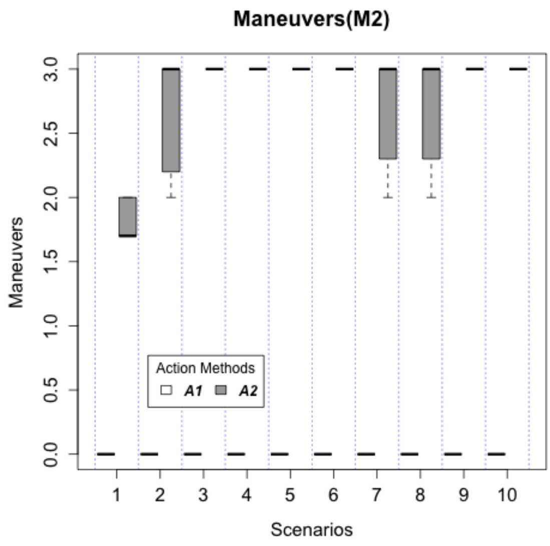



**Figure 5.10:** Maneuvering (M2)

Figure 5.11 shows results of Engagement Time (M3) for each scenario. In this case, the A2 method allows the system to keep running longer than A1, even in scenarios with more context changes. This suggests A2 provides more system availability due to its enhanced adaptation capability, as indicated by results for M1 and M2.

Regarding Effectiveness (M4), Figure 5.12 shows higher capacity of the C2 system to solve tasks by the A2 method compared to A1, suggesting the higher availability obtained for M3 is well-employed in dealing with the mission in A2, increasing the system capacity to deal with context changes and to perform the tasks.

Regarding the Resilience (M5) metric, Figure 5.13 shows that, in general, action method A2 is more capable of presenting a percentage of completed tasks closer to what would be obtained if the context remained unchanged. The only exception is for Scenario 3, since, upon exploration of simulation execution steps, the *memberFailure* events in this scenario prompt task reallocation that cannot be better than the initial one made with the complete set of members. Besides, the time spent on members' reconfiguration and tasks' reallocation leads to a slightly lower result than that obtained by the A1 method, which only continues the execution.

In terms of Reward (M6), Figure 5.14 shows better results for A2 indicating that the system adaptation provides more compatible configurations for performing tasks. Indeed, tracing into the simulation execution of method A2 reveals that all reconfiguration and C2 Approach change performed may lead to task reallocation. Such behavior always looks for the best member and sensor to perform the task to be allocated, reflecting the results presented in Figure 5.14. In contrast, A1 performs only one allocation at mission startup. This way, when context change events occur, tasks are not reallocated even when

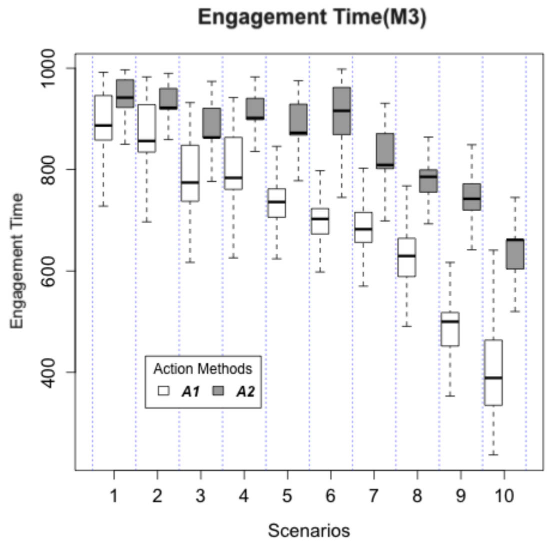



**Figure 5.11:** Engagement time (M3)

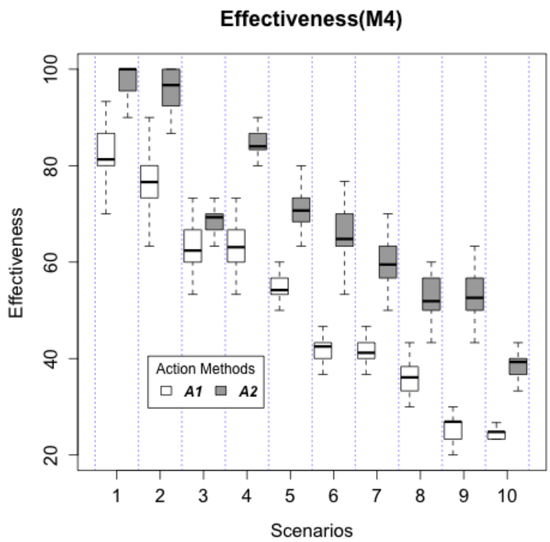



**Figure 5.12:** Effectiveness (M4)

employed sensors are damaged. Besides, if all 30 tasks in the mission are performed by the most compatible sensor for each one, i.e., with quality 1, we obtain the maximum result of 30 for reward (M6) metric.

Since removing members with the *memberFailure* event does not allow members to reconfigure, prompting instead a task reallocation or even a C2 Approach change, it explains the decrease on M1, M4, M5, and M6 in Scenario 3 for method A2. In contrast, *sensorFailure* and *envChange* enable member reconfiguration or change to the C2 Approach to search for an alternative allocation of pending tasks based on the algorithm used, i.e., swarm-gap based [13, 108], and the available context information. It gives to A2 advantage in all other scenarios. Moreover, for methods A1 and A2, it is possible to observe a significant reduction in results for M3, M4, M5, and M6 due to the progressive loss of system resources such as members, sensors, and autonomy, as exercised by the scenarios.

Nonetheless, even though both M3 and M4 decrease as scenarios become more complex, they do so at different rates. Indeed, mean engagement time decreases less than 20% before scenario 8, whereas mean effectiveness decreases about 46% for the same scenarios (cf. Table 5.5). This trend suggests that engagement time is not necessarily an indicator of effectiveness, strengthening the claim made by Alberts and Hayes [10] that "the quality of C2 should not be deduced solely from mission outcomes". In such a scenario, the compatibility between the sensors applied, and the tasks is fundamental to the evaluate the mission results.

Although the difference between the results with action methods A1 and A2 for all scenarios and metrics is graphically observable, a statistical test was performed to confirm

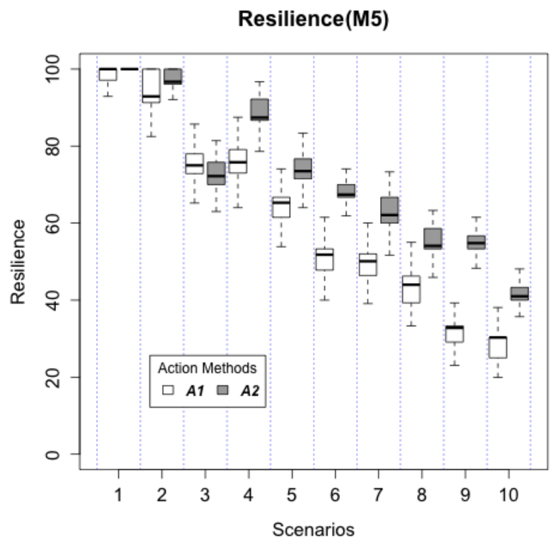



**Figure 5.13:** Resilience (M5)

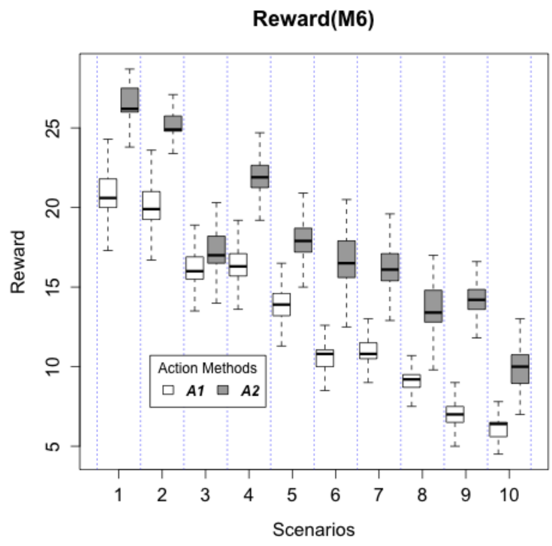



**Figure 5.14:** Total reward (M6)

these differences. With a Shapiro-Wilk test [58], a *p-value* less than 0.05 was obtained, thus indicating a non-normal distribution. Accordingly, an Mann-Whitney U Test described in Kassambara [77] was applied, i.e., Wilcoxon-test in R, to check differences among the samples of action methods A1 and A2 for all scenarios. According to the number and value of samples, the statistical analysis confirmed the difference between all results from A1 and A2 methods for all simulated scenarios. All *null hypotheses* $H_0$ in Table 5.2 were then refuted in favor of the alternative hypotheses. A2 has statistically significant higher values for all metrics, except for Resilience (M5) in Scenario 3, in which case A1 has a 7% higher value for the reasons discussed previously. Therefore, the simulation results provide evidence that the proposed model provides C2 Approach Agility (*RQ1*) and C2 Maneuver Agility (*RQ2*).

> In summary, simulation results suggest that the proposed model exhibits C2 agility. Indeed, the system stays more time in action, completes more tasks and more compatible ones, resulting in higher resilience thereby better coping with context changes and perturbations in dynamic scenarios.

## 5.2 Survey-Based Assessment

The validation of this study was carried out in the military domain. We selected such domain because its scenarios are naturally endowed with high dynamism, besides having, in its structure, roles and functions well defined and required for the proper implementation and deployment of concepts related to C2. Another factor that contributed to the choice of this domain was the proximity of the research group and support received by the Brazilian Army.

Conceptually, the existing organizational structures in the military domain employed for fulfillmenting their missions related to the final activity have hierarchically organized elements and are compatible with the model proposed in this work. In addition, the need for these structures to respond to changing circumstances is a relevant factor for the public in question.

To perform the validation of this empirical study, we applied a questionnaire to the chosen domain in order to gather perceptions about the completeness, usability, and compatibility of the proposed models with realistic scenarios, as well as to validate the terms and definitions applied. We adopted the use of questionnaires because it allows us to get closer to the domain that holds the most knowledge in this area, particularly in Brazil, and because realistic tests are not workable because of their financial costs. Even the simulation being focused on tactical level, we do not segregate the respondents in tactical and strategic levels. All professionals that work at the planning level have already worked at the operational level. Based on this, we do not consider the respondent group heterogeneity.

The questionnaire's design meets three quality requirements according to Pfleeger and Kitchenham [100]:

1. The questionnaire avoids bias so as not to influence responses. It took care not to induce the respondent to a particular response by withholding valuable information.

2. The questionnaire is appropriate for the domain and its questions. Besides being closely related to subjects' knowledge of the domain, it presents complexity compatible with the target audience.

3. The allocation of resources for the application and analysis of the results is compatible with reality and does not require excessive time commitment and application of other resources by the respondents. The low cost is in line with the high value and interest of the subject on the agenda by the members of the domain.

The design chosen for the questionnaire is the Descriptive, and Case Control sub-type, which makes use of the analysis of previous situations experienced by the respondent, in order to help explain the effects and aspects that are the object of the study. The presentation of simulated scenarios, in order to provide the elements of analysis, also explores the capacity of the respondent based on their previous experiences and knowledge.

In Section 5.3, we review essential aspects of the research method for this assessment. Sections 5.4 and 5.5 apply the research method to validate the models proposed and the

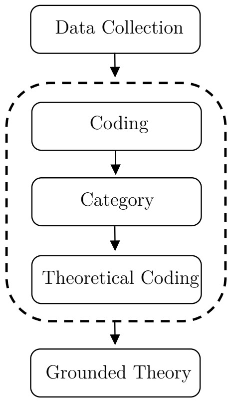



**Figure 5.15:** Ground Theory based method applied in the survey analysis (adapted from Hoda et al. [66])

study's relevance. We explored design elements and results found to identify important elements that collaborate with our study. In addition, Section 5.6 discusses the threats to validity related to the surveys applied.

## 5.3 Survey Method

Surveys are valuable tool in a research process. It permits collecting data, perception, and information from a group of respondents. However, it requires planning, execution and responses analysis to guarantee its reliability [100].

### 5.3.1 Data Analysis

An analysis type performed with the survey's data is called *coding*, that is based on the Grounded Theory method [66]. The coding or open-coding is an important tool to analyze the transcripts collected with open-ended questions. The code is the transcript's key idea represented by few words. We compare different codes from the same question to generate a common code when possible. The major concern from a set of codes forms the category.

Each statement has a set of codes obtained from the open-ended questions' answers. Analyzing these sets in higher level, we get the categories that summarize the key idea of this set of codes. Next, we organize the categories, removing repeated ideas and correlating similar ones in a so-called theoretical coding process. Finally, theoretical outline generated because of sorting and repeated theoretical coding process. Such ideas are organized and ordered to compose the theory.

For close-ended questions, we apply a graphical analysis to identify the participation levels of each response within the universe of respondents. According to Kitchenham and

Pfleeger [79] and based on the responses' observation, we can identify trends and the understanding level of what is being presented and asked.

### 5.3.2 Survey Reliability

It is appropriate to submit a survey to a consistency analysis of its scale to assess the magnitude of the questionnaire items correlation. The *alpha* coefficient, i.e., *Cronbach's alpha* [38], is an important tool for performing such a measurement. In summary, alpha is the average of the instrument items' correlations. Such an instrument is the questionnaire, and the items are its questions.

Considering a matrix $M_{n \times k}$, where $n$ is the number of respondents for $k$ questionnaire items, Leontitsis and Pagge [85] writes *Cronbach's alpha* as shown in the Equation 5.4. As $M$ handles the quantified responses from the questionnaire, such values are within a scale defined during the survey planning [79], e.g., 1 (fully disagreement) to 5 (fully agreement).

$$\alpha = \frac{k}{k-1}\left(\frac{\sigma_\tau^2 - \sum_{i=1}^{k}\sigma_i^2}{\sigma_\tau^2}\right) \tag{5.4}$$

The $\sigma_i^2$ is the variance of the responses for each questionnaire item $i$. Besides, $\sigma_\tau^2$ is the variance of the responses sums of each respondent, i.e., the variance of each line sum in $M$. It is important to emphasize that $k$ and $n$ must be greater than 1. Besides, he $k$ works in the Equation 5.5 as a correcting factor.

When the responses are consistent, the $\sigma_\tau^2$ becomes large and as a result, $\alpha$ approaches 1. If there are random responses, the $\sigma_\tau^2$ and sum of $\sigma_i^2$ become similar and $\alpha$ becomes small. The scale used for each questionnaire item and the number of respondents can affect in such results.

Described by Cronbach [38], the $\alpha$ coefficient shows the survey instrument reliability. Such a statistic emphasizes the instrument's items consistency. According to David L Streiner [40], the expected value for $\alpha$ is between $0,70$ and $0,90$. If $\alpha$ is less that $0,70$, it shows low consistency level for the scale applied. If the value is greater than $0,90$ we can identify redundant or repeated items and we must exclude them from the sample.

## 5.4 First Survey

This section shows details and analysis about the first survey applied to collect feedback from military domain on our study. Sections 5.4.1, 5.4.2, 5.4.3, 5.4.4, 5.4.5, and 5.4.6 describe the survey planning steps. Besides, Section 5.4.7 describes the survey's execution analysis and its analysis.

### 5.4.1 Goal and Research Questions

To validate the concepts used by the proposed dynamic model based on Channel System (cf. Section 4.3), besides assessing its relevance and compatibility with the military domain scenarios, we performed a survey to collect feedback from domain experts. In addition, the goal is to identify possible real military dynamic scenarios that present C2 agility concepts to be used in our study. Based on this, the survey tries to answer the following research questions:

- **RQ1**: Are the simulated scenarios relevant to the military domain ?
- **RQ2**: Could we add elements in the proposed model to make it more realistic ?
- **RQ3**: Is C2 agility a well-known definition ?
- **RQ4**: Was the proposed model well understood ?
- **RQ5**: Are there other dynamic scenarios in the military domain that apply C2 concepts ?

### 5.4.2 Survey Design

We performed a survey divided in two parts with questions in different level of detail and perspectives to immerse the respondent in the concepts and leave him/her free to position himself/herself according to the doctrines and his/her specific experiences in the organ, unit or department to which he/she is linked. To gather information more precisely about C2 concepts in the military domain, we write the underlying questionnaire in the respondent's natural language, i.e., in Brazilian Portuguese, and Appendix A shows the original questions and their objectives. We collected the data using Google Forms Personal tool [60] through a link sent by email to potential respondents. In addition, based on the strategic interest of the subject studied within the military domain, inciting a relative secrecy of the information provided, we distributed the questionnaire anonymously to make the respondent more comfortable and able to give spontaneous answers and faithful to domain reality.

We followed the activities presented by Pfleeger and Kitchenham [100] to define and to design the survey process. In summary, we organize survey's structure in sections as follows:

- Section 1 (Dynamic Model Assessment): We present a software simulation that represents UAVs application in a reconnaissance mission and applying C2 concepts. We evaluate the dynamic scenarios presented according to the relevance and adherence to the military domain;

**Table 5.6:** Open-ended questions applied in Survey #1.

| Section | Question | Description |
|---|---|---|
| **S1** | **Q2** | Scenario 1 simulates the use of UAVs where there are responses to context changes. Do you agree this scenario applies to your organization/unit ? Justify your answer. |
| **S1** | **Q3** | What could be included/changed in this scenario in order to make it closer to the reality of interest to your Organization/Unit? |
| **S2** | **Q1** | Describe a dynamic scenario planned or experienced by your Unit or Organization with the application of C2 principles (maximum of 30 lines. Do not insert sensitive information, just treat it as generic or publicly known cases. You can use fictitious nomenclature if you think it is necessary and suitable) |
| **S2** | **Q5** | In the scenario you describe, is the Unit or Fraction endowed with AGILITY? Justify your answer. |

**Table 5.7:** Correlation between RQ and survey questions

| RQ | Questions(Section 1) | Questions(Section 2) |
|---|---|---|
| RQ1 | 1, 2 | – |
| RQ2 | 3, 4 | – |
| RQ3 | 6, 7 | 4, 5, 6, 7 |
| RQ4 | 5 | 8 |
| RQ5 | – | 1, 2, 3 |

- Section 2 (Military Domain Scenarios): We ask to the military domain experts about possible realistic dynamic scenarios that present the concepts of C2 agility, based on their professional experience. To avoid confidentiality issues, we encourage the respondents to consider information about simulated scenarios.

Open-ended and closed-ended questions were used to assess the proposed model and to collect new scenarios from the military domain. Section 5.4.7 shows fifteen questions with their results and quantitative analysis. Besides, Table 5.6 shows the open-ended questions of this survey on which coding was applied, and Table 5.7 maps the survey's questions to the overall research questions.

We have created support materials, available through a link to GitHub repository[2], that include a set of slides for knowledge leveling and a reference source for the respondent, as well as a video of the execution of the simulator showing the use of UAVs in

[2]https://github.com/junieramorim/survey/tree/main/Survey_1

reconnaissance missions. The closed-ended questions use a scale showing a sort of values, which may vary from complete agreement, with the level 5, to complete disagreement with level 0. Besides the agreement scale, we use scales with number of UAVs, tasks and sensors. The open-ended questions were constructed to avoid leading respondents or inducing them with part of an expected response.

We did the survey planning and scheduling according to the organization's availability, and we applied it from June 2020 to August 2020, informing a deadline of fifteen days to answer it after receiving the invitation by email. The resources in terms of survey respondent were obtained through a previous contact with the organizations. Even after the deadline, we left the survey open for 45 days because of the difficulties in getting respondents with the right profile available.

### 5.4.3 Pilot Tests

To ensure the adherence between the simulation presented in the survey and the domain scenarios and concepts, as well as the type of question adopted, we submitted the survey to three domain experts with knowledge in technology, engineering, and command and control in military operations. Their comments gave us new ideas and suggestions to make the questions clearer to the respondents. In addition, we increased the simulation scenario with additional issues for the UAVs to characterize context changes. After survey's adjustment, it was distributed by email, and only one of the pilot tests respondents filled the survey's final version. Furthermore, such tests showed preliminary evidence of the study's relevance to the military domain.

### 5.4.4 Population and Sample

The survey's questionnaire was distributed to military personnel who exercise command, leadership, and/or management functions, who see the coordination of diverse resources for the execution of a task. Some of them have already had contact with the concepts of Command and Control, whether in a practical or theoretical way. The general population used in this survey comprise the staff of the following Organs: Ground Operations Center (COTER), Systems Development Center (CDS), Ministry of Defense, and Missile-launching Artillery Unit. From these organizations, we considered as target population the total number of personnel that meet the following criteria:

- Military personnel working on the documentation of doctrines related to the application of C2 concepts;

- Military personnel who work in the development of technological tools that help the application of the C2 concepts in the Organization's final activities;
- Military personnel of different hierarchical levels and with different functions within the planning process of operations using C2 concepts;
- Military personnel of different hierarchical levels and with different functions within the process of applying C2 in operations;
- Military personnel with a leading role in operational activities;

From this target population, according to the rules and authorizations of each Organ/Unit, the samples are defined through a stratification (strata) following the criteria below:

- Officers who plan actions using the C2 concept (Ground Operations Center-COTER, and Ministry of Defense-MD)
- Officers exercising command actions in operations using C2 concepts (Artillery, Communications and Infantry personnel in operational units)
- Military personnel responsible for process modeling and tools development used in C2 operations (Systems Development Center-CDS)

The COVID-19 pandemic period affected the size of the samples. The organs were working with a reduced number of on-site work. We received some feedback about the unavailability of the respondent and about doubts about the confidentiality of the subject.

The size of the samples depends on the organ and on the corresponding authorization of the managers directly involved or other particular requirements of the interviewed organization. According to the design of the questionnaire and the characteristics of the organizations from which we draw the population, we adopted the sampling for convenience. Besides, because of the pandemic period with interaction restrictions, we have chosen organs/units with easy access.

### 5.4.5 Data Collection

From the survey distribution through emails, we collected 17 responses, and this low number may be related to the confidentiality level applied to this subject by the organization. After a preliminary analysis, we excluded two responses because of incorrect filling out. Such answers presented meaningless content, and they had most of the aim answers unmarked. In addition, we identified two answers duplicated. We analyzed these

| Section.Question | Theoretical Code |
|---|---|
| S1.Q2 | context change, focus on the mission, timely response |
| S1.Q3 | context change, focus on the mission |
| S2.Q1 | importance of communication , focus on the mission, timely response |
| S2.Q5 | context change, focus on the mission, timely response |

**Table 5.8:** Theoretical coding results

responses to check if the respondent had the intention to correct or change the answer, and after confirmation, they were excluded. Based on this, we evaluated 13 remaining answers.

### 5.4.6 Data Analysis

We applied the technique described by Seaman [109] to coding the open-ended questions (cf. Table 5.6) and to perform a qualitative analysis. Answers to these questions are listed in Appendix B.

We employed visualization techniques, such as bar charts and descriptive statistics, to analyze quantitatively the answers from the closed-ended questions. In addition, there is an open question asking for possible new scenarios, even other simulated by the military domain. The coding in such a case was based on the key elements identified in the answers and related to C2 agility, e.g., battle groups operating in a coordinated way. In all scenarios suggested we could identify entities operating in an environment under a specific coordination strategy so-called C2 Approach, to accomplish a mission.

In addition, as described by Kitchenham and Pfleeger [79], we sorted and categorized the coding results to write theoretical codes and finally the grounded theory.

### 5.4.7 Results and Analysis

This section presents and analyzes the data from the questionnaire's answers per research question. In addition, we leverage the grounded theory previously formulated with the open-ended questions. After performing the coding process, we sorted and grouped the codes according to similarities. Next, we organized them in categories, making a comparative analysis so-called theoretical coding. Table 5.8 shows the result of this step according to Section 5.4.6.

As the responses are similar, even being from different questions, we analyzed these theoretical codes, and we generated an unique grounded theory transcribed below:

| |
|---|
| Communication requirement to deal timely with context changes aiming mission accomplishment. |

This section applies such a theory to address some research questions listed in Section 5.4.1. The first part of the survey presents the questions and corresponding answers about a hypothetical scenario created with the implemented simulator, which simulates a team of autonomous drones performing a reconnaissance or surveillance mission. The second part of the questionnaire is non-mandatory and the respondent can include it after completing the first section. However, all those who answered the first section agreed to continue with the second part.

#### 5.4.7.1 RQ1: Are the software simulated scenarios relevant to the military domain ?

Questions 1 and 2 of the survey's first section address this research question. Question 1/Section 1 (*"Scenario 1 simulates UAV use of software, where there are responses to context changes. Do you agree this scenario applies to your organization/unit ?"*) presented responses concentrated on the two highest levels. Figure 5.16 shows such responses. It suggests the relevance of the scenario for the military domain where level 5 shows fully agreement with the statement.

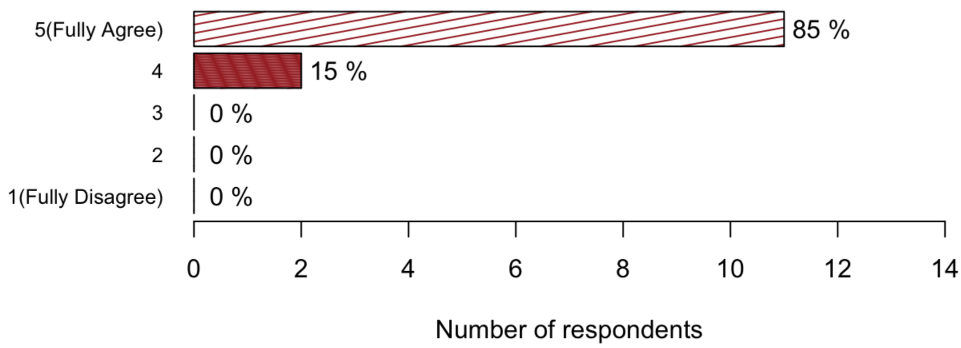



**Figure 5.16:** Responses to the Question 1 / Section 1.

Question 2/Section 1 (*"Justify the previous answer."*) extends the previous one and allows us to obtain more detailed feedback from the respondent. With the coding process, this question collaborated with the idea of context change, that is part of the theory described in Section 5.3.

The crucial aspect highlighted is the dynamic context in all military operations, identified by the reference made by the respondents. As an example, we have the following transcribed entry: *"Change of scenario is something very common in real missions"*.

Such a statement occurred in 61,5% of the answers, confirming the importance of context change support according to the theory presented in Section 2.1.4.

Another important and complementary aspect was the citation of real drone usage scenarios in reconnaissance missions by some respondents who have had this experience, e.g., army operations in the city of Rio de Janeiro [34], and peace keeping missions by the Brazilian Army in Haiti [46]. This emphasizes the relevance of the simulated scenario used as well as its adherence to the military domain. As discussed in this section, we can identify the simulation's relevance to the military domain.

##### 5.4.7.2 RQ2: Could we add elements in the proposed model to make it more realistic ?

The open-ended Question 3/Section 1 (*"What could be included/changed in this scenario in order to make it closer to an actual situation of interest to your Organization/Unit?"*), collected a sort of suggestions written in different formats. We performed an open-coding with the technique described by Juliet M. Corbin [74] to organize the answers in groups and extract the key idea of each answer. The results are the following with their proportion:

- 38,5% : The scenario represents the reality well.
- 23,0% : Additional threats and UAVs inclusion.
- 15,4% : The existence of multiple coordinating elements and that the drones could switch roles during the execution.
- 15,4% : Including the map of the region and transmission of information to the central command.
- 7,7% : Insert an operational code in order to facilitate the logistics chain.

Although there is a slightly higher percentage of respondents who agree with the compatibility of the simulated scenario with reality, we can identify a very scattered result and probably motivated by the difference in the functions and specialties of the various respondents. Based on this, there is no unanimity for a particular element that may miss from the simulation environment. However, we observe the assessed model supports most of the responses. All members can switch roles according to context changes impact and C2 Approach operated, and this adaptation occurs in runtime. Additionally, we can improve the system with map and interaction data with no change in the model.

The model does not support the threats and UAVs addition in its current configuration, and such insertion requires new events that guide the system to a variety of states to reorganize the allocation and task execution. However, since the model supports adaptation to deal with context changes, there is evidence of resilience in face of such perturbations. Similarly, this study does not address the relation with logistics chain.

In addition, the following response: *"Interaction with aircraft and armor"*, recaptured the idea and need to work with hybrid teams. The proposed dynamic model supports such a feature as long as the unique elements can be implemented and/or treated as a Dynamic Software Product Line (DSPL), i.e., able to perform self configuration and to activate a communication capability.

After constant comparative analysis of these codes to identify some relationship between categories, we obtained the theoretical codes to result in the theory presented in Section 5.3.1. Such a theory brings the requirements that need to be present in the new scenarios created. The scenarios presented already met these requirements.

Analyzing the answers of the Question 4/Section 1 (*"From your point of view and experience, what would be the quantities for each element of Scenario 1 presented ?"*), we see the most answers showing 5 UAVs, which is probably more compatible with the organization. Maybe a number with 6 or more UAVs are out of the reality in terms of control capacity and existing resources. Figure 5.17 shows the suitable team size, i.e., number of UAVs, according to the respondents to perform the mission.

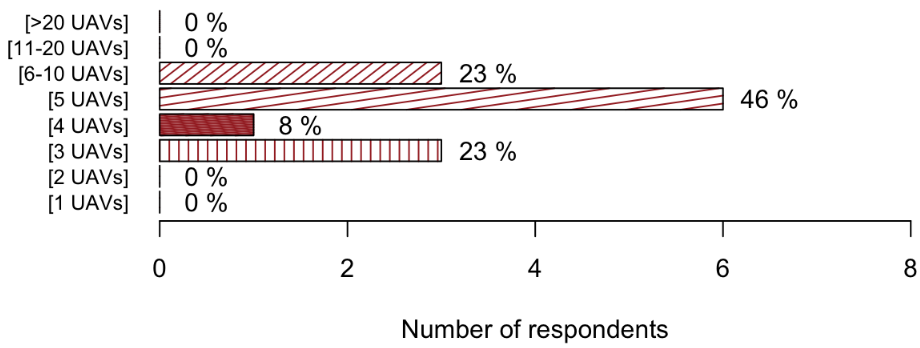



**Figure 5.17:** Answers to the Question 4 / Number of UAVs

In this same question, we could identify that at least 5 tasks are recommended, and a minimum of 1 task per agent. Such a result shown in Figure 5.18. Besides, multiple sensors onboard are common and suitable to the respondents. Figure 5.19 shows a similar number of answers to 3 up to 5 sensors in such a case.

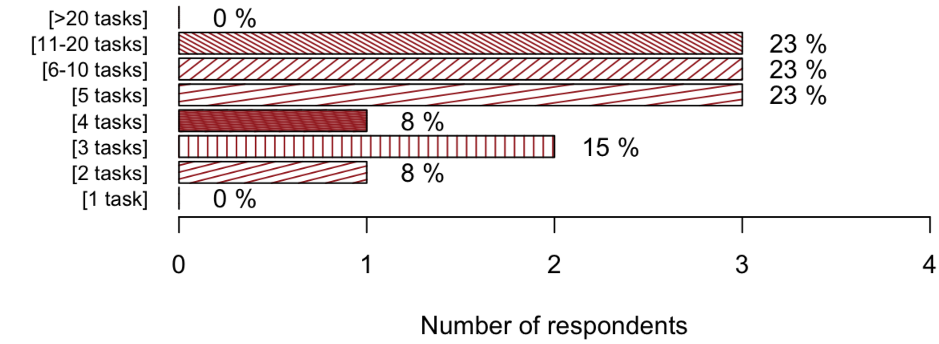



**Figure 5.18:** Answers to the Question 4 / Number of tasks

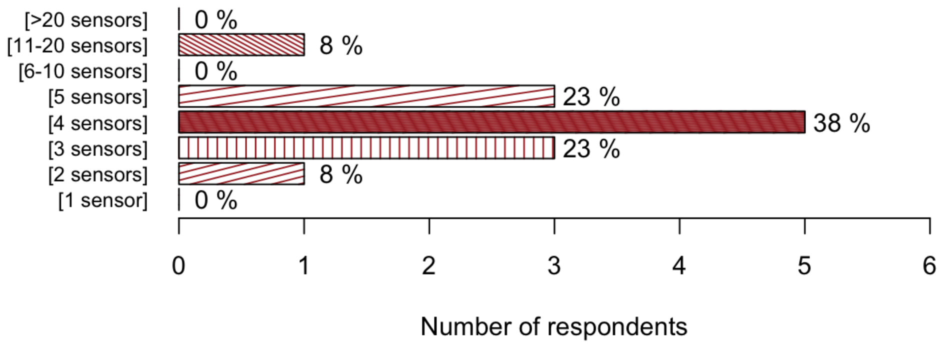



**Figure 5.19:** Answers to the Question 4 / Number of onboard sensors

Overall, Question 4 (Section 1) showed that the study environment created meets the expectations of domain experts, and suggesting it to be compatible with the scenarios handled by these professionals. However, the model is able to support any quantity suggested by the respondents. What limits this capability is the hardware capability of the employed elements, i.e., UAVs.

Based on the responses, we can see that the proposed model can meet all suggestions for inserting objects with little or no change. Most of the respondents consider the model already close enough to reality. Such an analysis indicates a positive response for RQ2.

#### 5.4.7.3 RQ3: Is the C2 agility a well-known definition ?

Figure 5.20 shows 92% agreement to the statement made by the Question 6/Section 1 (*"Reviewing the slides and the video describing the Scenario 1, we can state that the system composed of UAVs has agility. What is your level of agreement with this statement ?"*). It shows two important aspects: the respondents understood the concept of agility, and they identified the elements of the scenario created. The system agility to deal with context changes was properly identified. Such an analysis highlights the importance of agility in C2 context.

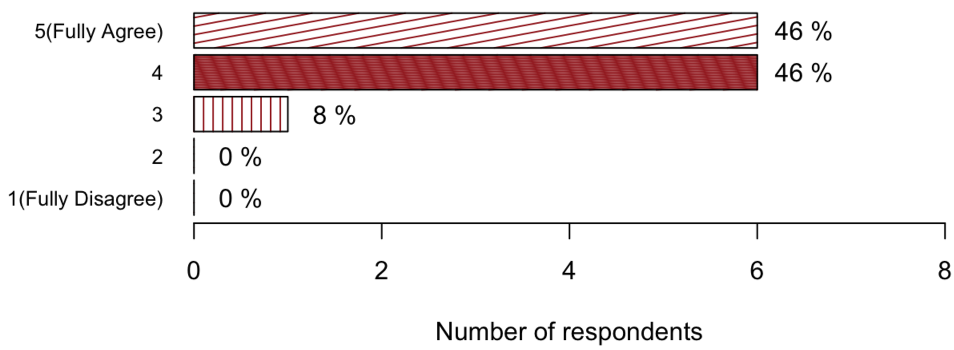



**Figure 5.20:** Answers to the Question 6 / Section 1

The closed-ended Question 7/Section 1 (*"Considering Scenario 1 from the presented slides, which of the features listed below gives C2 Agility to the UAV team ?"*) tries to identify the respondents' comprehension level about C2 Agility. The answers obtained are listed below with their respective occurrence ratio:

- a) Ability to reconfigure members to make them compatible with the tasks : 92,3%
- b) Ability to ignore the new context and its demands : 0,0%
- c) Ability to redistribute/allocate mission tasks among members : 92,3%
- d) Ability to give up the mission : 7,7%
- e) Ability to change C2 strategy to achieve a level of awareness : 92,3%
- f) None of the previous options : 0,0%

Such answers show the domain experts' understanding of the concept of agility applied to the simulated environment. In addition, we clearly observed the need to deal with context changes by adapting the system to the new conditions.

In addition, Question 4/Section 2 (*"In the Scenario you describe, is the Unit or Fraction endowed with AGILITY ?"*) tries to extract agility information about the respondent's example scenario. Figure 5.21 shows the answers to this question and highlights the agreement about agility presence in the example described by the respondent.

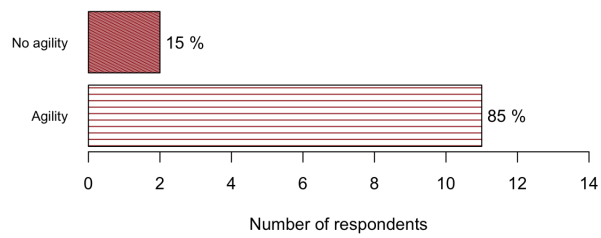



**Figure 5.21:** Answers to the Question 04 / Section 2

To give the opportunity to the respondent to give more information about the previous answer, we defined Question 5/Section 2 (*"Justify the previous answer."*). Based on this, we can identify in the responses, descriptions of adapting the unit to new conditions, or redeploying the limited resources for continued mission accomplishment. In addition, the answers refer to other generic capabilities of articulation applied with relative autonomy to adapt to the new context. All adaptations aim to continue the mission execution and accomplishment. Some of the answers representing this idea are transcribed below:

*(1) The units had autonomy to follow different decisions.*

*(2) By the ability to reorganize itself in function of an eventual change in the foreseen scenarios.*

We had an answer based on a financial scenario that describes the resources distribution according to the plan:

*"The mission accomplishment established by the annual budget or imposing by the operational necessity and urgency are sufficient and necessary reasons to demand agility from the actors involved."*

However, such a distribution and planning must be adapted according to context changes. These changes involve budget assignment changes and resources modification. The scenario, although not connected to military missions, shows C2 agility elements and becomes an interesting example. These comments were fed into the coding process,

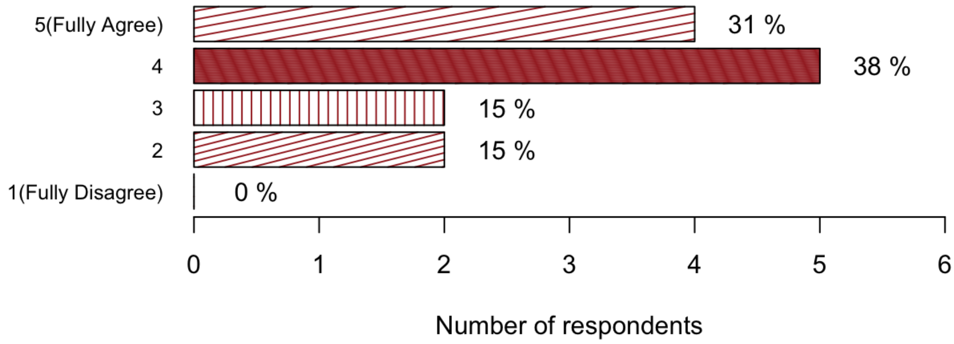



**Figure 5.22:** Answers to the Question 6 / Section 2

leading to the theory presented in the Section 5.3.1 that confirms the answer to this research question.

Analyzing the answers of Question 6/Section 2 (*"The total mission accomplishment, including all the tasks that compose it, is fundamental. What is your level of agreement with this statement ?"*) presented in Figure 5.22 we classify such mission completeness as a relevant requirement. Domain experts emphasized the importance of finishing all tasks.

Figure 5.23 shows a relatively widespread result to the Question 7/Section 2 (*"The mission must always be accomplished in the shortest possible time, even with the quality of its execution compromised. What is your level of agreement with this statement ?"*). However, we can observe a distribution of the answers to the result of disagreement, contrary to the results of previous questions where there is a tendency of total agreement with the statement. It shows the not absolute understanding of agility.

This occurs because the military domain values the concept of *fast*. In many scenarios, these experts presented the fast concept as a necessary requirement to the mission accomplishment.

According to the previous analyses, we identify a sufficient understanding of the C2 Agility concept. Although the agility concept in the military domain is related to the speed of the execution, we note the goal of not specifying the precise speed of the project as the key aspect.

#### 5.4.7.4 RQ4: Was the proposed model well understood ?

The classification asked by the Question 5/Section 1 (*"Slide 20 characterizes Scenario 1 as being composed of 1 UAV performing the roles of coordinator and task allocator,*

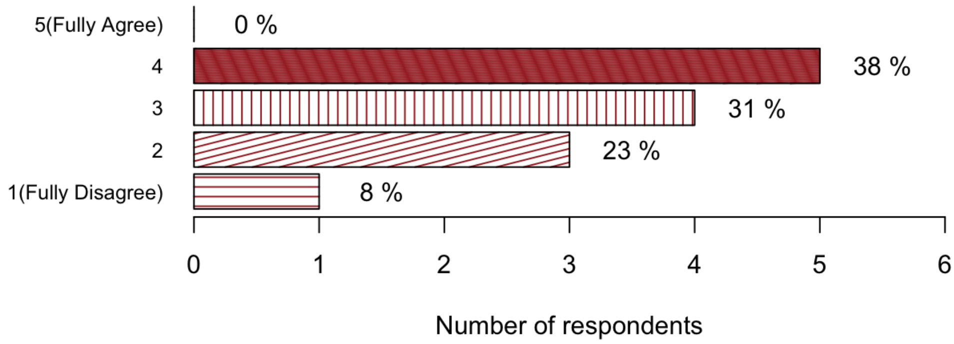



**Figure 5.23:** Answers to the Question 7 / Section 2

*and the other UAVs with the role of task performer. In this context, the coordinator has total autonomy (it does not depend on a central command) and uses the C2 strategy to exchange information with the rest of the group/fraction. Considering the list below of system responses to eventual context changes, how would you classify each one of them in terms of relevance/importance ?"'*) allows us to verify the correct identification of important elements in the scenario.

The answers show that context changes require some action from the system. Such a reaction is compatible with the treatments by the simulator in the Scenario 1, characterizing C2 Agility, i.e., dealing with context changes. However, as shown in Figure 5.24, not performing the tasks is the worst choice, and it requires the system to adapt to the new conditions seeking to continue the execution of the mission.

According to the answers of the Question 8/Section 1 (*"Which role(s)/function(s) can be identified in the scenario presented by you ?"*), all roles are presented in the example described by the respondent. Figure 5.25 shows how suitable these specified roles are to a real scenario in the military domain. Besides, we can see the role in the proposed model as a summary of all functions actually used by the domain modeling.

Based on these results, we observed that there was an adequate understanding of the proposed model, making it possible to transfer the concepts presented by it to other scenarios. In addition, the respondents understood the concepts related to the roles defined in the model.

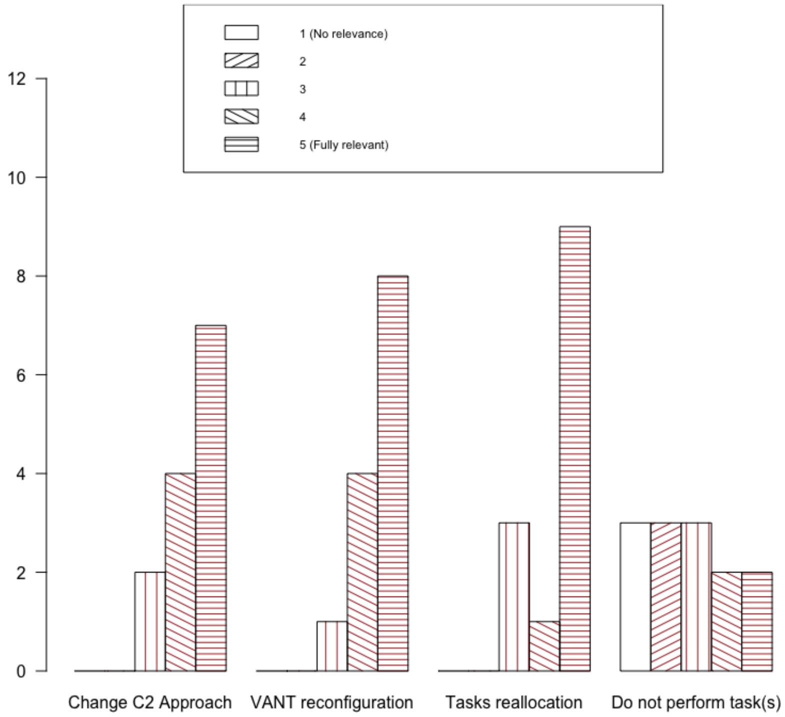



**Figure 5.24:** Answers to the Question 5 / Section 1 showing the relevance level (from 1-No relevance, to 5-Fully relevant)

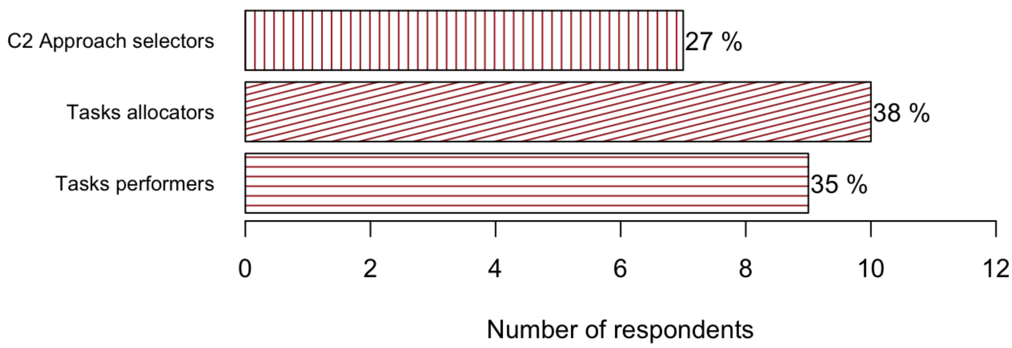



**Figure 5.25:** Answers to the Question 8 / Section 2

#### 5.4.7.5 RQ5: Are there other dynamic scenarios in the military domain that apply C2 concepts ?

The concepts related to C2 are under active study and application in the military domain [10]. With Question 1/Section 2 (*"Describe a dynamic scenario planned or experienced by your Unit or Organization with the application of C2 principles (Maximum of 30 lines. Do not insert sensitive information, just treat it as generic or publicly known cases. You can use fictitious nomenclature if you think it is necessary and suitable)"*), we could confirm C2 elements in different real scenarios.

All respondents described different real-life scenarios that apply to the concepts of C2 and C2 Agility. They were concise descriptions of different operations where one could easily identify changing circumstances during the operation by imposing the risks involved. The need to adapt the unit to the loss of the communication signal, and to readjust the structure because of elements damaged during the operation, presents C2 Agility concept in the scenarios described by the respondents.

The respondents cited the scenario of operation in urban environment as highly dynamic and under new constraints with the requirement to continue operating. This allows us to identify C2 Agility as an important resource applied in real scenarios where the respondents worked. Related to this, we highlight the following response: "A troop deployed to the front to perform a specific task moves over rough terrain so that during its movement there is an intermittent loss of communication between its elements and with its base.". In addition, we find two responses that describe similar scenarios of reconnaissance and the common occurrence explains this in an operational context during regular military maneuver.

To analyze the responses given, we employed the following coding:

- **Cod-1**: Domain scenarios are under context changes and present C2 concept elements.
- **Cod-2**: Domain scenarios are under context changes and present some C2 concept elements.

The distribution of Cod-1 and Cod-2 over the responses was 38,5% and 61,5%, respectively. Overall, we notice at least one C2 concept's element in the scenario described. In such a case, the most common was dynamic context, i.e., the scenario is under some change during the mission accomplishment.

Furthermore, the ratio for each C2 Approach identified by the Question 2/Section 2 (*Based on the description of the C2 Strategies presented in the slides provided, which one is being operated by the Organization/Unit in the scenario described by you ?"*) is listed below.

- Collaborative : 61,5%
- Coordinated : 23,1%
- De-Conflicted : 0,0%
- Conflicted : 7,7%
- None of the options : 7,7%

We observe the concentration on structure where there is at least one coordinator. A central element can perform this coordination when operating the coordinated C2 approach, or by multiple local coordinators in case of collaborative C2 approach. This aspect indicates a common element with the function of concentrating decisions and orders. In addition, we note that there is no solution as “one size fits all”, able to address all situations. The respondents themselves identified, in their scenarios, different C2 Approaches that can handle context changes.

In addition, for the Question 3/Section 2 (*“Based on the dynamic scenario in your Unit/Organization described by you, which of the following actions are applied in response to a context change during mission execution ?”*), we received the following ratios for each option presented:

- Rapid adaptation of the Fraction(s)/Unit(s)/Member(s) to the new circumstances : 76,9%
- Abort the mission immediately and retreat to base : 15,4%
- Change the allocation of the elements that compose the Unit, in order to redistribute the means for the completion of the tasks in the new context : 30,8%
- Stop the execution and report the new situation to the higher echelon, awaiting orders : 7,7%
- Do nothing and wait for favorable conditions for the execution of the mission : 7,7%
- Change the C2 strategy (structure of interaction of the elements) to adapt to the context : 46,2%
- Try to redistribute the unrealized tasks among the other members that can do it : 76,9%
- Accomplish the mission with lower level of performance, for example, instead of annihilating, just neutralize the enemy : 15,4%

- If not accomplishing the whole mission, accomplish one (or some) part(s) of the mission, example: the mission would be to neutralize an entire enemy battery, but it is only possible to neutralize a section of that battery : 30,8%

- None of the above : 0,0%

- OTHER : 15,4% (Fraction adaptation; Recovering, mutual support and task reallocation)

We note scenarios formed by police operations, peace-keeping missions, and rapid actions of small fractions were those cited by the respondents and all of them are characterized by the requirement of adapting to new circumstances. Such capability characterizes C2 agility. Based on this, we can confirm such a comment through the higher percentage showing the need for reconfiguration of the members, the change of the C2 Approach, and the redistribution of tasks among the elements. This shows that the C2's definition of agility is met by the elements selected in the questionnaire.

Despite not being mandatory, all respondents filled the second section of the questionnaire. Such questions explored any potential scenarios in the military domain where C2 concepts could be applied. Besides, we identified important elements and characteristics to deal with context changes that are required to make the scenarios presented by the respondents compatible with C2 and C2 agility concepts. Based on this, it positively answers RQ5.

### 5.4.8 Survey Conclusion

The closed-ended questions in Section 5.4.7 suggest the compatibility and relevance of presented simulated scenarios. In addition, to check the reliability of the survey, we calculate the Cronbach's alpha ($\alpha$) coefficient to validate the questionnaire internal structure. This analysis applied to the closed-ended questions that use the scale composed of the values 1 (fully disagreement) to 5 (fully agreement). Table 5.9 shows the results for each respondent to the questions that satisfy this form of response.

According to Section 5.3.2, we obtain $\sum_{i=1}^{k} \sigma_i^2 = 29.00$ for each analyzed question $i$ and the $\sigma_\tau^2 = 74.40$ from the survey's results of Table 5.9. This table represents the matrix $nxk$ with $n$ respondents and $k$ questions. Applying such results in Equation 5.4, we have:

$$\alpha = \frac{8}{8-1}\left(\frac{74.40 - 29.00}{74.40}\right) = 0.70 \tag{5.5}$$

Such a result of $\alpha$ shows a reliable questionnaire with consistency in the answers given to the listed questions, even though the value is at the lower bound for this coefficient.

**Table 5.9:** Responses for the closed-ended questions with scale 1-5 (in the format Section.Question) to Survey #1.

| Respondent | S1.Q1 | S1.Q5.1 | S1.Q5.2 | S1.Q5.3 | S1.Q5.4 | S1.Q6 | S2.Q6 | S2.Q7 |
|---|---|---|---|---|---|---|---|---|
| P1 | 4 | 3 | 4 | 4 | 2 | 3 | 4 | 4 |
| P2 | 5 | 4 | 5 | 5 | 1 | 5 | 2 | 2 |
| P3 | 5 | 5 | 4 | 3 | 3 | 4 | 3 | 3 |
| P4 | 5 | 4 | 5 | 5 | 3 | 4 | 5 | 4 |
| P5 | 5 | 4 | 4 | 5 | 2 | 4 | 2 | 2 |
| P6 | 5 | 5 | 3 | 5 | 1 | 5 | 4 | 3 |
| P7 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 |
| P8 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 1 |
| P9 | 4 | 5 | 4 | 3 | 3 | 4 | 4 | 4 |
| P10 | 5 | 4 | 5 | 5 | 1 | 5 | 3 | 3 |
| P11 | 5 | 3 | 5 | 3 | 2 | 4 | 4 | 4 |
| P12 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 2 |
| P13 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 3 |

**Table 5.10:** Summary conclusion related to research questions of Survey #1

| RQ | Responses analysis summary |
|---|---|
| RQ1 | All respondents confirmed the relevance of such a simulation with UAVs or other entities performing the same structure. |
| RQ2 | Some respondents have suggested some additional elements that could be added in order to improve the model and simulation. |
| RQ3 | The C2 agility concept is well comprehended definition to the respondents even when their scenarios do not have agility. |
| RQ4 | Most of the respondents understood the roles and the structure represented by the model. |
| RQ5 | Some respondents presented scenarios from their domain with C2 concepts included. |

Besides, the scale applied proved to be compatible and reliable, emphasizing the results from the qualitative analysis of the open-ended questions.

According to the qualitative and quantitative analysis of the responses, we revisit the research questions shown in Section 5.4.1 and show a summary for each question in Table 5.10.

Analyzing the responses, we can conclude that there was absolute agreement on the relevance of the simulated environment to the military domain, and the respondents understood all main concepts related to C2, including that C2 agility is not necessarily related to execution speed. Moreover, all scenarios from the respondents involve C2 elements to a certain degree, and all of them are in a dynamic context and require adaptation along the mission execution. Based on these aspects, this survey suggests military domain alignment with our study and validated the applied definitions.

## 5.5 Second Survey

This section shows the method to validate the structure and compatibility of the proposed static model, i.e., C2 meta-model (C2MM). Sections 5.5.1, 5.5.2, 5.5.3, 5.5.4, 5.5.5, and 5.5.6 describe the survey planning steps. Besides, Section 5.5.7 describes the survey's execution analysis and its analysis.

### 5.5.1 Goal and Research Questions

The goal of this survey is to collect the users' level of perception about the proposed static (cf. Section 4.2). We performed it creating scenarios resulting from context changes, using the static model and submitting them to the domain experts' evaluation. Such an evaluation assesses the usability and usefulness level of the model in terms of application in domain scenarios. Based on this, the survey aims to answer the following research questions:

- **RQ1**: Is the model easy to understand?
- **RQ2**: Is the model useful to the military domain?

RQ1 focuses on the ease of use of the proposed model in scenarios operated by the military domain; RQ2 focuses on the model's level of usability in the face of scenarios experienced by the military domain.

### 5.5.2 Survey Design

To provide all necessary guidance to the respondents, we organized the survey in ten sections. Section 1 explains the proposed model and Section 2 presents the initial configuration from where we create the manipulated scenarios. Sections 3 to 6 present four different scenarios submitted to the respondents' evaluation. All example scenarios follow a sequence of context changes, simulating a dynamic scenario similar to the one used in Survey 1. Section 7 explains the scale used in the responses and other instructions. Sections 8 and 9 group all survey's questions. Finally, Section 10 collects some respondent's feedback to identify work improvement opportunity.

We organized the questions of this survey in two groups, as described below.

- Group 1: Questions about the model's usability perceived by the users (perceived ease of use).
- Group 2: Questions about the proposed model's utility perceived by the user (perceived usefulness);

To gather information more precisely about these concepts in the military domain, we write the underlying questionnaire in the respondent's natural language, i.e., in Brazilian Portuguese, and Appendix C shows the original questions and their objectives. Similar to the Survey #1, we collected the data using Google Forms Personal tool [60] through a link sent by email to potential respondents and providing instructions to respond the survey. Such support material is available through a link to GitHub repository[3].

We base these two definitions on the Technology Acceptance Model (TAM) [41]. This theory applies subjective principles from Psychology, in order to measure the users' perception of technology. This process seeks an evaluation based on two variables: *perceived ease of use* and *perceived usefulness*. These variables seek to measure, subjectively, the receptiveness and added value of technology by users. These two groups of questions focused on the psychological theory highlight following objectives of this survey according to Pfleeger and Kitchenham [100].

The *perceived usefulness* variable seeks to identify the level of impact of the proposed solution on the respondent's domain, measuring the added value and positive impact on the internal processes of the environment in question with the use of the proposed solution. The *perceived ease of use* variable translates the level of understanding about the proposed model, characterizing its ease of application in the respondent's domain with affects on the learning curve and time spent learning the technology.

The closed-ended questions make the agreement level evaluation according to a scale ranging from 1 to 5 with the following value reference:

1. I totally disagree with the statement and there are no useful points;
2. I disagree with the statement, however some points are useful;
3. I neither disagree nor agree. Here, I can't evaluate the statement made, or it does not fit my profile;
4. I agree with the statement, however I could include and/or improve some points;
5. I totally agree with the statement and it covers the whole idea I have about the subject.

We use the open-ended questions to collect more information related to each closed-ended question, giving the opportunity to identify some hidden details or missing things in the diagrams generated from the metamodel. We divided the last section (Section 8) into three parts. The first and second, with 5 (five) and 10 (ten) questions, evaluate the perceived ease of use and the perceived usefulness, respectively. The third part has 3

[3]https://github.com/junieramorim/survey/tree/main/Survey_2

(three) questions to collect an anonymous feedback from the respondent, collecting ideas, suggestions and criticisms.

### 5.5.3 Pilot Tests

To ensure the appropriateness of the questions according to the survey's objectives, we submitted the questionnaire to two military personnel for a primary evaluation. We did a few adjustments in the questions and created a section to collect respondents' comments and new ideas. We improved the diagrams, i.e., increased the instances from the metamodel so that bringing more information to become them closer to the real scenarios.

### 5.5.4 Population and Sample

We distributed this survey by email to military personnel with some experience with Command and Control concepts application in the Brazilian Army to evaluate their perception about the subject. The pandemic period affected the size of the samples. The organs were working with a reduced number of on-site work. We have received some feedback about the unavailability of the respondent and about doubts about the confidentiality of the subject. Similar to the first survey, we distributed it to military personnel who exercise command, leadership, and/or management functions, who see the coordination of diverse resources for the execution of a task.

To obtain different views, we sent emails to members of the following Organizations: Training Center - South (CA-Sul), Systems Development Center (CDS), and Ministry of Defense. In addition, we sent to military personnel in the reserve who have extensive experience in various command and management roles. As this second questionnaire evaluates a higher level representation, capturing the perception of the respondents, we have been less restricted in the functions performed by the respondents. Here, we accepted military personnel who do not work directly with C2, but work in leadership, coordination, or management of projects, operations, and maneuvers.

We presented two questions in Section 10 of the survey to identify the professional experience level of the domain personnel. The first question asked how long the respondent has been in the military. It shows 57,1% with over 30 years of service, and the remaining 42,9% are between 15 and 30 years of military service. Based on this, we can observe a profile with a significant amount of career time.

The second question asked about the respondents' management experience. The answers showed that 85,7% of the respondents had over 10 years of experience in a management position, and only 14,3% with a time between 5 and 10 years. This confirms that

**Table 5.11:** Correlation between research questions and survey questions

| RQ | Questions(Group 1) | Questions(Group 2) |
|---|---|---|
| RQ1 | 1-5 | – |
| RQ2 | – | 6-15 |

the respondents' profile includes at least a partial knowledge of the concepts related to Command and Control, because of their functions exercised in the army.

Finally, we asked for suggestions, criticisms and comments about the questionnaire and the survey. This part is composed of open-ended questions that aim to bring new ideas to improve our study.

### 5.5.5 Data Collection

From the survey distribution through emails, we collected 11 responses and we justify this low number with the confidence level classification assigned to this subject in the military domain, and the time required for reading the support material. After a preliminary analysis, we excluded 1 response because the open-ended questions were not correctly filled in, and 1 answer with inadequate answers. Based on this, we evaluated 9 remaining answers.

To gather information more precisely from the military domain personnel, we wrote this survey to in the respondent's natural language, i.e., in Portuguese, and Appendix C shows the original questions and their objectives.

### 5.5.6 Data Analysis

We used bar charts and descriptive statistics to analyze quantitatively the answers from the closed-ended questions. In addition, there are open questions to collect the respondent's perception in a case of no complete agreement in the correspondent open-ended question. As the quantity of responses is small due to difficulties to obtain respondents in military organizations, no coding method Seaman [109] was necessary to perform a qualitative analysis, and we are analyzed each response individually. Table 5.11 shows the mapping between the research questions and the survey's questions.

### 5.5.7 Results and Analysis

This section shows the analysis of the questions' answers organized by research question. The first group of questions obtains the respondents' perception of the model's ease of use, whereas the second group obtains the respondents' perception of the model's usefulness.

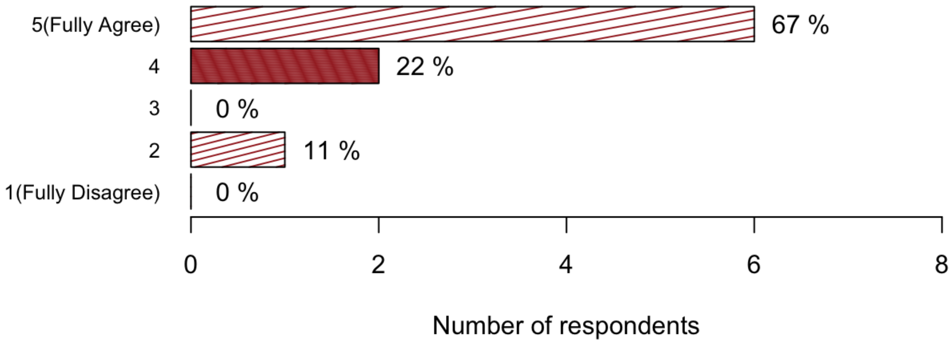



**Figure 5.26:** Responses to the Question 1 / Group 1.

#### 5.5.7.1 RQ1: Is the model easy to understand ?

To obtain the users' perception about how easy the model is to understand and to use, Question 1/Group 1 (*"It is possible to identify practically, in the scenarios listed above, all the key elements employed in a military domain operation (e.g. the composition of a platoon or battle group)"*). According to Figure 5.26, we see the highest concentration of answers to this question in total agreement (level 5) about the identification of key elements in the instances created from the model.

Question 2/Group 1 (*"Explain if you do not completely agree with the previous statement."*) collects all ideas from the respondents that do not totally agree with the statement presented in Question 1. Two comments show doubts about the model's ability to represent a hybrid team, as shown bellow:

*"For the example missions (UAV) the model was clear, but for the employment of troops with unique characteristics and missions acting together, I don't know if the algorithm will prioritize the missions."*

*"I missed a more complete hierarchy structure. Although each team has a leader, there is no overall leader element."*

However, the model is endowed with such capability with the connections between its former elements, but the drone example scenario presented does not explore it. In addition, we used Question 3/Group 1 (*"Which elements of a real scenario do you think it is not possible to identify or insert in the previous scenarios ?"*) to let the respondents free to identify any missing piece of information under their perception. Most of the respondents (80%) answered that the scenarios presented elements beyond the representations

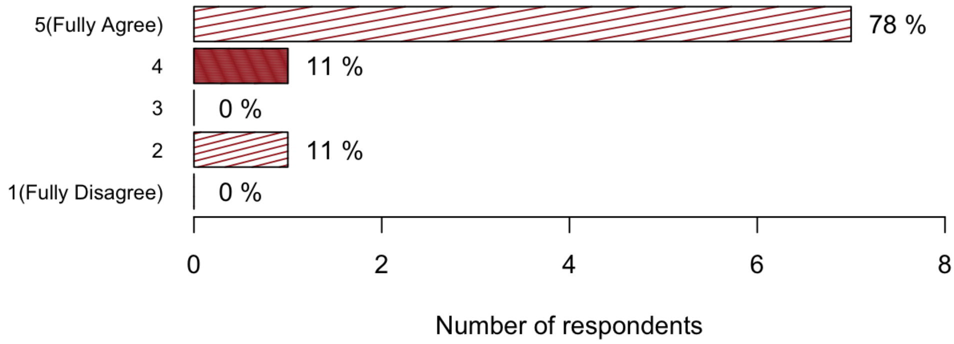



**Figure 5.27:** Responses to the Question 4 / Group 1.

used by them. In addition, one response brought up the need to include uncertainty in order to improve the current military models as transcribed below:

*"The uncertainty at any level."*

However, this characteristic is related to the behavior and it is more suitable to the dynamic model. As an interesting domain capability required, we transcribe another answer below:

*"Can I add more teams to work together with those existing?"*

The enquiry indicates a need for reinforcement during mission execution. Based on the static model, it is possible to obtain another situation's snapshot with another team's configuration. It makes this model compatible with such a requirement since it represents the state of the system at a time.

Question 4/Group 1 (*"It is possible to represent the fractions of Military Organizations in the operation being employed in a mission accomplishment and using the structure employed to represent the scenarios presented above."*) showed a concentration in the fully agreement, characterizing a good perception about the compatibility between the model and the domain scenarios. Figure 5.27 shows such a result.

Although there is no unanimous agreement on the previous statement, Question 5/Group 1 (*"Explain if you do not completely agree with the previous statement."*) did not receive any comment about the previous question. Thus, we considered the total agreement to the statement.

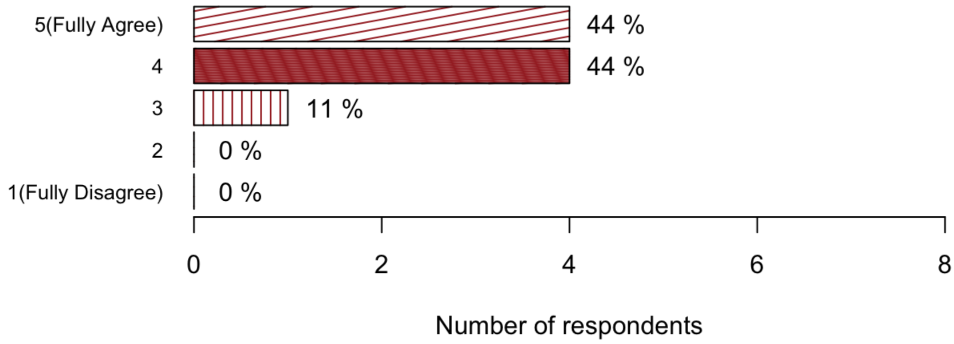



**Figure 5.28:** Responses to the Question 1 / Group 2.

#### 5.5.7.2 RQ2: Is the model useful to the military domain ?

To identify the proposed model usefulness perceived by the respondents, we presented Question 1/Group 2 (*"Real military operations are subject to changing conditions: changes in parts of the mission, changes in resource availability, changes in weather, etc. Based on this, you would agree that the representation used in the scenarios shown above can handle the different situations. It means that it shows all the situations of previous planning."*). Figure 5.28 shows the answers to this question and, despite the majority to the fully agreement, we have some agreement but with reservations.

The comments about the previous statement were collected by Question 2/Group 2 (*"Explain if you do not completely agree with the previous statement."*). The answers emphasize two aspects: the uncertainty and the team's removal due to some issue.

*"If it is necessary to consider dynamics and constant change of the operational environment, it would be important to also consider the visualization of uncertainties for the improvement of situational awareness."*

*"I just don't understand what happens if one team (T1, for example) is dropped. Will they distribute his missions to the other teams?"*

We consider both ideas related to dynamic model. However, we can represent any structural change with another instance from the meta-model. Besides that, we received an answer related to the environment:

*"I believe certain situations cannot be represented. I did not see, for example, the representation of the weather."*

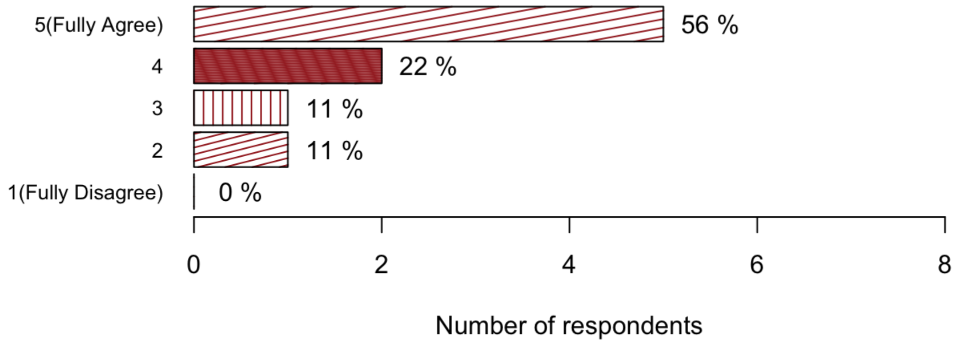



**Figure 5.29:** Responses to the Question 3 / Group 2.

However, we are considering external effects such as the weather as its impact on the onboard sensors in the entities, or on the entities themselves. The model represents such effects in the environment element, which synthetizes all external effects on the system.

In addition, Question 3/Group 2 (*"The representation employed in the scenarios presented can be used in the military domain as the only model for representing the basic information about a military operation (carried out at some level: Platoon, Company, Brigade, etc.), and the employment of the main maneuver elements."*) shows a significant level of perception about the solution's completeness. As shown in Figure 5.29, the answers to this question do not show disagreement.

Question 4/Group 2 (*"Explain if you do not completely agree with the previous statement."*) collected comments from the respondents that did not fully agree with the previous statement. We received three comments that present a limitation to small operational units, e.g., team combat or platoon. The answer below exemplifies this limitation.

*"For the General Staff on operations, I believe so - although it has a bit of a technician bias. For a Team Combat, I don't agree, given the level (of personnel) and type of the missions."*

Question 5/Group 2 (*"We can use the representation used in the scenarios presented besides the models and diagrams used in the military domain, complementing the information about the military operation."*) brought the result shown in Figure 5.30. There is only one response that disagrees with the statement.

The only comment related to the disagreement from the previous statement was collected with the Question 6/Group 2 (*"Explain if you do not completely agree with the*

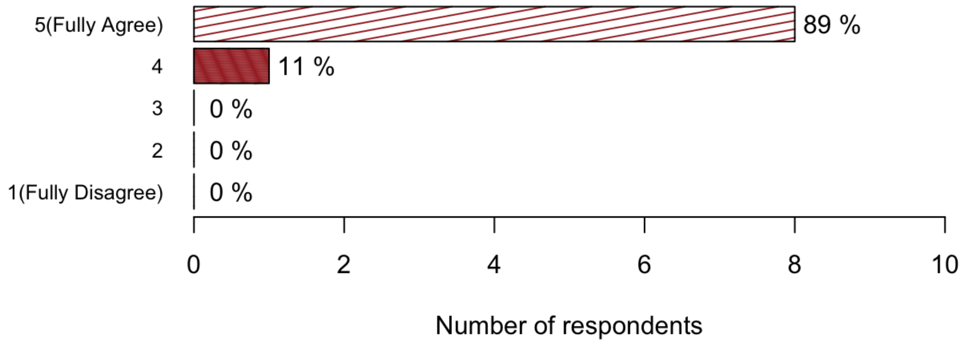



**Figure 5.30:** Responses to the Question 5 / Group 2.

*previous statement."*). The response received suggests the support of two entities' insertion in runtime. However, we consider it is related to the dynamic model. We can generate different instances according to the structure at given time. The answer below makes such idea explicit.

> *"Yes, it can, but we have to consider, besides what I previously mentioned, that a troop can receive other troops in direct support or in reinforcement. Here, the system should be prepared for this modification, because this implies a temporary change in the command structure or in the employment's purpose of the troops assigned in direct support or in reinforcement."*

Question 7/Group 2 (*"The visualization presented in the cited scenarios facilitates the understanding of the conditions and the employment of entities/resources in the mission's accomplishment."*) showed a significant agreement with the statement presented. Figure 5.31 makes these results explicit.

Related to the not fully agreement responses, Question 8/Group 2 (*"Explain if you do not completely agree with the previous statement."*) collected only one response emphasizing a question about the model's compatibility.

> *"For a Combat Team, I'm not sure it would clarify as much."*

However, we can generate an instance from the proposed model according to any organization level. Besides, to the domain's experts, it's possible to identify real scenarios similar to those generated from the proposed model. We obtained such a confirmation from Question 9/Group 2 (*"You visualize scenarios similar to the one shown, being executed*

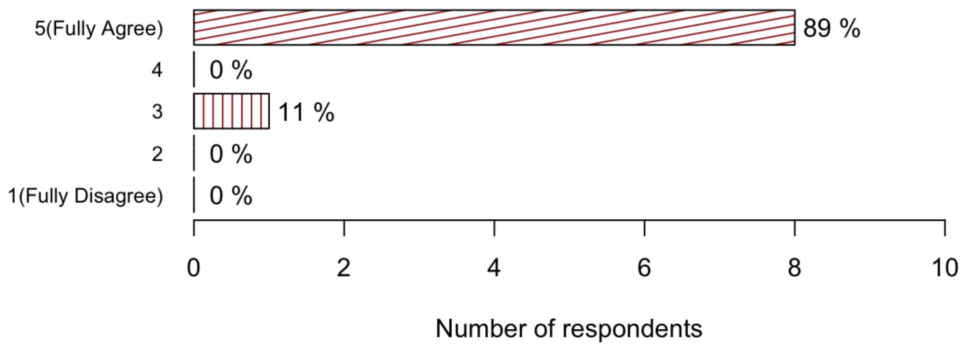



**Figure 5.31:** Responses to the Question 7 / Group 2.

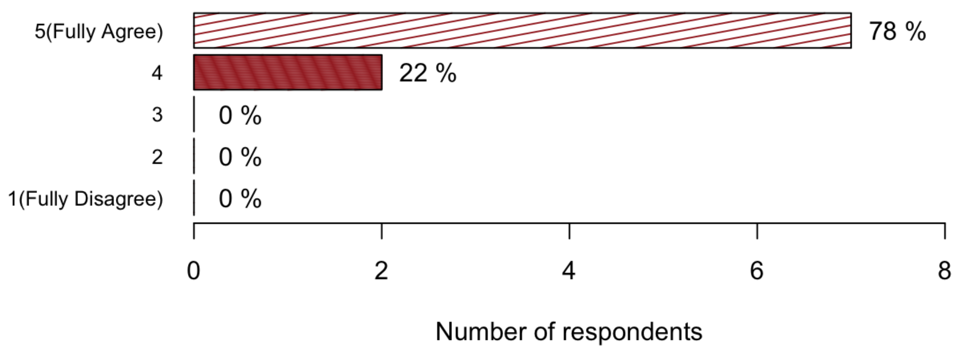



**Figure 5.32:** Responses to the Question 9 / Group 2.

**Table 5.12:** Summary conclusion related to the research questions

| RQ | Responses analysis summary |
|---|---|
| RQ1 | The proposed model exhibited ease of use with only one not fully agreement; |
| RQ2 | All respondents considered the proposed model useful, with the suggestion of adding few team members. |

*or operated by your Unit/Organization in some mission performed by it."*). Figure 5.32 shows the result for this question.

For those respondents who did not fully agree with the previous statement, Question 10/Group 2 (*"Explain if you do not completely agree with the previous statement."*) collected answers emphasizing some adaptation requirement, but the domain experts provided no example.

### 5.5.8 Survey Conclusion

According to the answers received, especially in the open questions, we identified significant agreement with the model presented, highlighting the clarity of the representation of the various elements of the C2 system, as well as the compatibility with scenarios experienced by the respondents.

Answers that suggested the insertion of some element or change in the model referred to dynamic aspects of the system, characterized by the context change. The dynamic model addresses this aspect. The focus of the proposed metamodel is a static view of the C2 system, capable of obtaining snapshots in all situations experienced by the running system. Table 5.12 shows the research question's summary following this reasoning.

In general, following all conclusions from the survey designed in Section 5.5.2, we could find evidence that suggests the ease of use and usefulness of the proposed model based on the feedback from the domain experts.

In addition, we use the Cronbach's alpha ($\alpha$) coefficient to validate the questionnaire's reliability. To perform this analysis we considered the closed-ended questions with the scale between 1 (fully disagreement) and 5 (fully agreement). Table 5.13 shows the results for each respondent. According to Section 5.3.2, we obtain $\sum_{i=1}^{k} \sigma_i^2 = 21.19$ for each question $i$ and the $\sigma_\tau^2 = 71.94$ from the survey's results of the Table 5.13. This table represents a matrix $nxk$ with $n$ respondents and $k$ questions. Applying such results in the Equation 5.4, we have:

$$\alpha = \frac{7}{7-1}\left(\frac{71.94 - 21.19}{71.94}\right) = 0.82 \tag{5.6}$$

**Table 5.13:** Responses for the closed-ended questions with scale 1-5 (in the format Section.Question) to Survey #2.

| Respondent | S1.Q1 | S1.Q4 | S1.Q6 | S1.Q8 | S1.Q10 | S1.Q12 | S2.Q14 |
|---|---|---|---|---|---|---|---|
| P1 | 5 | 5 | 4 | 5 | 5 | 5 | 5 |
| P2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| P3 | 2 | 2 | 3 | 3 | 5 | 3 | 4 |
| P4 | 4 | 5 | 4 | 4 | 4 | 5 | 4 |
| P5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| P6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| P7 | 5 | 5 | 4 | 4 | 5 | 5 | 5 |
| P8 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| P9 | 4 | 4 | 4 | 2 | 5 | 5 | 5 |

Such a result of $\alpha$ shows a reliable questionnaire with a confident response scale. It shows the consistency in the answers given to the closed-ended questions. Besides, it emphasizes this work validation according to the domain experts.

Finally, the last section of the questionnaire, with open-ended questions to collect feedback from the respondents, had only one response, as transcribed below:

*"Congratulations for the excellent and valuable research."*

This praise shows evidence of good perception about this work and its relevance.

## 5.6 Threats to Surveys' Validity

Since the surveys described in Section 5.4 and Section 5.5 have limitations in execution and data collection, we adopted the following mitigation strategies to address the corresponding threats to validity:

- **Conclusion validity**: To collect a complete feedback from the respondents and possible new scenarios, specially in the first survey (Section 5.4), we applied open-ended questions. However, such questions bring a sort of ideas which we could interpret in different ways with impact on analysis' result. To minimize such a problem, we applied a coding method Seaman [109] to the answers in order to identify the key elements and to account for them. In addition, an external researcher analyzed the responses to mitigate biases and remove conflicts.

- **Internal validity**: To address all the topics of interest of the study by submitting them to domain experts, the questionnaires are relatively long because of the support material needed to answer them. Such a requirement could have caused fatigue, prompting the respondents to give up. To reduce this risk, we used close-ended questions as much as possible.

- **Construct validity**: We use a simulated scenario that applies drones in a reconnaissance mission as a case study to base the questionnaire on. This scenario may have significant limitations that could bias respondents' answers due to lack of information or misunderstanding. To mitigate this risk, we explored the respondents' perception about the presented scenario and we asked them for real scenarios in their environment that could serve as complementary cases for this work's proposal.

- **External validity**: We had a small quantity of respondents (14 to the first survey and 9 to the second one) because of the domain experts' fear of sharing C2-related information, which naturally involves the organization's modes of operation, with people outside the Army, besides the lack of experience and knowledge on this subject. In addition, to stimulate people for responding to the survey, we keep anonymous answers. Such an aspect can create some limitation to generalize the ideas collected, and makes it impossible to analyze by type or organization of respondent. To reduce such a risk, we looked for selecting a set of respondents highly trained in the subject studied and with practical experience in order to obtain more reliable and useful responses.

# Chapter 6

# Related Work

This study combines three areas of knowledge: C2, Self-Adaptive System (SAS), and Channel System. Such a combination aims at providing C2 agility through the coordination and configuration definitions implementation. The entities' self-configuration capability modeled as DSPL, and the coordination among them to get the situation awareness, enable them to deal with context changes. Such a strategy leverages system's agility level and provides a way to model its elements. In addition, the applied model provides the required structure of messages exchanging and roles distribution, properly evaluated in agent-based simulated environment [1]. This chapter presents a literature review related to this work used as inspiration to identify the research problem and to elaborate on our proposal. Tables 6.1, 6.2, and 6.3 show a summary of the main related works' strengths, besides the identified opportunities to advance the state-of-the art used to guide the development of this work.

## 6.1 Command and Control (C2)

Alberts and Hayes [10] describes C2 as a set of procedures to get an appropriate resources application to perform a mission. This definition does not include context change principles that may eventually increment the mission execution. In addition, they present the C2 approach space with its three dimensions that define the C2 approach operated by the system. Our study applies this same concept by considering C2 as a process to achieve desired results. However, we extend such a definition by applying metrics in order to assess the C2 concepts application and to identify the capabilities of the different C2 strategies adopted at runtime. We perform such assessment considering dynamic contexts.

In more recent studies performed with North Atlantic Treaty Organization (NATO) collaboration and resulting the technical report *SAS-085*, Alberts et al. [6] analyze C2

Agility as the capability of dealing with different circumstances, and collects evidence to support some hypothesis to explain the relevance, effect and causes of agility. According to that study, such an agility results from adjustments in one or more dimensions that compose the C2 approach space.

To collect evidence, that work collects data from experiments based on software simulations and from historical results of operations performed by the army of NATO's countries. Even being complex and plenty of information, representing the state-of-the art of C2 agility, that report does not consider any cost notion. They do not measure quality attributes in the simulated environment and let opportunities, for future works, to explore the agility providing aspects in such an environment. Focused on these opportunities, we apply metrics, directly related to the definition of C2 agility, in simulated environments to collect evidence of how to deal with dynamic context.

Tran et al. [121] uses an agent-based system to perform simulations in order to validate their proposal. Their work uses adaptive networks to simulate C2 agility definition and to analyze the elements' behavior in different circumstances evaluated by the simulations performed. The scenario's complexity and the tool limitation do not allow a complete measurement of many C2 process attributes. In addition, that work does not explore the node's self-configuration, focusing on the connections between them. Based on this, our work uses the attributes from the C2 functions as metrics to analyze our proposal, and we address entities' self-configuration as a key facility to provide resilience and adaptation capability to the system in face of dynamic context.

Similar to the previous work, Leal et al. [83] uses network structure concepts to explore distinct conditions in C2 scenarios. It merges information-centric networking and software-defined networking to provide C2 agility within network structures. They address such networks as Internet of Battle Things (IoBT) and this approach supports military operations with C2 concepts application. However, it does not explore the concept of network nodes reconfiguration in dynamic context, i.e., network assets adaptation to deal with new conditions. To address an element that composes C2 agility definition, our work address this reconfiguration based on perceptions got from sensors.

Mason and Moffat [92] make an analysis of C2 processes to propose a software architecture capable of representing a headquarter applying C2 concepts during an operation. However, it does not address Quality Attributes (QAs) measurement in this process. Our work explores the agility attribute measurement as quality attributes to evaluate the proposal agility level.

Stanton et al. [116] present and compare four common models to investigate and to detail activities applied to the C2 processes. However, these models do not identify and explore explicitly the C2 agility elements, i.e., C2 approach agility and C2 maneuver

**Table 6.1:** Summary of C2 related works with their weaknesses used as the inspiration for the present study.

| Work | Strengths | Opportunity |
|---|---|---|
| Alberts and Hayes [10] | C2 definition and C2 approach space; | Do not consider dynamic context; |
| Alberts et al. [6] | C2 agility definition, historical and experiments results; | Metrics missing; |
| Tran et al. [121] | Agent based simulations with network structures; | Do not explore self-configuration and key metrics; |
| Leal et al. [83] | Networking approach to simulate C2 scenarios conditions and behaviors; | Do not explore self-configuration in network assets; |
| Mason and Moffat [92] | Software architecture as a model to represent a headquarter applying C2 concepts; | There are no Quality Attributes (QAs) analysis; |
| Stanton et al. [116] | Models to validate C2 processes; | Do not explore C2 agility aspects and its composition elements; |
| Alberts et al. [3], Anthony Alston [14] | Networking approach based on military domain analysis; | Complex structures make the models not enough flexible; |
| Alighanbari and How [11], Jose and Pratihar [73], Schwarzrock et al. [108] | Tasks allocation strategies used in simulation based on multi agent system; | Do not consider dynamic scenarios and their impacts on the allocation process; |

agility. We extend these definitions in our proposed models, aligned with the concepts that Alberts [4] introduces. Based on this, our proposal focuses on the main aspects related to circumstances changes and how to deal with such a dynamism using C2 agility.

Military domain experts in USA use the definition Network Centric Warfare (NCW) described by Alberts et al. [3], operating different positions within the C2 approach space so-called C2 approaches. This definition is like the original Network Enabled Capabilities (NEC) presented by England and described by Anthony Alston [14]. Basically, both definitions bring the same concept of awareness sharing and the network structure defines the team action capacity during the mission execution. We base our work on these definitions. However, we use the main characteristics of each C2 approach to make a correlation with network classical topologies. It provides a useful simplification of the model.

Works as presented by Alighanbari and How [11], Schwarzrock et al. [108] provide the task assignment for a fleet of UAVs, having a decentralized structure with no central command. In turn, Jose and Pratihar [73] uses a genetic algorithm to perform tasks assignment in a set of robots working on an inspection system. However, these works do not consider dynamic scenarios. Any perturbation causes undesired results or even a system stop. Based on this, our simulator uses the strategies explored in these studies to allocate tasks, but including strategies to deal with dynamic scenarios, as shown in Amorim et al. [13].

## 6.2 Self-adaptive System

According to Cheng et al. [35], self-adaptive systems (SAS) are currently widely used to deal with scenarios where there are changes during the system's execution. These systems can self-adapt according to the information collected with sensors usage, or to attend new requirements from the environment or from the users. That work applies these concepts to implement a system able to self-adapt in order to optimize the resources using optimization aiming for executing some given tasks. However, this study does not link this adaptation ability with C2 concepts. Our work explores this lack of combining C2 definitions with SAS, to provide the ability to deal with dynamic circumstances to a team of entities.

Iftikhar and Weyns [68] describe an SAS as a composition of the managed and managing system. The first one is the system adapted according to some goals. The managing system implemented by a feedback loop, i.e., MAPE-K loop, performs such an adaptation. Our proposal explores this structure in two levels. Here, we have two managed systems that represent the C2 approach and the entities' configuration, organized in layers. We adapt both with a feedback loop performance and the C2CS proposed models their behavior dealing with context changes.

As explained by Alberts and Hayes [9, 10], with some realistic scenarios' analysis, we naturally relate C2 with dynamic context and with an intrinsic uncertainty level. The possibility of any circumstance change occurrence represents this uncertainty during the mission execution. We face these uncertainties as new requirements from the circumstances. In order to deal with this dynamic context, our work applies SPL theory to model the entities in a C2 system to generate products suitable for the context. However, we use such a theory extended to deal with dynamic context. Managing uncertainty in SAS requires a complex and powerful mechanism to provide all possible circumstances analysis [19]. Based on this, we apply the Dynamic Software Product Line (DSPL) concept presented by Rosenmüller et al. [104] as a simplifier approach to model a SAS.

Bencomo et al. [25] describes a DSPL as a combination of structure, behavior, and goals. We use such a description in our study to model a set of entities performing a mission. In our work, we leverage the DSPL self-configuration capability with roles assignment. It provides the required entities' configuration to deal with context changes.

**Table 6.2:** Summary of SAS related works with their weaknesses used as the inspiration for the present study.

| Work | Strengths | Opportunity |
|---|---|---|
| Cheng et al. [35] | SAS principles to implement a system to perform tasks according to the scenarios; | Do not correlate SAS and C2 definitions; |
| Rosenmüller et al. [104] | DSPL as an approach to model a SAS; | Do not correlate DSPL and C2 definitions; |
| Arcaini et al. [19] | Uncertainty managing impacts in SAS; | Do not present an approach to deal with uncertainty. Probabilistic methods as an approach; |
| Bencomo et al. [25] | DSPL structure with self-configuration and roles assignment; | Do not use the roles to coordinate multiple DSPLs; |
| Rosenmüller et al. [104] | Dynamic Software Product Line (DSPL) as isolated and independent elements; | Do not explore collaboration between DSPLs.; |
| Baresi and Quinton [22, 23], Bécan et al. [24], Bencomo et al. [26], da Silva et al. [39], Lochau et al. [88], Pessoa et al. [99], Rosenmüller and Siegmund [103], Rosenmüller et al. [104], Shariﬂoo et al. [110], Trollmann et al. [123] | There is no interaction or information exchanging between DPSLs. This capability is well known in Software Product Line (SPL) with static scenarios; | Do not provide information sharing to get awareness and to obtaining the best configuration and coordination to deal with the tasks; |
| Lienhardt et al. [86], Rosenmüller and Siegmund [103] | SPL composition therough Multi Product Line (MPL); | There is no mechanismo to deal with dynamic context; |

In addition, we model such entities as Feature Model (FM) with some attributes and constraints inclusion in order to create rules to be satisfied when we combine features to generate products. These constraints can regulate the interdependent relationship between entities. Bécan et al. [24] present the paradigm that defines the insertion of attributes into a FM and we extend such a proposal to create the conditions to satisfy the C2 agility definition through the configuration and coordination of the modeled entities.

Besides, current studies about DSPL [22, 23, 26, 39, 88, 99, 103, 104, 110, 123] do not explore DSPL grouping and coordination. They do not explore the communication between DSPLs that affects on the collaboration and final global configuration, i.e., the set of each member's configuration. In these studies, the DSPLs are independent elements and there are no side effects between different DSPLs as presented by Rosenmüller et al. [104]. Our study extends the reasoning used by MPL theory [87], where a group of SPL shares a mutual contribution among its elements. The proposed extension uses DSPL to deal with dynamic context.

As shown by Rosenmüller and Siegmund [103] and Lienhardt et al. [86], a set of SPLs composing a Multi Product Line (MPL) performs an organization to share information in order to provide a better configuration according to the general information got from the surrounds, and the effects caused by the SPLs to each other. Our study applies this theory to DSPLs handling the possibility of interaction and interdependence between distinct elements in a complex structure. Based on this, we model a C2 System as a set of interdependent elements that perform a self-configuration to keep the mission accomplishment. This approach models the realistic scenarios as software components getting a suitable awareness to adjust themselves according to the context.

## 6.3 Modeling

According to Rutten et al. [105], we can use an automaton to model an SAS behavior. Besides, we can connect multiple automata to form a network representing a set of components. Baier and Katoen [20] show that we can write such automata as a Transition System (TS) with a finite number of states represented by Labeled Kripke Structures. Based on this, Sousa et al. [115] use TS theory to model Dynamic Software Product Line (DSPL). However, modeling directly with this structure can become unfeasible because of a huge number of states that represent the valid configurations of the DSPL.

To address the TS limitation, Baier and Katoen [20] present the program graph abstraction. We leverage this approach to model the behavior of C2 system entities, initially modeled as DSPLs, using the channels as connectors between different processes at distinct levels of a C2 structure. In addition, we explore the structure composed of multiple parallel program graphs, so-called channel system, to model the entities' behavior. We extend such a structure with the type-parameterized constraint to achieve C2 agility results. This strategy gives us the capability to generate, in runtime, a new channel system according to the context changes, providing the channel system with the ability to adapt itself according to the elements inserted or removed from the system in runtime.

Inspired by Dunin-Kęplicz and Verbrugge [47], Dunin-Kęplicz et al. [48], Fernandes et al. [49], Grant et al. [61], Jennings and Bussmann [71], Kraus and Grosz [82], Schwarzrock et al. [108], where real agents, e.g., Unmanned Aerial Vehicle (UAV), are simulated in Multi-Agent System (MAS) with a suitable tasks allocation process, we extend such modeling with the dynamic context principles. Some studies, as they presented by Jose and Pratihar [73], use a centralized coordination of the agents. However, we adopt a decentralized coordination, increasing the agent's autonomy, in order to avoid a single point of failure and to provide more resilience in realistic scenarios for the military domain. In such scenarios, we use a slight adaptation of Alighanbari and How [11], Schwarzrock

**Table 6.3:** Summary of modeling related works with their weaknesses used as the inspiration for the present study.

| Work | Strengths | Opportunity |
|---|---|---|
| Baier and Katoen [20] | Program graph (PG) and Channel System (CS) definitions and elements; | The Channel Systems are closed systems, i.e., there is no data exchange between distincts CS; |
| Sousa et al. [115] | Transition System modeling a DSPL; | Huge number of states can be unfeasible; |
| Dunin-Kęplicz and Verbrugge [47], Dunin-Kęplicz et al. [48], Fernandes et al. [49], Grant et al. [61], Jennings and Bussmann [71], Jose and Pratihar [73], Kraus and Grosz [82], Schwarzrock et al. [108] | Multi-Agent System (MAS) simulation with centralized point of coordination; | Single point of failure; |
| Alighanbari and How [11], Schwarzrock et al. [108] | Tasks allocation based on swarm algorithm; | Not optimized for dynamic scenarios; |
| Abar et al. [1] | Comparison between agent-based systems; | C2 principles implementing using a suitable agent-based system; |
| Schwarzrock et al. [108] | C2 scenario simulation using an agent-based system; | No C2 agility approach; |
| Gerasimou et al. [57] | Architecture that uses a controller based on MAPE-K loop; | Single loop limits multiple roles within the same agent; No Quality Attributes (QAs) exploration |
| Santos et al. [106] | Model-Driven Development approach to support the modeling of an agent-based simulation system; | Do not use Self Adaptive System (SAS) definition to improve the adaptation capability; |

et al. [108], where agents perform the task assignment for a fleet of UAVs with no central command.

Empirical studies with the use of simulation are widely explored and relevant to many domains, e.g., military [83], and environmental monitoring [124]. The simulation can create several distinct circumstances to evaluate a solution or product, or for personnel training, reducing resources requirement and cost for realistic circumstances creation. In line with this philosophy and based on the comparison presented by Abar et al. [1], we choose RePast Simphony [97] as the simulation environment to perform tests and to collect evidence related to our proposal. The usability, the knowledge base, and the underlying Java technology applied, reduces the learning curve of the tool. We base our proposal on strategies of tasks allocation presented by Schwarzrock et al. [108] in a scenario with UAVs performing a reconnaissance mission. However, we extend those works to address

C2 Agility in the agent-based simulation to include the maneuver agility concept combined with team members' reconfiguration.

Similar to the previous works that perform simulations, Gerasimou et al. [57] presents an architecture that uses a controller defined by a MAPE-K loop to handle and to process the system information. Based on that strategy, the collected information provides the awareness required to perform the system self adaptation. We extend this solution to include multi feedback loops, organized in layers, that define our system roles. These loops characterize the agent-based simulation proposed architecture under dynamic context. The collected results show evidence of effects in Quality Attributes (QAs) measurements caused by C2 approach change, i.e., entities' configuration and coordination adjustments.

To model the members of our C2 scenario, we base on the proposal presented by Santos et al. [106] that uses a Model-Driven Development approach to support the modeling of an agent-based simulation system. However, we extend such a solution by making an approach using Self Adaptive System (SAS) definitions. We use this extension to collect evidence of any important detail in C2 domain operation and to validate the proposal in the present work. Wohlin et al. [127] describes this validation process, emphasizing the risk of some technology limitation hiding important aspects from the realistic domain.

# Chapter 7

# Conclusions

Besides the broad area of application, e.g., Civil Defense, Military operations, financial control and disaster relief operations, this work focused on C2 application applied to the military domain. We base this choice on the possibility of having access to domain experts in this area, as well as the very dynamic characteristics of the application scenarios in this domain. To mobilize entities' efforts with the proper resources to achieve a goal, the C2 system is always dealing with dynamic context. Such a condition defines a natural possibility of changes in one or more variables and elements of the scenario.

Changes in the context require self adaptation of the C2 system to deal with new circumstances in runtime. This context is composed of the entities, the mission, and the environment. Limited analysis performed over C2 theory in dynamic scenarios, so-called C2 agility, motivate the exploration of proposals aiming to collect evidence of how to get agility to timely deal with context changes.

To show how we may handle C2 agility, we propose two computational models of a C2 system, i.e., the structure that involves everything required to the mission accomplishment. These models, one static and one dynamic, are composed of entities collaborating towards performing tasks. The dynamic model defines how entities' reconfiguration and coordination may help to handle context changes occurring in the mission, in the environment, or in the entities themselves. A metamodel with all main C2 system elements defines the static model from which it is possible to instantiate different structures able to deal with different contexts.

We formalize the dynamic model as a typed-parameterized extension of a channel system, which defines the roles and responsibilities handled by the entities composing a C2 system, where each entity is modeled as a dynamic software product line. Besides, the conceived metamodel brings complementary information to the dynamic model in order to deal with different contexts, i.e., to provide C2 agility, and to support complex structures as teams of teams. The combination of these two models provides all information required to provide agility for a C2 system.

Considering the complexity involved with C2, the mathematical models and simulation tools cannot treat all variables and constraints existing in a realistic scenario, specially the related to the environment. Simpler models can hide important effects of some variables and be incompatible with realistic scenarios. To mitigate this threat, our proposed models represent the dynamic and static views, besides they represent the interaction and dependences of the entities' roles.

Overall, we apply the action research to develop our proposal, i.e., systematic enquiry, to explore weaknesses and not answered questions related to C2 agility and its challenges. We use Software Product Line (SPL) concept to model the entities with self-adaptive capability for dealing with changes in circumstances. It provides a required simplification of the original problem and context. This modeling treats the realistic entities as software components with a set of features that can be activated according to the mission requirements, self status, and environment conditions. Basically, these features represent the resources and capacities onboard, e.g., sensors and engine power, which are represented and managed with a Feature Model (FM).

In addition, this work uses Network Enabled Capabilities (NEC) principles aligned with the C2 approaches operated by the entities, to provide a coordination of these entities through a combination of all resources available in order to improve the results got during the mission performance. This structure provides the suitable coordination condition with a network topology characterized by a topology that represents the key characteristics of the related C2 approach. Such a simplification looks for become its simulation feasible.

Finally, we perform simulations using an agent-based system representing a team of UAVs performing a reconnaissance mission. Such an implementation follows the models proposed by the present work, with all their characteristics and rules strictly followed. Based on this result, this work identifies the C2 agility providing the system's capability of dealing with context changes.

In addition, we insert perturbations in the system to measure its capability to continue executing tasks looking for optimized results. To improve such an analysis, our method submits the simulated scenarios and their results to domain experts analysis to collect their perception about the proposed models. We perform this evaluation through two surveys to measure the usability and utility of the proposed models in the military domain. Thus, the obtained results and feedback confirm the subject relevance and the proposal's compatibility with realistic scenarios operated by the military domain, besides it shows a correlation between the surveys and simulation results.

In summary, the models proposed by this work and the assessment performed aimed to provide a tool able to simplify a real C2 system, representing all major elements that compose such a system and their behaviors based on roles distributed according to the C2 approach operated. Finally, we organize this structure to obtain the context awareness and suitable engaging for expected mission's results, providing C2 agility to the system's operation.

## 7.1 Limitations of this Work

Despite the found evidence, this work presents limitations that are related to the inherent complexity within C2 scenarios. Functions to provide task allocation and to choose the suitable C2 approach are some of these complex aspects. These procedures can significantly affect our simulation results got by the team. Our simulator performs a non-optimized task allocation and presents limited resilience in this process. Such a process is not the present work focus and we use a solution previously tested. It can replace it with low impact on the proposed structure. In addition, we sequentially perform the C2 approach selection according to its communications structure. Such a choice does not consider a previous analysis of the context or even a knowledge database. Here, the several attempts of the C2 approach, looking for the best one, have a cost of operating time and onboard resources, affecting the operation results.

Empirical studies present intrinsic restrictions on the method related to the variations got in the observations, which can lead the analysis to a bias. Based on this, the simulations have limitations if compared with realistic scenarios and the present study applies the C2 only in the computational environment. This application creates restrictions to the extension of the proposed model to other domains.

Dynamism is a C2 intrinsic characteristic, and it increases the scenario's complexity. During the mission accomplishment, following a selected C2 approach, various factors can change and cause side effects, such as mission change, loss of entities, and impact of the environment on luminosity reduction. Such a dynamism insertion in the simulation is always challenging, and it is not entirely compatible with the real world and the simplification is mandatory. We can not simulate all changes that can occur within the context, which makes simulation of C2 scenarios limited and it may generate biases and misunderstandings when applied to realistic scenarios. It is not possible to represent all the environment complexity existing in these scenarios.

In this work, we use UAV's operation scenarios to simulate the events that provide dynamism to the context. Such a specificity may create some compatibility issue with other realistic scenarios. To extend the proposed model to other domains, making it compatible with other contexts, it is necessary to include new events and attributes that may affect the model's behavior and may require significant adjustments, especially in the parameterized channel system.

The proposed dynamic model, based on the channel system theory, has a limitation on the capability of modeling multilevel teams, i.e., teams of teams. However, the present study follows the principles presented by Baier and Katoen [20] that define the channel system (CS) as a closed structure, i.e., there is no data exchanging with an external CS or environment. To represent such a complex structure, this model must include a communication structure to provide data exchange between distinct Channel System (CS) that represents distinct levels in the team.

In realistic scenarios, other variables beyond the entity, environment, and mission can change during C2 system operation. These variables and constraints have a high level of interdependence and uncertainty. Simulating all the effects that can occur with all combinations is practically unfeasible. To simplify this context aiming to get a mathematical model, our work defines a set of assumptions to reduce the problem scope (cf. Section 3.4).

Finally, the C2 context is a broad knowledge area and it can be applied in many circumstances and scenarios, e.g., Civil Defense, Military operations, financial control and disaster relief operations. This large number of possibilities has specific characteristics of each scenario that can become unfeasible as a *"one size fits all"* solution. Based on this limitation, we limit this work to C2 applied to military context. However, we highlight the organization maturity requirement to explore C2 agility concepts and the related concepts properly.

## 7.2 Future Work

The application of C2 concept is based on a set of resources employed to solve a group of activities or tasks, so-called mission. However, uncertainty is always present in this scenario because of changes in the elements employed, in the mission, or in the environment where operation runs. We mitigate these uncertainties in the planning phase, where we try to foresee as many situations as possible. This set of planned situations defines the endeavor space (cf. Section 2.1.2). During the mission execution, the entities handle different situations within the endeavor space through a configuration and coordination defined by the C2 approach operated. Figure 7.1 shows the sequence of situations S1, S2 and S3, that require distinct configurations with their respective coordination techniques between the elements.

The proposed models aim to represent any context scenario within the endeavor space by applying C2 concepts. Such an approach allows us to perform a previous analysis and

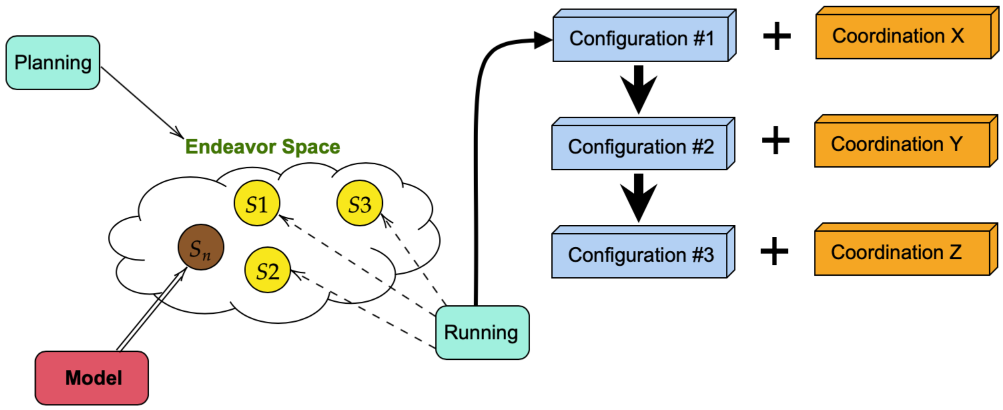



**Figure 7.1:** Different situations within the endeavor space faced by configuration and coordination defined by a C2 approaches. In addition, the proposed model able to represent any scenario.

to predict results, defining the quality limits to be met in the mission accomplishment. We can perform these analyzes using simulation that follows the behavioral and properties proposed by the models presented in this study. These models represent these elements, their behaviors and roles, as well as the actions to be taken in case of context changes. Even with the empirical analysis based on simulation to validate the proposed models and to collect evidence about their usability and utility, we do not have a mathematical C2 agility definition to perform a deeper, formal analysis and validation. An initial exploration shows that the Quantified Propositional Temporal Logic (QPTL) [53, 54] would be a suitable choice to make such a formalization. It provides all conditions to perform an automatic verification with model checkers or reasoning with proof assistants, e.g., Prototype Verification System (PVS).

Our proposal does not explore the temporal aspect inherent to C2 system execution. It is quite important because of the steps sequence defined by the roles' acrfullCS to provide C2 agility. The temporal issue is quite important to be explored, including its influence on the C2 Agility definition. This idea establishes a relation among the C2 elements, i.e., members, mission, and environment. Tasks allocation and execution depend on the time variable with the using of *eventually* term. Based on this, we can extend the C2 Agility as below:

| |
|---|
| For all tasks $t$ of a mission, it is possible to eventually find a C2 approach $\omega$, such that a team $E$, i.e., set of members, becomes constrained to operate under $\omega$, in a way that $t$ is eventually allocated to a member $e \in E$, and $e$ eventually adopts an internal configuration $c$ among its valid configurations, such that this configuration makes $e$ capable of dealing with $t$. |

We encourage other studies about C2 that explore different domains, such as financial market, civil defense, network incident handling, and vaccination campaign. In such scenarios, the proposed metamodel, i.e., C2MM, may represent all structure's elements, and the C2CS may model their behavior and procedures adopted during the operation. Indeed, many domains have applied C2 concepts to their operation. A current domain example is related to COVID-19-pandemic [56, 111, 114, 129]. From health services and agencies to the coordination of health and vaccination campaign, require the coordination of several elements to perform dozens of tasks. Figure 7.2 shows a simplified representation of some elements in such a scenario, where there is information exchange, decision making, and agent activation to assist the patients' care. Current studies show that shared information and decisions in hospitals and related environments are decisive elements to obtain positive results in saving life's mission. These scenarios are plenty of distinct elements with responsibilities and activities well defined.

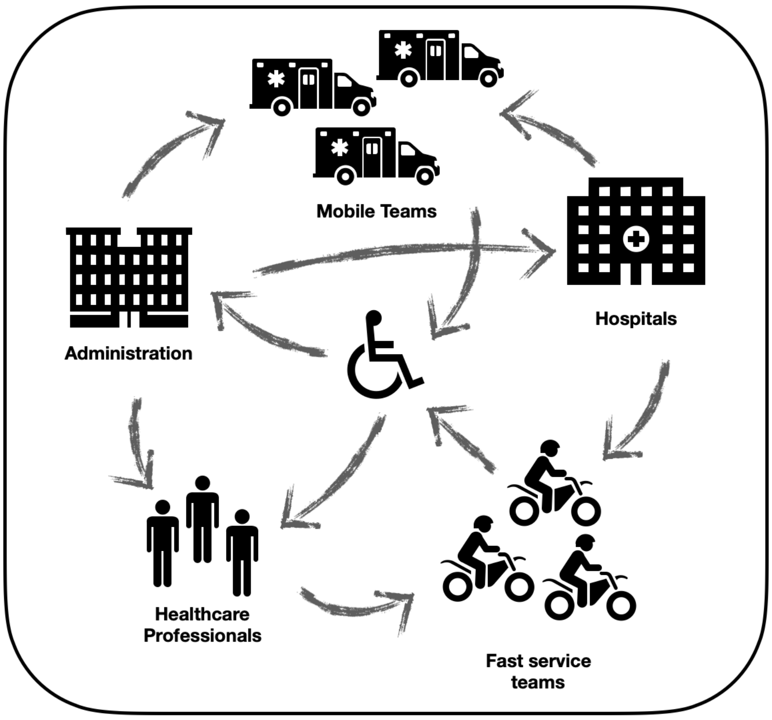



**Figure 7.2:** Elements that compose the simplified structure of patient care in a pandemic scenario..

Since 60th, the UAV usage in military operations, including combat missions, have an increasing [81]. For such an application, the aircraft actively takes part in a mission to attack a target, e.g., an aircraft, a building, or enemies. Future works may explore the proposed models' employment in these scenarios. However, it is important to validate if the mission type affects the model's usability. Besides, considering autonomous systems, the ethical aspects of this employment must be considered.

However, for the models presented in this work to support complex scenarios like the one presented above, in which the heterogeneity of the elements and their great stratification, there is the need to provide communication between different levels, enabling the modeling of complex coordinated actions. In this way, incorporating entities grouped into layered teams can make the structure of the C2 System more complex. Based on this, the proposed dynamic model, that uses the channel system theory, does not consider a chained structure, i.e., a team of teams. In such a case, the roles defined by the proposed C2CS need to exchange data with roles at another level in the structure. As the CS is a closed system (cf. Section 2.3), the proposed model does not support multiple layers. To extend this definition, Baier and Katoen [20] presents the concept of Open Channel System (OCS). This concept seeks to allow the CS to communicate with the surrounding environment.

Thus, OCS concept seeks to allow the CS to communicate with the surrounding environment, receiving information from the outside through a synchronous channel. For the channel c present in the OCS, whose domain is $dom(c)$ and $cap(c) = 0$, it can receive a variable $v \in dom(c)$. At this moment, all PGs listening to this channel perform the change of location, receiving data from an external OCS. The sending of $v$ occurs in a non-deterministic way by the environment that surrounds the OCS. To use the Open Channel System (OCS) concept within our proposal, we must extend the following aspects:

- The environment is another CS that surrounds a set of CS, as sub teams, and performs information exchanging;
- The domain of the channel(s) could contain composite types (Ex.: tasks, C2 Approach, time, etc);
- The channel(s) that would receive the data exchanged in CS need to be asynchronous in order to allow a CS to receive information coming from multiple CS without crashing its execution; and
- The channel to this communication needs to be two-way: from the environment to the CS and vice versa.

Originally working with synchronous channels, we must extend the OCS with receiver asynchronous channels in order to allow a CS to receive information coming from multiple CS without stopping its execution and exploring parallelism. Based on this, we can explore such resources and be able to model complex scenarios. In addition, the CS can receive more actions that affect the system's variables to represent more context changes possibilities, e.g., *addMember* to include new members in the team.

A C2 system structure requires a mechanism to provide data exchanging among the entities. We represented this communication in the C2 space with one of its dimension so-called Distribution of Information (cf. Section 2.1.3). Based on this, we can explore the usage of other modeling strategies instead of CS, e.g., Bulk Synchronous Parallel (BSP) [120], to represent the communication between the entities' roles. Besides, it is possible to explore all aspects that affect the information exchanged comprehension in future works. The understanding, knowledge, language, code and data are components that give meaning to the information transmitted and/or received, and involve many variables not explored by the present study.

As introduced by our study, the combination of C2 concepts and software engineering is innovative and opens great opportunities to go further in terms of study areas. As C2 shows a high capability of application in many areas, this approach brings the chance of exploring systems in many human activities areas. Psychology specialists can explore leadership and behavior aspects from C2 functions, while we can use mathematics elements to represent C2 components and to prove them. In addition, the approach proposed by us can motivate a broad analysis of Brazilian Army doctrine, increasing the C2 agility concept applied in realistic scenarios.

The survey applied to validate all concepts, and the proposed model's compatibility receives a low quantity of valid feedback. This result is mainly because the respondents attributed confidentiality to the subject of the questionnaires. To extend its application, the surveys may be applied in officer training schools of Brazilian Army, e.g., Captains Career School (ESAO) and Command and General Staff College (ECEME).

Finally, future work can explore the use of improved solutions applied to the function of maneuvering search and task allocation. One possibility is the use of machine learning to process data collected from sensors in order to select the most suitable configuration and coordination structure. We can plug this strategy into our proposed model to provide better results related to the mission execution in dynamic scenarios. We can do such an extension by exploring the knowledge step of a MAPE-K loop used to implement our MDSPL (cf. Section 4.1). In addition, other metrics, in particular those related to agility, e.g., robustness and innovation, can be inserted to make a deeper agility analysis.

# References


[1] Sameera Abar, Georgios K Theodoropoulos, Pierre Lemarinier, and Gregory M P O'Hare. Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, 2017. ISSN 1574-0137. doi: https://doi.org/10.1016/j.cosrev.2017.03.001. 14, 59, 86, 128, 134

[2] Abdulla Al-Kaff, David Martín, Fernando García, Arturo de la Escalera, and José María Armingol. Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*, 92:447–463, 2018. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2017.09.033. URL `http://www.sciencedirect.com/science/article/pii/S0957417417306395`. 29

[3] D S Alberts, J Garstka, and F P Stein. *Network Centric Warfare: Developing and Leveraging Information Superiority*. CCRP publication series. Department of Defense. Center for Advanced Concepts and Technology (ACT), 1999. ISBN 9781579060190. URL `http://dodccrp.org/files/Alberts{_}NCW.pdf`. 11, 15, 57, 130

[4] David Alberts. Agility, Focus, and Convergence: The Future of Command and Control. *The International C2 Journal*, 1:33, 2007. 130

[5] David Alberts, Reiner K Huber, and James Moffat. *NATO NEC C2 maturity model*. CCRP, 2010. URL `http://dodccrp.org/files/N2C2M2_web_optimized.pdf`. 3, 4, 15, 16, 17, 19, 59, 84, 88

[6] David Alberts, Agostino Bruzzone, Claudia Baisini, Kevin Chan, Douglas Ball, Margaret Daly-Hayes, Július Baráth, Paul Davis, Micheline Bélanger, Lorraine Dodd, François Bernier, Philip Farrell, Fernando Freire, Herman Joglar, Robert Gregg, Björn Johansson, Richard Hayes, Tor Langsæter, Michael Henshaw, and Marco Sebastiani. Command and Control (C2) Agility. Technical report, North Atlantic Treaty Organization - Research and Technology Organization, 2014. 2, 3, 4, 10, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 56, 58, 59, 73, 82, 86, 128, 130

[7] David S Alberts. *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*. Command and Control Research Program, Washington, DC, USA, 2011. ISBN 978-1-893723-23-8. URL `https://apps.dtic.mil/dtic/tr/fulltext/u2/a631225.pdf`. 3, 4, 12, 18, 80

[8] David S Alberts. Agility, Automation, and Autonomy. Technical report, Institute for Defense Analyses, 2017. URL `http://www.jstor.org/stable/resrep22727`. 3, 4, 19

[9] David S Alberts and Richard E Hayes. *Power to the Edge: Command, Control in the Information Age.* Command and Control Research Program (CCRP) Publication Series, 2003. ISBN 1-893723-13-5. URL `http://www.dodccrp.org/files/Alberts_Power.pdf`. 1, 2, 10, 11, 13, 17, 18, 19, 20, 29, 58, 131

[10] David S Alberts and Richard E Hayes. *Understanding Command and Control (Future of Command and Control).* DoD Command and Control Research, 2006. ISBN 1893723178,9781893723177. URL `http://www.dodccrp.org/files/Alberts_UC2.pdf`. 1, 3, 10, 11, 12, 13, 14, 15, 17, 19, 30, 57, 58, 92, 111, 128, 130, 131

[11] Mehdi Alighanbari and Jonathan P How. Decentralized task assignment for unmanned aerial vehicles. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 5668–5673. IEEE, 2005. 130, 133, 134

[12] Vander Alves, Daniel Schneider, Martin Becker, Nelly Bencomo, and Paul Grace. Comparitive study of variability management in software product lines and runtime adaptable systems. In *Third International Workshop on Variability Modelling of Software-Intensive Systems, Seville, Spain, January 28-30, 2009. Proceedings*, pages 9–17, 2009. URL `http://www.vamos-workshop.net/proceedings/VaMoS_2009_Proceedings.pdf`. 62

[13] Junier Caminha Amorim, Vander Alves, and Edison Pignaton de Freitas. Assessing a swarm-GAP based solution for the task allocation problem in dynamic scenarios. *Expert Systems with Applications*, 152:113437, 2020. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2020.113437. 6, 29, 30, 50, 52, 84, 87, 92, 131

[14] Anthony Alston. Network Enabled Capability – the concept. *Journal of Defence Science*, 8(3):108–116, 2003. 130

[15] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. *Feature-Oriented Software Product Lines, Concepts and Implementation.* Springer, 2013. 24

[16] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. *Feature-Oriented Software Product Lines, Concepts and Implementation.* Springer, 2013. 22, 23

[17] P Arcaini, E Riccobene, and P Scandurra. Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 13–23, may 2015. doi: 10.1109/SEAMS.2015.10. 63

[18] P Arcaini, E Riccobene, and P Scandurra. Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 13–23, may 2015. doi: 10.1109/SEAMS.2015.10. 60

[19] Paolo Arcaini, Elvinia Riccobene, and Patrizia Scandurra. Formal design and verification of self-adaptive systems with decentralized control. *ACM Transactions on*

*Autonomous and Adaptive Systems*, 2017. ISSN 15564703. doi: 10.1145/3019598. 131, 132

[20] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking (Representation and Mind Series)*. Massachusetts Institute of Technology, Cambridge, MA, USA, 2008. ISBN 026202649X, 9780262026499. 26, 71, 133, 134, 139, 142

[21] Dariusz Barbucha. Search modes for the cooperative multi-agent system solving the vehicle routing problem. *Neurocomputing*, 88:13–23, 2012. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2011.07.032. URL `http://www.sciencedirect.com/science/article/pii/S0925231212000124`. 30

[22] L Baresi and C Quinton. Dynamically Evolving the Structural Variability of Dynamic Software Product Lines. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 57–63, may 2015. doi: 10.1109/SEAMS.2015.24. 25, 132

[23] L Baresi and C Quinton. Dynamically Evolving the Structural Variability of Dynamic Software Product Lines. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 57–63, may 2015. doi: 10.1109/SEAMS.2015.24. 132

[24] Guillaume Bécan, Razieh Behjati, Arnaud Gotlieb, and Mathieu Acher. Synthesis of attributed feature models from product descriptions. In *Proceedings of the 19th International Conference on Software Product Line*, SPLC '15, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336130. doi: 10.1145/2791060.2791068. URL `https://doi.org/10.1145/2791060.2791068`. 25, 132

[25] Nelly Bencomo, Jaejoon Lee, and Svein Hallsteinsen. How dynamic is your Dynamic Software Product Line? In *Proc. 14th SPLC*, pages 61–68, 2010. 131, 132

[26] Nelly Bencomo, Svein O Hallsteinsen, and Eduardo Santana de Almeida. A View of the Dynamic Software Product Line Landscape. *{IEEE} Computer*, 45(10):36–41, 2012. doi: 10.1109/MC.2012.292. URL `https://doi.org/10.1109/MC.2012.292`. 62, 63, 132

[27] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., 1999. 30, 31, 32

[28] Lauren Borja and M V Ramana. Command and Control of India's Nuclear Arsenal. *Journal for Peace and Nuclear Disarmament*, pages 1–20, 2020. doi: 10.1080/25751654.2020.1760021. 1, 4

[29] Vincent Boulanin and Maaike Verbruggen. Availability and military use of uavs. In *SIPRI Literature Review for the Policy and Operations Evaluations Department of the Dutch Ministry of Foreign Affairs*, pages 121–132. Stockholm International Peace Research Institute, 2017. URL `https://www.iob-evaluatie.nl/binaries/iob-evaluatie/documenten/rapporten/2017/08/01/sipri-literature-review-for-iob/SIPRI+Literature+Review+for+IOB.pdf`. 84, 86

[30] Brazilian Federal Government. Oil spills on the brazilian coast controlling, cleaning and preserving, 2019. URL https://www.brasil.gov.br/manchanolitoral/english-version/. 1

[31] Peter C. Bruce. *Introductory Statistics and Analytics: A Resampling Perspective*. John Wiley & Sons, Inc., 1st edition edition, 2014. ISBN 1118881354. 85

[32] Y Brun, Giovanna Di Marzo Serugendo, Cristina Gacek, Holger Giese, Holger Kienle, Marin Litoiu, Hausi Müller, P Pezze, and Mary Shaw. *Engineering self-adaptive systems through feedback loops*, pages 48–70. Springer, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-02161-9_3. 25

[33] J Carver, Natalia Juristo, Forrest Shull, and Sira Vegas. The Role of Replications in Empirical Software Engineering. *Empirical Software Engineering*, 13(2):211–218, 2008. URL http://dx.doi.org/10.1007/s10664-008-9060-1. 30, 35, 36

[34] Centro de Comunicação Social do Exército. Comando Conjunto de Prevenção e Combate ao Terrorismo. *Revista Verde-Oliva*, pages 30–34, 2016. URL https://www.calameo.com/exercito-brasileiro/read/00123820665efb7b8121c. 103

[35] Betty H C Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, Jeff Magee, Jesper Andersson, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Giovanna Di Marzo Serugendo, Schahram Dustdar, Anthony Finkelstein, Cristina Gacek, Kurt Geihs, Vincenzo Grassi, Gabor Karsai, Holger M Kienle, Jeff Kramer, Marin Litoiu, Sam Malek, Raffaela Mirandola, Hausi A Müller, Sooyong Park, Mary Shaw, Matthias Tichy, Massimo Tivoli, Danny Weyns, and Jon Whittle. Software Engineering for Self-Adaptive Systems: A Research Roadmap. In Betty H C Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee, editors, *Software Engineering for Self-Adaptive Systems*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-02161-9. doi: 10.1007/978-3-642-02161-9_1. URL https://doi.org/10.1007/978-3-642-02161-9{_}1. 25, 131, 132

[36] Betty H C Cheng, Holger Giese, Paola Inverardi, Jeff Magee, and Rogério de Lemos, editors. *Software Engineering for Self-Adaptive Systems*, volume 5525 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2009. 25

[37] Cochran W.G. *Planning and analysis of observational studies*. John Wiley & Sons, Inc., 1983. ISBN 0471887196,9780471887195. doi: 10.1002/9780470316542. 85, 89

[38] Lee Cronbach. Coefficient Alpha and Internal Structure of Tests. *Psychometrika*, 16:297–334, 1951. doi: 10.1007/BF02310555. 96

[39] J R F da Silva, F A P da Silva, L M do Nascimento, D A O Martins, and V C Garcia. The dynamic aspects of product derivation in DSPL: A systematic literature review. In *2013 IEEE 14th International Conference on Information Reuse Integration (IRI)*, pages 466–473, aug 2013. doi: 10.1109/IRI.2013.6642507. 132

[40] David L Streiner. Being Inconsistent About Consistency: When Coefficient Alpha Does and Doesn't Matter. *Journal of personality assessment*, 80:217–222, 2003. doi: 10.1207/S15327752JPA8003_01. 96

[41] Fred D Davis. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3):319–340, 1989. ISSN 02767783. URL `http://www.jstor.org/stable/249008`. 116

[42] Robert Davison, Maris Martinsons, and Ned Kock. Principles of canonical action research. *Inf. Syst. J.*, 14:65–, 2004. doi: 10.1111/j.1365-2575.2004.00162.x. 5

[43] Robert Davison, Maris G. Martinsons, and Ned Kock. Principles of canonical action research. *Information Systems Journal*, 14(1):65–86, 2004. doi: 10.1111/j.1365-2575.2004.00162.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2575.2004.00162.x`. 35

[44] Josh Dehlinger and Robyn R. Lutz. Gaia-pl: A product line engineering approach for efficiently designing multiagent systems. *ACM Trans. Softw. Eng. Methodol.*, 20 (4):17:1–17:27, September 2011. ISSN 1049-331X. doi: 10.1145/2000799.2000803. URL `http://doi.acm.org/10.1145/2000799.2000803`. 64

[45] Ralph Depke, Reiko Heckel, and Jochen Küster. Roles in Agent-Oriented Modeling. *International Journal of Software Engineering and Knowledge Engineering*, 11:281–302, 2001. doi: 10.1142/S0218194001000529. 86

[46] Centro de Comunicação Social do Exército. Brasil e Haiti: Uma lição de Missão de Paz. *Revista Verde-Oliva*, pages 6–8, 2018. 103

[47] Barbara Dunin-Kęplicz and Rineke Verbrugge. *Teamwork in Multi-Agent Systems, A Formal Approach*. Wiley, 2010. 133, 134

[48] Barbara Dunin-Kęplicz, Rineke Verbrugge, and Michał Ślizak. TeamLog in action: A case study in teamwork. *Computer Science and Information Systems*, 7(3):570–595, 2010. ISSN 18200214. doi: 10.2298/CSIS100209016D. 133, 134

[49] Ricardo Fernandes, Michael Hieb, and Paulo Costa. Levels of autonomy: Command and control of hybrid forces. In *21th ICCRTS – C2 in a Complex Connected Battlespace*, 2016. 133, 134

[50] Ricardo Fernandes, Michael Hieb, and Paulo C.G. Costa. Levels of Autonomy: Command and Control of Hybrid Forces. In *21st International Command and Control Research and Technology Symposium (ICCRTS-KSCO 2016)*, page 16, London, United Kingdom, 2016. URL `http://internationalc2institute.org/2016-t3`. 1, 5, 84, 86

[51] Paulo R Ferreira Jr, Felipe S Boffo, and Ana LC Bazzan. A swarm based approximated algorithm to the extended generalized assignment problem (e-gap). In *Proceedings of the 6th International Joint Conference on Autonomous Agents And Multiagent Systems (AAMAS)*, pages 1231–1233. ACM, 2007. 29, 31, 32, 44, 48

[52] Paulo Roberto Ferreira Jr, Fernando Dos Santos, Ana LC Bazzan, Daniel Epstein, and Samuel J Waskow. Robocup rescue as multiagent task allocation among teams: experiments with task interdependencies. *Autonomous Agents and Multi-Agent Systems*, 20(3):421–443, 2010. 40

[53] T French. Quantified propositional temporal logic with repeating states. In *10th International Symposium on Temporal Representation and Reasoning, 2003 and Fourth International Conference on Temporal Logic. Proceedings.*, pages 155–165, jul 2003. doi: 10.1109/TIME.2003.1214891. 140

[54] Tim French and Mark Reynolds. A Sound and Complete Proof System for QPTL. *Advances in Modal Logic*, 2002. 140

[55] H. Friman and G. E. Horne. Using agent models and data farming to explore network centric operations. In *Winter Simulation Conference*, pages 7 pp.–, 2005. doi: 10.1109/WSC.2005.1574364. 84

[56] Xiang Gao and Jianxing Yu. Public governance mechanism in the prevention and control of the COVID-19: information, decision-making and execution. *Journal of Chinese Governance*, pages 1–20, 2020. doi: 10.1080/23812346.2020.1744922. 1, 141

[57] S Gerasimou, R Calinescu, S Shevtsov, and D Weyns. UNDERSEA: An Exemplar for Engineering Self-Adaptive Unmanned Underwater Vehicles. In *2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pages 83–89, may 2017. doi: 10.1109/SEAMS.2017.19. 134, 135

[58] Asghar Ghasemi and Saleh Zahediasl. Normality Tests for Statistical Analysis: A Guide for Non-Statisticians. *International journal of endocrinology and metabolism*, 10:486–489, 2012. doi: 10.5812/ijem.3505. 93

[59] Jesús M. González-Barahona and Gregorio Robles. On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Empirical Software Engineering*, 17(1):75–89, Feb 2012. ISSN 1573-7616. doi: 10.1007/s10664-011-9181-9. URL `https://doi.org/10.1007/s10664-011-9181-9`. 35

[60] Google. Google Forms, 2020. URL `https://www.google.com/forms/about/`. 97, 116

[61] John Grant, Sarit Kraus, and Don Perlis. *Formal Approaches to Teamwork*. NSF Award Search, 2005. 133, 134

[62] Lucas Gren and Per Lenberg. Agility is responsiveness to change: An essential definition. In *Evaluation and Assessment in Software Engineering*, pages 348–353, New York, NY, USA, 04 2020. Association for Computing Machinery. ISBN 9781450377317. doi: 10.1145/3383219.3383265. URL `https://doi.org/10.1145/3383219.3383265`. 12, 18, 61, 80

[63] Svein Hallsteinsen, Mike Hinchey, Sooyong Park, and Klaus Schmid. Dynamic software product lines. *Computer*, 41(4):93–95, April 2008. ISSN 0018-9162. doi: 10.1109/MC.2008.123. URL `https://doi.org/10.1109/MC.2008.123`. 74

[64] Svein O Hallsteinsen, Erlend Stav, Arnor Solberg, and Jacqueline Floch. Using Product Line Techniques to Build Adaptive Systems. In *Software Product Lines, 10th International Conference, SPLC 2006, Baltimore, Maryland, USA, August 21-24, 2006, Proceedings*, pages 141–150. IEEE Computer Society, 2006. ISBN 0-7695-2599-7. doi: 10.1109/SPLINE.2006.1691586. URL `https://doi.org/10.1109/SPLINE.2006.1691586`. 62

[65] Mike Hinchey, Sooyong Park, and Klaus Schmid. Building Dynamic Software Product Lines. *IEEE Computer*, 45(10):22–26, oct 2012. doi: 10.1109/MC.2012.332. 25

[66] Rashina Hoda, James Noble, and Stuart Marshall. Developing a Grounded Theory to Explain the Practices of Self-Organizing Agile Teams. *Empirical Softw. Engg.*, 17(6):609–639, dec 2012. ISSN 1382-3256. doi: 10.1007/s10664-011-9161-0. URL `https://doi.org/10.1007/s10664-011-9161-0`. 95

[67] Gerald Holl, Paul Grünbacher, and Rick Rabiser. A systematic review and an expert survey on capabilities supporting multi product lines. *Information and Software Technology*, 54(8):828 – 852, 2012. ISSN 0950-5849. doi: https://doi.org/10.1016/j.infsof.2012.02.002. URL `http://www.sciencedirect.com/science/article/pii/S095058491200033X`. Special Issue: Voice of the Editorial Board. 62

[68] M. Usman Iftikhar and Danny Weyns. Activforms: Active formal models for self-adaptation. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS 2014, page 125–134, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328647. doi: 10.1145/2593929.2593944. URL `https://doi.org/10.1145/2593929.2593944`. 131

[69] Muhammad Jaseemuddin, Abrar Alam, and Nuzhat Gawhar. Mqtt pub-sub service for connected vehicles. In *2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, pages 167–172, 2021. doi: 10.1109/HONET53078.2021.9615393. 71

[70] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, 1998. 86

[71] Nicholas R. Jennings and Stefan Bussmann. Agent-based control systems. *IEEE Control Systems Magazine*, pages 61–73, 2003. 133, 134

[72] Carl-Oscar Jonson, Heléne Nilsson, R Lundin, and A Rüter. (P1-79) Regional Medical Command and Control Management of Influenza A (H1N1) Mass-Vaccination in the County of Östergötland, Sweden. *Prehospital and Disaster Medicine*, 26, 2011. doi: 10.1017/S1049023X11004110. 1

[73] Kelin Jose and Dilip Kumar Pratihar. Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems*, 80:34–42, 2016. 130, 133, 134

[74] Anselm Strauss Juliet M. Corbin. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory.* SAGE Publications, Inc, 3rd edition, 2007. ISBN 1412906431,9781412906432. 103

[75] Natalia Juristo and Omar S. Gómez. Replication of software engineering experiments. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7007 LNCS(19):60–88, 2012. ISSN 03029743. doi: 10.1007/978-3-642-25231-0_2. 30, 35

[76] A. Spencer Kang, Kyo C; Cohen, Sholom G.; Hess, James A.; Novak, William E.; Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Carnegie Mellon University, feb 1990. 24, 74

[77] A Kassambara. *Practical Statistics in R for Comparing Groups: Numerical Variables.* Practical Statistics in R Series. Independently Published, 2019. ISBN 9781712330883. 89, 93

[78] Jeffrey Kephart and D M Chess. The Vision Of Autonomic Computing. *Computer*, 36:41–50, 2003. doi: 10.1109/MC.2003.1160055. 63

[79] Barbara Kitchenham and Shari Pfleeger. Principles of survey research part 6: Data analysis. *ACM SIGSOFT Software Engineering Notes*, 28:24–27, 2003. doi: 10.1145/638750.638758. 96, 101

[80] Joshua Knowles and David Corne. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, volume 1, pages 98–105, feb 1999. doi: 10.1109/CEC.1999.781913. 30

[81] Cyprian Kozera. Military Use of Unmanned Aerial Vehicles – A Historical Study. *Safety & Defense*, 4:17–21, 2018. doi: 10.37105/sd.4. 141

[82] Sarit Kraus and Barbara Grosz. Collaborative plans for complex group action. *Artificial Inteligence*, 86(2):269–357, 1996. 133, 134

[83] Gabriel Leal, Iulisloi Zacarias, Jorgito Stocchero, and Edison de Freitas. Empowering Command and Control through a Combination of Information-Centric Networking and Software Defined Networking. *IEEE Communications Magazine*, 57: 48–55, 2019. doi: 10.1109/MCOM.2019.1800288. 1, 4, 82, 84, 129, 130, 134

[84] Thomas Lemaire, Rachid Alami, and Simon Lacroix. A distributed tasks allocation scheme in multi-uav context. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3622–3627. IEEE, 2004. 29

[85] Alexandros Leontitsis and Jenny Pagge. A simulation approach on Cronbach's alpha statistical significance. *Mathematics and Computers in Simulation*, 73:336–340, 2007. doi: 10.1016/j.matcom.2006.08.001. 96

[86] Michael Lienhardt, Ferruccio Damiani, Simone Donetti, and Luca Paolini. Multi software product lines in the wild. In *ACM International Conference Proceeding Series*, pages 89–96, feb 2018. ISBN 9781450353984. doi: 10.1145/3168365.3170425. 25, 132

[87] Michael Lienhardt, Ferruccio Damiani, Simone Donetti, and Luca Paolini. Multi software product lines in the wild. In *ACM International Conference Proceeding Series*, pages 89–96, feb 2018. ISBN 9781450353984. doi: 10.1145/3168365.3170425. 132

[88] Malte Lochau, Johannes Bürdek, Stefan Hölzle, and Andy Schürr. Specification and automated validation of staged reconfiguration processes for dynamic software product lines. *Software & Systems Modeling*, 16, 2015. doi: 10.1007/s10270-015-0470-4. 132

[89] Lech Madeyski and Barbara Kitchenham. Would wider adoption of reproducible research be beneficial for empirical software engineering research? *Journal of Intelligent and Fuzzy Systems*, 32(2):1509–1521, 2017. ISSN 18758967. doi: 10.3233/JIFS-169146. 35, 53

[90] Sara Mahdavi Hezavehi, Vinicius Durelli, Danny Weyns, and Paris Avgeriou. A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems. *Information and Software Technology*, 2017. doi: 10.1016/j.infsof.2017.03.013. 63

[91] Jonathan Russell Agre Marius S. Vassiliou David S. Alberts. *C2 Re-envisioned: The Future of the Enterprise*. CRC Press, 2015. ISBN 1466595809,9781466595804. 1

[92] Colin R. Mason and James Moffat. Representing the C2 process in simulations: Modelling the human decision-maker. *Winter Simulation Conference Proceedings*, 1:940–949, sep 2000. ISSN 02750708. doi: 10.1109/WSC.2000.899896. 1, 10, 129, 130

[93] James Moffat. *Complexity Theory and Network Centric Warfare*. DoD Command and Control Research Program, 2003. ISBN 1-893723-11-9. 3

[94] James Moffat, Reiner Huber, and David Alberts. Achieving agile c2 by adopting higher levels of c2 maturity. In *International Command and Control Research and Technology Symposium*, 06 2012. 55

[95] S Niemczyk and K Geihs. Adaptive Run-Time Models for Groups of Autonomous Robots. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 127–133, may 2015. doi: 10.1109/SEAMS.2015.21. 2

[96] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer Science & Business Media, 2010. 29

[97] Michael J North, Nicholson T Collier, Jonathan Ozik, Eric R Tatara, Charles M Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, 1(1):3, 2013. ISSN 2194-3206. doi: 10.1186/2194-3206-1-3. URL `https://doi.org/10.1186/2194-3206-1-3`. 86, 134

[98] Obeo. Obeo Designer Modeling Tool, 2018. URL `https://www.obeodesigner.com/en/`. 64

[99] Leonardo Pessoa, Paula Fernandes, Thiago Castro, Vander Alves, Gena\'\ina N Rodrigues, and Hervaldo Carvalho. Building reliable and maintainable Dynamic Software Product Lines: An investigation in the Body Sensor Network domain. *Information and Software Technology*, 86:54–70, 2017. 132

[100] Shari Pfleeger and Barbara Kitchenham. Principles of survey research: Part 1: Turning lemons into lemonade. *ACM SIGSOFT Software Engineering Notes*, 26: 16–18, 2001. doi: 10.1145/505532.505535. 94, 95, 97, 116

[101] Xingcheng Pu, Li Ren, Yi Liu, and Lianghao Ji. Group consensus of multi-agent systems with cooperative-competitive interaction and communication delay in switching topologies networks based on the delta operator method. *Neurocomputing*, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.01.076. URL `http://www.sciencedirect.com/science/article/pii/S0925231220301284`. 30

[102] A. Ronveaux. Quasi-orthogonality and differential equations. *Journal of Computational and Applied Mathematics*, 29(2):243–247, 1990. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(90)90361-3. URL `https://www.sciencedirect.com/science/article/pii/0377042790903613`. 62

[103] Marko Rosenmüller and Norbert Siegmund. Automating the Configuration of Multi Software Product Lines. In *VaMoS*, 2010. 132

[104] Marko Rosenmüller, Norbert Siegmund, Mario Pukall, and Sven Apel. Tailoring Dynamic Software Product Lines. *ACM Special Interest Group on Programming Languages Notices*, 47(3):3–12, oct 2011. ISSN 0362-1340. doi: 10.1145/2189751.2047866. 131, 132

[105] Eric Rutten, Nicolas Marchand, and Daniel Simon. *Feedback Control as MAPE-K Loop in Autonomic Computing*, pages 349–373. Springer, 01 2018. ISBN 978-3-319-74182-6. doi: 10.1007/978-3-319-74183-3_12. 133

[106] Fernando Santos, Ingrid Nunes, and Ana L C Bazzan. Model-driven agent-based simulation development: A modeling language and empirical evaluation in the adaptive traffic signal control domain. *Simulation Modelling Practice and Theory*, 83:162–187, 2018. ISSN 1569-190X. doi: https://doi.org/10.1016/j.simpat.2017.11.006. URL `http://www.sciencedirect.com/science/article/pii/S1569190X17301673`. 134, 135

[107] Pierre-Yves Schobbens, Patrick Heymans, and Jean-Christophe Trigaux. Feature Diagrams: A Survey and a Formal Semantics. *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 139–148, 2006. 24, 74

[108] Janaína Schwarzrock, Iulisloi Zacarias, Ana L. C. Bazzan, Edison P. de Freitas, Ricardo Queiroz de Araujo Fernandes, and Leonardo Henrique Moreira. Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence. *Engineering Applications of Artificial Intelligence*, 2017. 5, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 44, 48, 50, 51, 53, 87, 92, 130, 133, 134

[109] C B Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, 1999. doi: 10.1109/32.799955. 101, 118, 126

[110] Amir Molzam Sharifloo, Andreas Metzger, Clément Quinton, Luciano Baresi, and Klaus Pohl. Learning and evolution in dynamic software product lines. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2016, Austin, Texas, USA, May 14-22, 2016*, pages 158–164. ACM, 2016. ISBN 978-1-4503-4187-5. doi: 10.1145/2897053.2897058. URL `https://doi.org/10.1145/2897053.2897058`. 132

[111] Nidhi Sharma. 45 smart city command and control centres turn into Covid-19 war rooms, 2020. `https://economictimes.indiatimes.com/\news/politics-and-nation/45-smart-city-command-and-control-centres-turn-into-covid-19-war-rooms/articleshow/74904329.cms`, last accessed on 06/01/2020. 1, 141

[112] Shuzi Shimizu, Ho-Seong Han, Kuriko Kudo, Yoshiko Houkabe, Yasuaki Antoku, and Naoki Nakashima. *Chapter 123 Telemedicine: Principles and the Future*, volume 2, pages 1352–1356. Jaypee Brothers Medical Publishers (P) Ltd, 2016. ISBN 9789351529989. doi: 10.5005/jp/books/12748_124. 58

[113] Eneias Silva, Alessandro Leite, Vander Alves, and Sven Apel. ExpRunA : a domain-specific approach for technology-oriented experiments. *Software & Systems Modeling*, 2019. ISSN 1619-1374. doi: 10.1007/s10270-019-00749-6. URL `https://doi.org/10.1007/s10270-019-00749-6`. 54

[114] SmartCitiesWorld. Command-and-control platform aims to help cities better use their data to tackle Covid-19, 2020. `https://www.smartcitiesworld.net/news/news/command-and-control-platform-aims-to-help-cities-better-use-their-data-to-tackle-covid-19-5225`, last accessed on 07/18/2020. 1, 141

[115] Gustavo Sousa, Walter Rudametkin, and Laurence Duchien. Extending Dynamic Software Product Lines with Temporal Constraints. In *12th {IEEE/ACM} International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2017, Buenos Aires, Argentina, May 22-23, 2017*, pages 129–139. {IEEE} Computer Society, 2017. ISBN 978-1-5386-1550-8. doi: 10.1109/SEAMS.2017.6. 133, 134

[116] Neville A. Stanton, Guy Walker, Daniel Jenkins, Paul Salmon, Mark Young, and Amerdeep Aujla. Models of Command and Control. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4562 LNAI, pages 600–608, sep 2007. ISBN 9783540733300. doi: 10.1007/978-3-540-73331-7_66. 1, 10, 86, 129, 130

[117] United States. Department of Defense dictionary of military and associated terms [electronic resource] / Joint Chiefs of Staff, feb 2019. URL `https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/dictionary.pdf`. 10

[118] Kunio Takezawa. Basic graphics. In *Guidebook to R Graphics Using Microsoft® Windows*, chapter 1, pages 1–64. John Wiley & Sons, Ltd, 2012. ISBN 9781118270165. doi: https://doi.org/10.1002/9781118270165.ch1. 89

[119] Guy Theraulaz, Eric Bonabeau, and JN Denuebourg. Response threshold reinforcements and division of labour in insect societies. *Proceedings of the Royal Society of London B: Biological Sciences*, 265(1393):327–332, 1998. 31

[120] Alexander Tiskin. *Bulk Synchronous Parallelism (BSP)*, pages 192–199. Springer US, Boston, MA, 01 2011. ISBN 978-0-387-09766-4. doi: https://doi.org/10.1007/978-0-387-09766-4. 143

[121] Huy Tran, Jean Domercant, and Dimitri Mavris. Evaluating the agility of adaptive command and control networks from a cyber complex adaptive systems perspective. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 12, 2015. doi: 10.1177/1548512915592517. 3, 80, 86, 129, 130

[122] Guilherme Travassos and Márcio Barros. Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering. *The Future of Empirical Studies in Software Engineering: Proceedings of the ESEIW 2003 Workshop on Empirical Stuides in Software Engineering, WSESE 2003; Roman Castles, Italy, September 29th, 2003*, 2:117–130, 2003. 58, 79

[123] Frank Trollmann, Johannes Fähndrich, and Sahin Albayrak. Hybrid adaptation policies: towards a framework for classification and modelling of different combinations of adaptation policies. In Jesper Andersson and Danny Weyns, editors, *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2018, Gothenburg, Sweden, May 28-29, 2018*, pages 76–86. ACM, 2018. doi: 10.1145/3194133.3194137. 132

[124] M. Tsvetkov and C. Alexandrov. Software simulator of marine monitoring platform. In *2016 19th International Symposium on Electrical Apparatus and Technologies (SIELA)*, pages 1–4, 2016. doi: 10.1109/SIELA.2016.7543057. 59, 134

[125] Rini Van Solingen, Vic Basili, Gianluigi Caldiera, and H Dieter Rombach. Goal question metric (gqm) approach. *Encyclopedia of software engineering*, 01 2002. 80

[126] Duncan J Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness (Princeton Studies in Complexity)*. Princeton University Press, 2003. ISBN 0691117047,9780691117041. 15

[127] Claes Wohlin, Per Runeson, Martin Höst, Magnus Ohlsson, Björn Regnell, and Anders Wesslén. *Analysis and Interpretation*, pages 123–151. Springer-Verlag Berlin Heidelberg, 05 2012. ISBN 978-3-642-29043-5. doi: 10.1007/978-3-642-29044-2_10. 30, 35, 86, 135

[128] Husheng Wu, Hao Li, Renbin Xiao, and Jie Liu. Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm. *Physica A: Statistical Mechanics and its Applications*, 491:127–141, 2018. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2017.08.094. URL `http://www.sciencedirect.com/science/article/pii/S0378437117308166`. 30

[129] MD Wurmb, MD Ertl, MD Ernestus, and Patrick Meybohm. Command and control in hospitals during sars-cov-2 pandemic: The windmill model of disaster response. *Journal of Emergency Management*, 18:19–22, 12 2020. doi: 10.5055/jem.2020.0520. 141

[130] H Zhuge and X Sun. A Virtual Ring Method for Building Small-World Structured P2P Overlays. *IEEE Transactions on Knowledge and Data Engineering*, 20(12): 1712–1725, dec 2008. ISSN 1041-4347. doi: 10.1109/TKDE.2008.102. 15

# Appendix A

# Survey 1 Questions (Original Language)

Below are the Survey 1 questions, containing 2 sections, with their respective original texts in Portuguese (Brazil), the objective of each question, and the response type description.

**a) Seção 1**

**Pergunta 1:** O Cenário 1 apresentado utiliza-se de simulações de uso de VANTs através de software, onde há respostas a mudanças de contexto. O Sr concorda que este cenário é relevante para a sua Organização/Unidade?

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente)

Objetivo: Atender ao item 2.2 dos objetivos recebendo o retorno da real relevância do cenário proposto quando comparado com as situações enfrentadas pelo domínio.

**Pergunta 2:** Justifique a resposta anterior (máx. 15 linhas)

Tipo de resposta: Texto livre

Objetivo: Permitir ao especialista exprimir as motivações da resposta anterior de modo a identificarmos o grau de distanciamento do cenário simulado em relação às aplicações no mundo real.

**Pergunta 3:** O que poderia ser incluído/alterado no referido cenário de modo a torná-lo mais próximo de uma situação real de interesse para a sua Organização/Unidade?

Tipo de resposta: Texto livre

Objetivo: Permitir ao especialista listar o que poderia ser incluído e/ou alterado no cenário de simulação apresentado de modo que ele se torne mais fiel às condições reais vividas pela organização/unidade do entrevistado.

**Pergunta 4:** Sob seu ponto de vista e experiência, quais seriam as quantidades adequadas para cada elemento do Cenário 1 apresentado?

Tipo de resposta: Fechada com números pré-definidos para serem selecionados

Objetivo: Obter do especialista do domínio, quais serias as quantidades mais adequadas de elementos no cenário de modo a tornar-se mais aderente ao mundo real.

**Pergunta 5:** O slide 20 caracteriza o cenário 1 como sendo composto de 1 VANT executando os papeis de coordenador e distribuidor das tarefas, e os demais VANTs com a função de executor. Nesse contexto, o coordenador possui total autonomia (não depende de um comando central) e utiliza a Estratégia de C2 para a troca de informações com o restante do grupo/fração. Considerando a lista abaixo de possíveis respostas do sistema a eventuais mudanças de contexto, como o Sr classifica cada uma delas em termos de relevância/importância? (1 indica nenhuma relevância/importância e 5 indica totalmente relevante/importante)

Tipo de resposta: Escala de 1 a 5 para cada reação do sistema em resposta às mudanças de contexto.

Objetivo: Permitir que o especialista do domínio classifique, em grau de relevância, as atuais possíveis respostas implementadas no cenário simulado apresentado.

**Pergunta 6:** Revendo os slides e o vídeo que descrevem o Cenário 1, podemos afirmar que o sistema composto pelos VANTs é dotado de agilidade. Qual é o nível de concordância do Sr diante dessa afirmação?

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente).

Objetivo: Identificar se, sob a visão do especialista, o cenário é capaz de deixar explícito as características de um sistema ágil aplicando estratégias de C2.

**Pergunta 7:** Considerando o Cenário 1 dos slides apresentados, quais dos recursos listados abaixo conferem agilidade de C2 ao time de VANTs ?

Tipo de resposta: Múltipla escolha com 6 alternativas que podem ser marcadas livremente.

Objetivo: Identificar a capacidade do especialista em caracterizar o sistema ágil apresentado por meio do Cenário 1.

**Seção 2**

**Pergunta 1:** Descreva um cenário dinâmico planejado ou vivido pela sua Unidade ou Organização com a aplicação dos princípios de C2. (Máximo de 30 linhas)

Tipo de resposta: Texto livre.

Objetivo: Permitir que o especialista descreva, de modo livre e com a linguagem do domínio, um cenário que aplique os princípios de C2 e que seja dinâmico.

**Pergunta 2:** Baseado na descrição das Estratégias de C2 apresentadas nos slides disponibilizados, qual delas está sendo operada pela Organização/Unidade no cenário descrito pelo Sr. ?

Tipo de resposta: Múltipla escolha com 6 alternativas onde apenas 1 pode ser marcada.

Objetivo: Verificar se o especialista do domínio identifica alguma das Estratégias de C2 apresentadas em um cenário real vivido pela própria organização.

**Pergunta 3:** Baseado no cenário dinâmico da sua Unidade/Organização descrito pelo Sr, quais das ações abaixo são aplicadas em resposta a uma mudança de contexto durante a execução da missão?

Tipo de resposta: Múltipla escolha com 9 alternativas que podem ser marcadas livremente.

Objetivo: Permitir que o especialista identifique quais comportamentos esperados pelo cenário do domínio conferem resposta à possibilidade de mudanças de contexto.

**Pergunta 4:** No Cenário descrito pelo Sr., a Unidade/Fração é dotada de AGILIDADE?

Tipo de resposta: Múltipla escolha com 2 alternativas (SIM/NÃO) onde apenas 1 pode ser marcada.

Objetivo: Verificar se, sob o ponto de vista do especialista do domínio, o cenário real apresentado garante agilidade à sua Unidade/Fração.

**Pergunta 5:** Justifique a resposta anterior.

Tipo de resposta: Texto livre.

Objetivo: Entender o ponto de vista do especialista do domínio acerca da avaliação da sua própria Organização/Unidade em termos de nível de agilidade.

**Pergunta 6:** É fundamental o cumprimento total da missão, incluindo todas as tarefas que a compõe. Qual é o nível de concordância do Sr diante dessa afirmação?

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente).

Objetivo: Identificar se, para o domínio, existe coerência e relevância na execução parcial de uma missão, permitindo que os elementos possam descartar as tarefas que por ventura não sejam realizáveis.

**Pergunta 7:** A missão deve sempre ser cumprida no menor tempo possível, em detrimento da qualidade de sua execução. Qual é o nível de concordância do Sr diante dessa afirmação?

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente).

Objetivo: Identificar a visão do especialista do domínio sobre a relação velocidade versus qualidade do cumprimento de uma missão.

**Pergunta 8:** Qual(is) papel(eis)/função(ções) podem ser identificados no cenário apresentado pelo Sr?

Tipo de resposta: Múltipla escolha com 4 alternativas que podem ser marcadas livremente.

Objetivo: Verificar se o especialista do domínio identifica algum dos papéis definidos em nosso modelo no cenário apresentado por ele representando uma situação real.

# Appendix B

# Survey 1 Answers to Open-ended Questions (Original Language)

This appendix shows the answers to the open-ended questions of Survey 1 (Section 5.4) in the original language (Portuguese-BR). From these answers, we extracted the codes that gave rise to the planned grounded theory.

**Table B.1:** Answers to open questions (Section 1/Question 2; Section 1/Question 3; Section 2/Question 1; Section 2/Question 5) of Survey #1 in the original language (Portuguese-BR)

| Attendee | Question | Answer |
|---|---|---|
| P1 | Q1 | Ocasionalmente, há mudanças no contexto da missão que impactam em cumprimento. Importante definir as alternativas para a adoção de estratégias de comunicação. |
| | Q2 | Inclusão de novos VANT durante a missão e a reprogramação de responsabilidades distintas a cada um (incluir mudança de tempo de reconfiguração da nova estratégia). Deve-se prever também interferências momentâneas durante a comunicação entre VANT para medir os efeitos. |

*Continued on next page*

TableB.1 - *continued from previous page*

| Attendee | Question | Answer |
|---|---|---|
| | Q3 | Uma Pelotão C Mec destacado a frente para realização de uma tarefa específica se desloca em um terreno acidentado de forma que, durante seu deslocamento, há perda intermitente de comunicação entre seus elementos e com sua base. A tropa executa sua missão e reporta a informação logo após seu cumprimento. Durante o cumprimento, pode haver situações de solicitação de apoio ou reporte de informações esporádicas. Neste contexto, cresce de importância os períodos de comunicações para realizar os apoios solicitados e/ou reportes esporádicos. também é importante ressaltar quanto a sinalização de presença de tropa (do próprio pelotão ou de outras frações) para não ocorrência de fogo amigo. Neste cenário, utiliza-se VANT para apoio às coordenações de ação e manutenção da consciência situacional durante todo o período da missão e, com especial foco, durante os momentos críticos. |
| | Q4 | A necessidade de pronta resposta às novas condicionantes da missão. Neste caso, cresce de importância a conduta e o aspecto de liderança para coordenação de ações de pronta resposta. |
| P2 | Q1 | O uso de VANTs por meio de software promove segurança e precisão das informações. |
| | Q2 | Interação com aeronaves e blindados. |
| | Q3 | Operação de cerco a uma área urbana. |
| | Q4 | Devido a capacidade de movimentação e articulação. |
| P3 | Q1 | A mudança de cenário é algo muito comum em missões reais; digo isso baseado nas 03 operações das quais participei. |
| | Q2 | Nada.Estão bem próximo de um cenário possível. |
| | Q3 | Operação de Contra-Guerrilha Urbana, nível Cia Fuz, no Complexo da Maré. |
| | Q4 | As frações tinham autonomia para tomar diversas decisões. |
| P4 | Q1 | Vai auxiliar para obtenção de consciencial situacional |
| | Q2 | O mapa da região |

*Continued on next page*

TableB.1 - *continued from previous page*

| Attendee | Question | Answer |
|---|---|---|
| | Q3 | Um navio de transporte de tropa precisa desembarcar 140 (cento e quarenta) fuzileiros navais em uma pequena praia para dominar e controlar o acesso a referida localidade. Serão utilizadas 4 LCVP (Landing Craft Vehicle/Personnel). |
| | Q4 | Pela capacidade de se reorganizar em função de uma eventual, alteração do cenário previsto. |
| P5 | Q1 | O Gab Cmt Ex (atual OM), por estar em um nível estratégico, poderia se valer do uso de VANTs para difundir informações táticas e operacionais ao mesmo tempo que coleta dados para obter a consciência situacional. |
| | Q2 | Dependendo da tarefa e terreno (ambiente), seria interessante haver múltiplos VANTs no papel de coordenador. VANTs executores também poderiam ter condições de imediatamente assumirem o papel de coordenador caso o coordenador seja abatido. A estratégica de C2 Agregado (Edge) parece mais adequada em situações onde existe eminência de ataque e a necessidade de manter a conectividade, fluxo de informações e capacidade de tomada de decisões. |

*Continued on next page*

TableB.1 - *continued from previous page*

| Attendee | Question | Answer |
|---|---|---|
| | Q3 | 2006, Cucuí/AM, tríplice fronteira (Brasil, Venezuela e Colômbia). Após denúncia, em uma incursão não planejada, cinco militares são deslocados para averiguar suposta movimentação irregular no interior da selva. A Operação é comandada por um tenente e conta com três sargentos e um cabo. O objetivo é encontrar os meliantes, porém, a densa selva dificulta a progressão. Após 40 minutos de buscas sem sucesso, a fração foi supreendida por um disparo de arma de fogo contra um dos militares, que felizmente não foi alvejado. Em uma imediata reação conjunta, o confronto foi inevitável. Resultado, um meliante fugiu, dois se renderam e um morreu no local. Foram apreendidos armas, munições e dinheiro que teriam destino às FARCs (Colômbia). Os cinco militares retroagiram à base (pelotão) trazendo os prisioneiros e os produtos do contrabando. Posteriormente, segundo informes da Inteligência, ficou-se sabendo que haviam ainda pelo menos mais 20 meliantes agrupados em outros pontos que, inclusive, cogitavam retaliação para resgatar seus comparsas e bens perdidos. O pelotão acionou o plano de defesa do aquartelamento (PDA) até a chegada do reforço 24 horas após o ocorrido. |
| | Q4 | A equipe não dispunha de meios apropriados para lidar com a situação, exceto pelo armamento e treinamento militar, mas que lhes salvaram a vida. |
| P6 | Q1 | Contexto, atitude e reação à mudança são necessários a qualquer contexto computacional. Não somente para aplicação em VANTS mas também códigos e aplicações inteligentes capazes de mudarem comportamento e decisões de maneira independente de acordo com novos dados ou situações. |

*Continued on next page*

TableB.1 - *continued from previous page*

| Attendee | Question | Answer |
|---|---|---|
| | Q2 | A Cadeia de Comando e distribuição de tarefas para C2 em combate são similares a mesma estrututa da composição para Aquisições Internacionais. Assim um código capaz de adaptar-se a diferentes mudanças do contexto e influências orçamentárias, financeiras ou operacionais pode colaborar no contexto logístico. |
| | Q3 | Planejamento e execução de aquisição internacional. |
| | Q4 | O cumprimento da missão estabelecido pelo exercício orçamentário anual ou imposição da necessidade e urgência operacional são razões suficientes e necessárias para exigir agilidade dos atores envolvidos. |
| P7 | Q1 | Missões reais sempre envolvem mudanças de contexto. |
| | Q2 | Para Operação autônoma, não vejo maiores acréscimos, mas para acompanhamento pelo escalão superior, informações sobre o status de autonomia (combustível) e disponibilidade dos vants durante a missão auxiliam na antecipação de recompletamento no caso de abate de vants. |
| | Q3 | Forças híbridas em Operação de C2 (Defesa Civil, Exército, Bombeiros, Força Policial, etc). |
| | Q4 | Dificuldade de Interação. |
| P8 | Q1 | Esse recurso de Rec foi utilizado nas Op de pacificação no Haiti, com muito sucesso, evitou emboscadas. 2006. Também utilizada na Maré, RJ. |
| | Q2 | Plj de modo a fasear as ações de coordenação. Com Plj Pcp, Alt, Abortivo. Se presta bem às Op Esp. Exige adaptações às Op conv. Outras ações táticas nas LCt... |
| | Q3 | As peças de Op Esp possuem as L Cd de atingimento dos Obj nos 5 campos do poder. Tais obj foram reportados por diversos meios conv ou não, Mnt a consciência situacional. Qto mais meios eu tivesse, melhor, pois ao longo da Op os meios se perdem. Um complementa o outro. |
| | Q4 | Os meios disponíveis p Op mais complexa p FEsp, Guerrilha, devem permitir a sobreposição de meios (Rad, fio, cache Msg, Msg BMP, Pacotes, internet.....) |

*Continued on next page*

TableB.1 - *continued from previous page*

| Attendee | Question | Answer |
|---|---|---|
| P9 | Q1 | O emprego de um sistema de VANTs é objeto de estudo para os próximos anos na Artilharia de Campanha. Suas capacidades em automatizações são desejáveis, desde que haja possibilidade de controle humano que evite perdas desnecessárias. |
| | Q2 | Os momentos em que a intervenção humana pudesse ser necessária, sendo ideal a qualquer momento, justamente devido a dinâmica da Operação. |
| | Q3 | Necessidade de busca e análise de alvos para diversos níveis para a Artilharia de Campanha. Entretanto tal situação foi realizada de forma simulada e sem estratégia de C2 conforme descrito nos slides apresentados nesta pesquisa. |
| | Q4 | O conceito de agilidade de C2 utilizado nos slides não cabe por exemplo a uma fração de VANTS sendo empregados, necessariamente. Mas cabe a adaptação de baterias ou grupos de artilharia sendo realocados para missões táticas diferentes no decorrer de uma Operação. Se pensarmos nesse segundo caso, a resposta seria sim. |
| P10 | Q1 | O atual cenário de conflitos exige elevada capacidade de adaptação e resposta a novas situações ou imprevistos |
| | Q2 | Nada a acrescentar |
| | Q3 | Operação de Garantia da Lei e da Ordem em ambiente urbano, com extrema volatilidade da situação. Elementos de Força Adversa misturados entre a população. Emprego dos meios (pequenas frações de soldados) de maneira dispersa. Dificuldade de C2 e consciência situacional. E elevada necessidade de decisões rápidas. |
| | Q4 | Os meios disponíveis permitem rápido deslocamento e concentração de meios para redistribuição. |
| P11 | Q1 | As missões de Guerra Eletrônica são caracterizadas por mudança de contexto. |
| | Q2 | Surgimento de ameaças que alteram todos os fatores. |

*Continued on next page*

TableB.1 - *continued from previous page*

| Attendee | Question | Answer |
|---|---|---|
| | Q3 | Atuação de sensores de Guerra Eletrônica em ambientes de espectro eletromagnetico denso, com multiplos emissores ou diante de atividades de medidas de ataque eletrônico. |
| | Q4 | A agilidade de C2 permite capacidade atuacao de atuacao dinâmica com eficiência em cenários dinamicos que não podem ser previstos até o inicio da missão, além da integração da consciência situational de todos os participantes. |
| P12 | Q1 | Independente da missão da Organização/Unidade o cenário sempre estará sujeito a mudanças de contexto de acordo com cada missão, que podem ser mudanças nas tarefas, no disponibilidade de meios, etc. Logo é de extrema relevância que a estrutura de C2 responde com agilidade à mudanças de contexto. |
| | Q2 | Acredito que como há uma estrutura de comunicações nos drones, onde os mesmos possuem a capacidade de se conectar com outros drones/membros e assim trocar informações em redes de dados, pode-se dizer que o cenário estão bem próximo da realidade de um sistema de C2. Onde as redes de dados devem ser de topologia dinâmica (ad-hoc), modificadas de acordo com as alterações no cenário. O que talvez tornaria mais próximo da situação real seria a Inclusão da representação de um elemento terrestre estático que receberia e controlaria todo tráfego de informações na rede e redirecionaria para o Escalão Superior, que seria algo como um Centro de Operações (COp), onde o Cmt da missão acompanharia as evoluções de cenário e interviria SFC (adicionando mais drones, ou mesmo retraindo e abortando a missão). |
| | Q3 | O apoio era realizado a partir da composição dos Centros de C2 de Comandos Conjuntos e da Força Terrestre Componente (FTC) e ligação dos meios de C2 dos elementos subordinados. |
| | Q4 | A existência de Centros de C2 agiliza a tomada de decisões e as reações às mudanças de cenário. |

*Continued on next page*

TableB.1 - *continued from previous page*

| Attendee | Question | Answer |
|---|---|---|
| P13 | Q1 | Pela natureza dos meios empregados, não há dúvidas sobre a relevância da autorregulação dos dispositivos, implicando rápida adaptação às novas situações, recobrimento de tarefas e o cumprimento, ao máximo, das missões originalmente planejadas. |
| | Q2 | há necessidade de informações de combate. Além de terreno, missão e meios, exige-se informações das potencialidades da força adversa que podem comprometer a missão planejada. |
| | Q3 | No tempo em que comandava pequenas frações (pelotão e subunidade), ainda não existia conceitos formais de C2. Na prática, contudo, o comando e o controle eram feitos por sinalização visual, eventualmente, e orevalentemente por comunicação via rádio. |
| | Q4 | As principais características da fração são mobilidade, proteção blindada e potência de fogo. Combinadas, essas possibilidades conferem agilidade na readaptação às condições do ambiente. |

# Appendix C

# Survey 2 Questions (Original Language)

Below are the Survey 2 questions, with their respective original texts in Portuguese (Brazil), and the response type description.

**a) Grupo 1**

**Pergunta 1:** É possível identificar de modo prático, nos cenários listados anteriormente, todos os principais elementos empregados em uma operação do domínio militar (Ex.: composição de um pelotão ou GC ).

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente)

Objetivo: Verificar o grau de aderência e compatibilidade do meta-modelo com os cenários do domínio, de acordo com a visão de seus especialistas.

**Pergunta 2:** Caso não concorde totalmente (5) com a afirmação anterior, explique.

Tipo de resposta: Texto livre

Objetivo: Permitir ao especialista exprimir as motivações da resposta anterior de modo a identificarmos oportunidades de aperfeiçoamento e/ou ajustes.

**Pergunta 3:** Quais elementos de um cenário real, o(a) senhor(a) acha que não é possível identificar ou inserir nos cenários anteriores?

Tipo de resposta: Texto livre

Objetivo: Permitir ao especialista listar as deficiências do meta-modelo em representar os cenários reais.

**Pergunta 4:** É possível representar facilmente as frações de Organizações Militares em operação, sendo empregadas no cumprimento de missão, utilizando a estrutura empregada para representar os cenários apresentados anteriormente.

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente)
Objetivo: Obter do especialista do domínio, a percepção do nível de facilidade de uso do meta-modelo para representar elementos do domínio.

**Pergunta 5:** Caso não concorde totalmente (5) com a afirmação anterior, explique.
Tipo de resposta: Texto livre
Objetivo: Permitir ao especialista do domínio fornecer elementos que auxiliem na melhoria da facilidade de uso do meta-modelo para representar cenários do domínio.

**Grupo 2**

**Pergunta 1:** As operações militares reais estão sujeitas a mudanças de condições: mudanças em partes da missão, mudanças na disponibilidade de recursos, mudanças nas condições climáticas, etc. Baseado nisso, o(a) senhor(a) concorda que a representação utilizada nos cenários mostrados acima consegue tratar as possíveis diferentes situações. Ou seja, consegue-se mostrar todas as situações de planejamento prévio.
Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente)
Objetivo: Verificar se o respondente tem a percepção de que o modelo carrega os conceitos de agilidade de C2.

**Pergunta 2:** Caso não concorde totalmente (5) com a afirmação anterior, explique.
Tipo de resposta: Texto livre.
Objetivo: Captar do respondente o que estaria faltando ao modelo para representar os elementos necessários à agilidade de C2.

**Pergunta 3:** A representação empregada nos cenários apresentados pode ser utilizada, no domínio militar, como único modelo para representação das informações básicas sobre uma operação militar (realizada em algum nível: Pel, Cia, Bda, etc) e o emprego dos principais elementos de manobra.
Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente).
Objetivo: Verificar junto a especialistas do domínio, se o modelo apresentado é suficiente como única forma de uso para representar um cenário de C2.

**Pergunta 4:** Caso não concorde totalmente (5) com a afirmação anterior, explique.
Tipo de resposta: Texto livre.
Objetivo: Permitir que os especialistas do domínio possam sugerir elementos que complementam o modelo de modo a ser possível seu uso na modelagem de um cenário de C2.

**Pergunta 5:** A representação utilizada nos cenários apresentados pode ser utilizada em complemento aos modelos e diagramas utilizados no domínio militar, complementando as informações necessárias sobre a operação militar.

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente).

Objetivo: Permite que o especialista do domínio expresse a compatibilidade do modelo proposto com o que atualemtne é utilizado pela Organização à qual ele faz parte.

**Pergunta 6:** Caso não concorde totalmente (5) com a afirmação anterior, explique.

Tipo de resposta: Texto livre.

Objetivo: Identificar se, para o domínio, existem outros elementos fundamentais que tornam o modelo proposto compatível com os seus cenários reais.

**Pergunta 7:** A visualização apresentada nos cenários citados facilita a compreensão das condições e do emprego de entidades/recursos no cumprimento da missão.

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente).

Objetivo: Identificar a percepção do especialista do domínio sobre a facilidade de uso e entendimento do modelo proposto.

**Pergunta 8:** Caso não concorde totalmente (5) com a afirmação anterior, explique.

Tipo de resposta: Texto livre.

Objetivo: Verificar os elementos que tornam difícil o entendimento do modelo proposto.

**Pergunta 9:** O Sr(a) visualiza cenários similares ao que foi mostrado, sendo executados/operados por sua Unidade/Organização em alguma missão desempenhada por ela.

Tipo de resposta: Escala de 1(discordo totalmente) a 5 (concordo totalmente).

Objetivo: Identificar se os cenários utilizados como casos de teste, são compatíveis com cenários reais vividos pela Organização.

**Pergunta 10:** Caso não concorde totalmente (5) com a afirmação anterior, explique.

Tipo de resposta: Texto livre.

Objetivo: Permite identificar detalhes sobre a dissonância do modelo proposto em relacão aos cenários reais vividos pela Organização do respondente.

In addition to the questions above, we have three other questions aimed at identifying the respondent's level of experience, as well as opening a field to receive suggestions, criticisms, and comments. These questions are the following:

**Pergunta 1:** Quanto tempo de carreira militar o(a) senhor(a) possui?

Tipo de resposta: 3 opções distribuídas em: menos de 15 anos, entre 15 e 30 anos, e mais de 30 anos.

Objetivo: Buscamos identificar o nível de experiência dos especialistas baseado no tempo como integrante da Organização.

**Pergunta 2:** Do tempo na carreira militar, qual é sua experiência em função de comando, chefia, administração ou gerência em quaisquer níveis?

Tipo de resposta: 4 opções distribuídas em: até 5 anos, entre 5 e 10 anos, mais de 10 anos, e não possuo experiência.

Objetivo: Permite identificar se o respondente atuou, e durante quanto tempo, em funções que exigem a coordenação de meios e permite a aplicação dos conceitos relacionados a C2.

**Pergunta 3:** Caso o(a) senhor(a) tenha algum comentário, crítica ou sugestão sobre este questionário, use o espaço abaixo para registrar:

Tipo de resposta: texto livre.

Objetivo: Permitir um canal de recebimento de sugestões, críticas e comentários sobre o questionário de modo a identificarmos oportunidades de melhorias e ajustes.

# Glossary

**C2 Agility** Capability of C2 to successfully effect, cope with, and/or exploit changes in circumstances. It is the composition of C2 Approach Agility and C2 Maneuver Agility.

**C2 Approach** It's the combination of how decision rights are allocated, how entities interact, and how information is distributed.

**C2 Maneuver** The ability to maneuver in the C2 approach space, adjusting one or more C2 dimensions (PoI, DoI, ADR).

**C2 Strategy** It is the same of C2 Approach and defines a portion within the C2 Approach Space.

**context** Involves the circumstances, state of the entities and the environment.

**Endeavor Space** It's the set of all predictable circumstances and conditions of a mission.

**FIFO** A method for organizing the manipulation of a data structure, where the first in is the first out.

**Grounded Theory** The systematic generation of theory from data analyzed by a rigorous research method.