Applying Autonomy to Distributed Satellite Systems: trends, challenges and future prospects

Carles Araguz*, Elisenda Bou-Balust, Eduard Alarcón

Nano-Satellite and Payload Laboratory,

Technical University of Catalonia – UPC BarcelonaTech

ABSTRACT

While monolithic satellite missions still pose significant advantages in terms of accuracy and operations, novel distributed architectures are promising improved flexibility, responsiveness and adaptability to structural and functional changes. Large satellite swarms, opportunistic satellite networks or heterogeneous constellations hybridizing small-spacecraft nodes with high-performance satellites are becoming feasible and advantageous alternatives requiring the adoption of new operation paradigms that enhance their autonomy. While autonomy is a notion that is gaining acceptance in monolithic satellite missions, it can also be deemed an integral characteristic in Distributed Satellite Systems. In this context, this paper focuses on the motivations for system-level autonomy in DSS and justifies its need as an enabler of system qualities. Autonomy is also presented as a necessary feature to bring new distributed Earth observation functions (which require coordination and collaboration mechanisms) and to allow for novel structural functions (e.g. opportunistic coalitions, exchange of resources or in-orbit data services).

Mission Planning and Scheduling frameworks (MPS) are then presented as a key component to

*Corresponding author. E-mail addresses: {carles.araguz, elisenda.bou, eduard.alarcon}@upc.edu

implement autonomous operations in satellite missions. An exhaustive knowledge classification explores the design aspects of MPS for DSS, and conceptually groups them into: components and organizational paradigms; problem modeling and representation; optimization techniques and metaheuristics; execution and runtime characteristics; and the notions of tasks, resources and constraints. This paper concludes by proposing future strands of work devoted to study the tradeoffs of autonomy in large-scale, highly dynamic and heterogeneous networks through frameworks that consider some of the limitations of small spacecraft technologies.

Key words: autonomy; mission planning and scheduling; distributed satellite systems.

---

1 Introduction

While transforming our planet, climate change is calling for more accurate, reliable and frequent data to study meteorological phenomena, monitor and predict natural disasters or estimate fauna migratory movements. At the same time, the ever increasing human exploitation of natural resources, transportation services and massive agricultural activities are demanding remotely sensed data to monitor deforestation, study sea-ice coverage or watch crops for weather damage. Even for humanitarian aid, governments and NGO's are unveiling the need to monitor oceans and conflict zones in near-real-time. As a result, some satellite-based missions are facing an architectural change of paradigm driven by the needs of these new Earth observation, security and surveillance applications. Distributed Satellite Systems (DSS) encompassing several interacting spacecraft are leading the way in applications where monolithic satellites have become obsolete in terms of risk, cost or even performance.

Next-generation satellite systems are envisioned as dynamically-networked formations of heterogeneous satellites designed to satisfy ambitious revisit times, cover large areas with higher resolutions or minimize data access latencies. These novel architectures leverage on concepts and technologies that have been widely explored for on-ground applications, namely, Wireless Sensor Networks, the Internet of Things, multi-core computing or nowadays' pervasive cloud services. As a result, satellite nodes are not conceived as independent instrument-centric components, but are rather expected to provide data services to their infrastructure, to engage in collaborative endeavors or even to wirelessly exchange resources among them. Certainly, some of these concepts are still technologically immature and present challenges both at the implementation, design and operation levels. This technological and architectural landscape is being complemented by the growing interest in small satellite platforms and their adoption in new designs. Given their cost-effectiveness in launch and development, small-spacecraft could be technological enablers in the deployment of practical DSS. With the advent of highly-miniaturized systems and ultra-low-power technologies, mini-, micro- and nano-satellites have become suitable platforms to implement Earth observation missions with multiple observing nodes. At the same time the use of these technologies imposes limitations on spacecraft, mostly in their communication, computation and attitude control capabilities. Nevertheless, audacious ventures are already demonstrating their suitability (e.g. *Planet Labs*' constellation of 120 hyperspectral nano-satellite imagers or *Sky and Space Global*'s communication constellation based on 3U CubeSat with inter-satellite link capabilities.)

Designing systems with multiple interacting satellites, however, could be deemed as a non-revolutionary approach. Notwithstanding that systems with multiple interacting satellites such as Globalstar or Iridium have already been in orbit for two decades, the mission requirements, functions and inter-satellite interactions suggested in novel DSS concepts pose the need for novel operation concepts capable of handling the complexity, heterogeneity and dynamism of next-generation architectures. In this context, this paper explores the need to enable autonomous DSS missions in order to address the latter aspects.

Briefly presenting their characteristics and summarizing different instances of DSS in Section 2, this paper lists some relevant features that need be designed from an operations perspective (Section

2.1.) After that, Section 3 discusses the need for autonomy; first in monolithic satellite missions and then emphasizing the need for autonomy in DSS (Section 3.1.) Section 4 devotes special efforts to scrutinize the design and runtime characteristics of Mission Planning and Scheduling frameworks by exploring previous works and classifying them with regards to: the system characteristics and organizational paradigm (Section 4.1); their problem modeling and representation (Sections 4.2, 4.3 and 4.6); the optimization approach and common metaheuristics (Sections 4.4 and 4.6); the runtime and execution characteristics (Section 4.5); the notions of a task (Section 4.7); the commonly modeled resources and constraints (Section 4.8); and, finally, some remarks on network characteristics (Section 4.9). Finally, the paper concludes in section 5 by emphasizing some of the less explored scenarios in DSS and suggesting future studies related to the application of autonomy in this context.

2 Distributed Satellite Systems

Usually conceived as complementary space assets, DSS are defined as mission architectures consisting of multiple space elements that interact, cooperate and communicate with each other. The term comprises many architectural approaches, namely, constellations (Figure 1a), satellite trains (Figure 1b), clusters or formations (Figure 2a), satellite swarms (Figure 2b) and innovative mission concepts such as fractionated spacecraft [1] and Federated Satellite Systems [2] (Figures 3a and 3b, respectively). Their differences and detailed taxonomy can be found in [3]. Because distributed satellite systems are not only expected to be structurally and functionally disaggregated but are envisioned as interconnected systems-of-systems where the nodes cooperatively achieve common goals, it should not be surprising to find classification approaches concerned about node inter-dependencies and the information exchanged between them. In an approach to study their fundamental commonalities and elaborate a comprehensive taxonomy, Lluch and Golkar have classified them in terms of: spatial distribution of their nodes (or the so-called *degree of fractionation*); the dynamic nature of their network and physical structure (i.e. static, dynamic, opportunistic); their orbit configuration; their inter-satellite communication and coordination capabilities; and, finally, the type of goals they fulfill (i.e. shared with the rest, individual). In spite of the relevance of the above-mentioned taxonomy, such classification approach can be considered crucial to understand distributed satellite systems because it addresses the many different characteristics and qualities of DSS architectures and provides a broad view to their fundamental design challenges.

2.1 New mission functions and requirements

Some practical examples of DSS have already been demonstrated, namely: communications constellations (e.g. IRIDIUM, GlobalStar, Orbcomm); global navigation satellite systems (e.g GPS, Galileo, GLONASS, Beidou); trains [4, 5]; and clusters [6, 7]. Nonetheless, practical examples of fully-fledged DSS which demonstrate systemic improvements and bring new functionalities have yet not been reported. In order to understand their design challenges and be able to address their operational requirements later in this paper, this section briefly discusses their expected system qualities and potential new functions.

2.1.1 Resilient, flexible and highly-adaptable systems

Owing to their distributed nature, DSS may represent an advantageous alternative in terms of robustness and system availability. Decentralizing their functionalities can eliminate single points of failure and enable the possibility to provide acceptable levels of service upon node failures. If the necessary internal mechanisms to react to failures are provided, the services and functions of the failing nodes could be absorbed by the system and would endow architectures with enhanced resiliency [8]. Similarly, their functional and structural segmentation also allows to replace or add new nodes, either when their on-board technology becomes deprecated by new advances, when new functions need to be added to the system or simply to overcome permanent failures. Their responsiveness and adaptability is also enabled by incremental developments or deployment strategies. In contrast to monolithic spacecraft, the financial outlay of distributed architectures could also be split in several phases, making them affordable options when costs can not be assumed all at once. By means of gradually adding new nodes to a DSS, the requirements of its associated data products would be satisfied in a series of stages or with incremental performances. Adding new functionalities to an already deployed DSS could also complement their flexibility by allowing these systems to adapt to a changing context or user needs [8]. Completing the set of qualities shared with many other distributed systems, satellite missions with several failing spacecraft would also degrade gracefully and might still deliver data even when most of its nodes would not be functional.

2.1.2 Resource exchange and collaboration

The literature describing fractionated spacecraft and Federated Satellite Systems (FSS) frequently discusses the exchange of resources as an essential function in these DSS approaches [2, 9]. In either of the two propositions, satellite networks or satellite fractions are expected to exchange data through inter-satellite links (ISL) to either coordinate among themselves or to provide services to such a distributed infrastructure. This fact would also be true for satellite swarms or constellations where nodes interact to one another. In-orbit data services could comprehend data processing (e.g. compression, analysis, on-board data fusion) temporal storage, transportation to other points of the network or the data relay to ground. FSS concepts envision the exploitation of under-utilized spaceborne commodities (e.g. processing power, or bandwidth) as a means to increase the value of already flying assets. This translates to the ability to collaborate with other systems by communicating with them and supplying or consuming some of their on-board resources. Even the bolder capability of wirelessly exchanging power is seen as a potential internal function of some DSS implementations [10].

2.1.3 Distributed Earth observation

Aside from the previous qualitative attributes and structural functions, distributed satellite systems open the door to alternative observation approaches that could improve mission performances. On the one hand, simultaneous acquisition of data from multiple geographical locations would improve temporal resolutions (i.e. *revisit times.*) In relation to this, architectures where nodes can download their data indirectly (e.g. through ISL networks) could also reduce data-access latency and achieve near-real-time monitoring of large, if not global, target areas. At the same time, increasing the number of observing satellites can also attain wider coverage at high resolutions without the need of wider swaths. A practical example of the latter is the large constellation of multi-spectral imagers *PlanetScope*, briefly introduced in [11]. Developed and operated by *Planet*, this constellation of about 120 3U nano-satellites will image the entire Earth every day with high spatial resolutions once it enters normal imaging operation.

In addition to the improvement in the temporal, spatial and spectral domain, DSS can also bring improved angular resolutions by capturing data from multiple spacecraft, either simultaneously or with shorter delays between the acquisitions. Several Earth observation applications leverage on multi-angular observations, such as hydrological modeling, soil mapping or the study of terrain stability, to name a few. The latter are based upon Digital Elevation Modeling techniques, which are commonly implemented with interferometric synthetic aperture radars (InSAR). By correlating images from two different satellite passes over the same area, Earth's topography and ground surface deformations can be mapped. This type of measurement, generally performed in repeat-passes by a single satellite, could also be performed by several spacecraft in distributed architectures. In that case, the time between consecutive acquisitions could be lower if the delay between the two passes is shorter than that of a repeat-pass. This would also improve the mission performance as it would minimize the likelihood of different atmospheric conditions (i.e. humidity), which are known to be strong limiting factor for this technology [12]. While InSAR techniques require extremely accurate orbit determination and precise pointing, stereo photogrammetry is much more flexible and could also be accomplished by DSS with less accurate pointing capabilities. As a matter of fact, stereo photogrammetry could even be achieved on an opportunistic manner. Lower resolution elevation maps can be obtained by combining optical imagery taken from different viewing angles, as in [13]. Although photogrammetry techniques may require more than two images to converge, this type of measurement could be performed by coordinating different satellite nodes and could also improve revisit times. Finally, capturing data from different viewing angles is also essential in mapping applications that rely upon Bidirectional Reflectance Distribution Function (BDRF) data products. BDRF needs to be estimated in order to capture the multiangular reflectance of opaque surfaces. This measurement allows to study land surface albedo [14], one of the most important parameters characterizing the Earth's radiative regime and its impact on biospheric and climatic processes. The design of distributed satellite missions that provide this functionality has been recently explored in [15, 16] where the authors proposed to address their design with formations of 6U-CubeSat and proved their feasibility and potential performance improvement with respect to monolithic alternatives.

2.1.4 On-board triggering of new observational requests in DSS

Enabling on-board data processing may allow fast detection of areas of interest (e.g. geographical changes, natural disasters or even new human constructions). While this is true and has been done in monolithic missions (e.g. EO-1 [17], IPEX [18]), distributed systems designed to scan large target areas with wide-swath, coarse-resolution instruments could identify interest zones and trigger localized observational requests. These secondary requests could be served by other nodes in the architecture, which could potentially embark higher resolution or specialized instruments as illustrated in Figure 4. This approach would efficiently use the resources of high-resolution spacecraft and could provide high-quality data to users with shorter latencies, thus improving the system responsiveness in cases where having near-real-time updates is critical (e.g. disaster monitoring.)

2.1.5 Based on small-spacecraft technologies

Agencies, universities and corporations have proved that micro- and nano-satellites are cost-effective platforms that reduce development times and present moderate performances. Their technologies, capabilities and suitability has been explored in several relevant survey papers [19–22], concluding that these types of platforms are capable of delivering data with scientific value. Most authors have pointed out that despite power, size, and mass constraints, small spacecraft have turned into extremely low-cost remote sensing platforms. Notwithstanding the fact that many of the high-performance instruments (e.g. SAR, lidar, high-resolution optical imagers, hyperspectral imagers, etc.) are generally not compatible with small spacecraft constraints, they can address a broad variety of measurements with high societal and scientific return [19]. In that sense, architectures based upon small spacecraft could provide very high temporal resolutions at moderately high spatial resolutions.

Small spacecraft are key enablers for distributed architectures where multiple spacecraft are required to attain global coverage. They have also been deemed essential when the system needs enhanced spatial, spectral or temporal resolutions at a reduced costs and risk [19, 23–26]. Innovative projects such as NASA's Cyclone Global Navigation Satellite System [27] are examples of how agencies are currently changing their paradigm and are already exploring solutions based on many small satellites to address new scientific needs. Simultaneously, relevant initiatives like the European QB50 project, aimed at studying the middle and lower thermosphere [28], are also exploring the feasibility of large, heterogeneous, CubeSat-based clusters.

3 Autonomy

Systems with ever increasing degrees of autonomy are more prevalent and important today than ever before (e.g. unmanned aerial vehicles, autonomous underwater vehicles, smart buildings, smart cars...) However, *autonomy* per se is a concept positioned at the intersection of philosophical, scientific and engineering interests. Autonomous systems are commonly defined as systems capable of generating their own laws or norms, but their modelling and design approaches are motivated by different perspectives and interests, as so it is their ultimate purpose.

Providing autonomous capabilities to monolithic spacecraft has been a concept of growing interest for more than one decade. Known as one of the first approaches towards autonomous operations, NASA's Deep Space One (DS-1) included the Remote Agent eXperiment (RAX), a software system aimed at managing the spacecraft's resources and activities [29]. This mission triggered new developments that tackled autonomous management at several different areas, including rover navigation, GNC, monitoring and FDIR, signal processing, and mission planning and scheduling. Traditionally, autonomous satellite technologies have been explored in order to address the following issues:

- *Communication delays:* Space exploration missions where the spacecraft is at huge distances from Earth (e.g. Mars exploration), present delays in the communication subsystem that precludes ground operators to manage the spacecraft with agility. Not being able to react to unexpected situations may lead to catastrophic effects for the mission and hence has triggered the development of autonomous controllers that can detect and correct failures at subsystem and mission level [30].
- *Reduced visibility:* Aligned with the previous issue, spacecraft orbiting in Low Earth Orbits (LEO) can only communicate with ground operators when they have visibility to ground stations. Therefore, there are large periods of time during which the spacecraft needs to operate autonomously.
- *Improvement of science return:* In addition to correcting faults, autonomous satellites can present improved performances provided that they generate their own mission plans. NASA's Earth Observing One (EO-1) was able to autonomously detect and respond to dynamic, scientifically interesting events observed from its Low Earth Orbit [17]. By combining on-board data processing algorithms with its Continuous Activity Scheduling Planning Execution and Replanning (CASPER), EO-1 was able to plan observation activities focusing on autonomously detected interesting regions.
- *Mission robustness and tolerance against failures:* Reacting to faults or ensuring the consis-

tency of the plan computed on ground is critical and has to be done without supervision (i.e. ensuring that the values predicted with models are within the expected ranges, checking that no system constraint is violated...) Moreover, when a given inconsistency is found, the satellite needs to re-calculate the plan in order not to miss any observation opportunity.

A thorough review of autonomous monolithic satellite missions can be read in [31], where the authors explore autonomy as an on-board functionality that can address both intelligent sensing, mission planning and execution, fault management and distributed decision making. Their work suggests to take the definition of autonomy levels proposed in the European standard ECSS-E-ST-70-11C [32] as a starting reference, and refer to the recent progress in Fault Detection, Isolation and Recovery techniques (FDIR), On-Board Control Procedures (OBCP) in European missions and autonomous spacecraft reconfiguration schemes based on Markov Decision Processes.

3.1 Autonomy in DSS

Besides the benefits studied for monolithic satellite architectures, autonomy is also considered a key enabler for distributed missions. As suggested in [31, 33], autonomy is key to operate spacecraft under uncertainty and to cope with unexpected situations. Vamvoudakis et al. also stated that full autonomy enables mission tailoring, control reconfigurability to allow for safe recovery, improved responsiveness and agility, and a general adaptability to changing environmental conditions [34]. Nevertheless, autonomy in DSS goes beyond local management of failures and generation of plans of action and needs be rather seen as an enabler for the system qualities and function discussed in Section 2.1. In this sense, Iacopino and Palmer enumerated up to seven motivations to autonomy that can help to visualize the broad scope of this matter [35]. While some of their motivations are very much related to the issues mentioned above, the rest describe additional values and qualities that have been scarcely tackled in other works, namely: decrease of operational costs; optimal management of resources to reduce inactivity periods; responsiveness; flexibility and adaptability.

This comprehensive notion of system-level autonomy is central to achieve the envisioned architectural functions of DSS, and needs to encompass: the adaptation to context and internal changes; distributed coordination and decentralized decision making; trading or exchange of resources; and optimal management of infrastructure capacities. Moreover, it is critical that the design of autonomous distributed satellite missions does not disregard two important notions of DSS, namely: their *dynamic nature* and their *complexity*. The dynamism not only refers to internal failures, but also tries to describe fundamental features of these adaptive architectures, such as incremental deployments, replacement of nodes or the creation of opportunistic coalitions. On the other hand, Distributed Satellite Systems can also be regarded as complex systems-of-systems by virtue of three dominating factors: heterogeneity; their potentially large dimensionality; and the influence of intermittent interactions characterized by orbital trajectories. DSS heterogeneity can be understood in many different ways. Some DSS studies are considering systems that hybridize traditional satellite platforms with small spacecraft technologies [26]. Others, are implicitly asserting the design of heterogeneous systems either by proposing opportunistic coalitions among existing systems [2] or by considering segmented nodes in which each part implements a different functionality (i.e. fractionated spacecraft.) Similarly, other approaches are also considering constellations with multiple specialized satellites (i.e. with different functions or instruments) which orbit at different planes or orbital altitudes [36]. Furthermore, interactions between two nodes in a distributed satellite mission heavily depend upon their orbital parameters. So much so that some task allocation approaches are even taking advantage of their intermittent and predictable communication processes [37].

Therefore, autonomy is not conceived as an additional feature to implement but rather as the solution to operate such dynamic and complex systems; to achieve the envisioned high-level qualities and to allow new functions. Coping with their operability with traditional approaches could be considered impractical. Computing static plans of action on-ground and then distributing time-tagged commands to spacecraft may preclude the expected flexibility of DSS. Even from the point of view of some of their required functions, such as flight formation, autonomy is deemed an essential characteristic that, especially for large scale systems, may not be addressable with traditional approaches. Even responsive OBCP may be functionally too restrictive to achieve control of a large-scale distributed architecture in which nodes should interact to one another.

In this scenario, autonomy in DSS is being studied under many perspectives. One of such perspectives understands autonomy as an integral characteristic of a DSS, like in the ANTS project by NASA [38]. With the Autonomous Nano-Technology Swarm (ANTS), NASA is investigating the application of swarm intelligence and decentralized computing techniques to provide self-configuration, self-optimization, self-protecting and self-healing capabilities to their architecture of pico-satellites [39]. A different view is that of the achievement of autonomous functions, such as flight formation. Both for inter-satellite communications (e.g. optical links) and remote sensing techniques (e.g. bistatic radars), being able to maintain virtual flight structures is a critical enabler for DSS. Some formations have been able to automatically maintain and correct their separation and orbits [40], while previous missions like PRISMA [41] or the up-coming PROBA-3 [42] have presented important progresses in the area of autonomous structure build up. At the same time, some authors have addressed satellite systems that manage and exchange their resources, as in [43]. This work delved into the problem of resource exchange and collaboration between systems in FSS, an interesting topic that can be extended to multiple DSS instances. Such exchange of resources (e.g. power, data processing, downlink capacity) goes beyond the technological capability and comprehends the necessary coordination mechanisms to do so. While crucial to enable system-level autonomous operations, these mechanisms foster the agreement among spacecraft involved in resource transfers (i.e. the service consumer and producer) and coordinate the appropriate maneuvers to interface them. The coordination mechanisms are very much intertwined with the networking characteristics of DSS. The literature in ISL and Inter-Satellite Networks is extensive and has been looking at several aspects, such as: the limitations of small satellite platforms [44]; the adoption of ground Internet standards for space [45]; the design of new network protocols for specific DSS architectures [46]; or the exploration of delay-tolerant network concepts in Earth-imaging constellations [47]. Many networking characteristics may influence the design of coordination mechanisms in DSS, since these are the foundations of the decentralized interactions envisioned in many cases.

Likewise, the design of autonomous DSS, needs to cope with the technological constraints in small spacecraft technologies. Among their limitations, the impaired or extremely limited communications capabilities results in the most critical one for the context of this paper. Albeit recent studies have been able to develop optical downlinks for small satellites [48], the ISL capabilities of these platforms still require many advancements to allow the deployment of large and interconnected swarms or constellations. In contrast, the use of off-the-shelf components (COTS) in their development may confer them with computational capabilities slightly above their traditional counterparts. However, it is also true that their reduced power generation could preclude them from processing large volumes of data or executing of other algorithms that require high computing power.

In summary, the need for autonomy in DSS is driven by the factors presented herein. Autonomy is indeed essential to cope with problems that are present in monolithic satellites; to improve performance and tolerance to failures. Nonetheless, DSS are dynamic architectures where components with different capabilities and capacities interact to achieve global goals. Their operation must be reliable and their nodes need to optimally manage shared resources. Autonomy is hence seen as the cornerstone of their operability and an enabler for many of their qualities and proposed functions.

3.2 Designing autonomous systems

One common notion in the design of autonomous system is to understand them as systems capable of reasoning, deciding and executing their activities without human intervention, in pursuit of a given set of goals set by operators. In many cases, this very concept has led to describing the system in terms of a range of behaviors associated with its agents: their ability to sense the world, maintain their state and make decisions about their actions [33]. Modeling autonomous systems as Multi-Agent Systems (MAS) is a common trend in many industrial and academic works (including aerospace systems and satellite missions) and is applied when problems are either too large or when the information sources are spatially distributed. MAS frameworks can provide performance improvements to the system and allow complex functions [49, 50].

In line with this, works in artificial intelligence have studied and implemented bio-inspired MAS in which the entities present complex collective behaviours (e.g. self-organization) but where none of the entities was explicitly programmed or designed for such purpose [51]. Certainly, among all species of the animal kingdom, the behaviors of some insect families have fruitfully given birth to many works in the areas of swarm intelligence and collective organization. Collective behaviors are seen to emerge from direct or indirect interactions among entities and usually involve no sophisticated cognitive processes [52, 53]. Moreover, self-organized control inspired by biological systems has been receiving considerable attentions as a promising concept for realizing robustness, scalability, and adaptability [54]. Although entities' behaviors can be simple, it is through a series of non-linear interactions that their behavior outcomes are accumulated and scaled-up so that some collective behaviors or patterns emerge[1]. As a result, approaches that mimic some of their characteristics and mechanisms (e.g. stigmergy) have been exhaustively applied in robotics (e.g. [55–60]), task allocation [61] or optimization [62] and can also be adopted to implement autonomous systems. The following sections will present how some of these concepts and autonomous system paradigms have been applied to satellite missions and what are the necessary ingredients to ultimately achieve fully autonomous DSS.

3.3 From autonomous systems to Mission Planning and Scheduling

Either autonomous or orchestrated by human operators, decision making in satellite missions has been implemented and performed in Mission Planning and Scheduling systems. Satellite MPS software transform high-level requests and system constraints into a set of commands that satisfy the former without violating the latter. The literature is abundant with examples of MPS that

[1] The reader is directed to [51] for a comprehensive classification of collective behaviors as well as detailed study on the characteristics of self-organization in biological systems

present different design and execution characteristics. Still, all of them serve the same purpose: allocate tasks and control the actions of the spacecraft. In line with previous works (e.g. [31]), the authors of this paper propose that autonomous reasoning, node coordination and planning of activities has to be designed as part of the mission planning framework.

Despite the availability of commercial MPS solutions dedicated to Earth observation constellations, swarms or clusters, these products are not convenient for fully distributed satellite missions [63]. Current MPS designs provide automated scheduling and plan execution mechanisms that satisfy the requirements of traditional missions (i.e. allocate tasks and control system constraints). However, only a few of them has been designed to exploit some of the qualities of a DSS (e.g. [63–68]).

4 Mission Planning Systems

4.1 System, architectural and design characteristics

The noticeable growth in the number of Earth Observation missions has lead to the development of many Mission Planning Systems. In the literature, it is easy to find both commercial products (e.g. GMV's *flexplan* [69], DEIMOS's *plan4EO* [70]), and a myriad of academic approaches. While the former tend to include sophisticated models, detailed mission-specific constraints, security policies and multi-user access methods, the latter tend to focus on the design aspects, the algorithms employed and their performance. Because the literature is so abundant but at the same time so diverse, this section will start by identifying and categorizing their similarities from a system/mission perspective and from a software architecture design point of view.

4.1.1 Components and architecture

Whether executed on-board or on-ground, among the components of an MPS, it is very common to find a clear separation between the planning algorithm, the models that capture the dynamics of the system and the constraint checkers that ensure that there is no violation of safety conditions or resources (e.g. [17, 64, 71, 72]). Surrey Space Centre's NEAT algorithm, for instance, divides the planning software between an heuristic component that generates schedules with a decoupled resource allocator that validates the correctness of the generated plan of action and feeds the heuristic back with a fitness value [72]. In some cases, such as the scheduling software for EO-1 [17], the MPS also encompasses on-board data processing algorithms that analyse scientific data and provide inputs to the scheduler.

4.1.2 MPS for distributed missions

Since the appearance of the first Earth-observing constellations and distributed missions, researchers have focused on specialized MPS targeted for DSS [36, 65, 66, 68, 69, 71, 73–75]. Two remarkable examples of those are the MPS for the Sentinel-S1 constellation, presented in [69], and the one for TerraSAR-X/TanDEM-X, detailed in [66]. Although these satellite missions only encompass two spacecraft, the level of complexity in their models and constraints make them a practical illustration of automated distributed satellite operations. In addition, the fact that these two missions have been launched in the past and are operating in nominal conditions also demonstrates the feasibility of autonomous missions at some degree. One notable feature of the TerraSAR-X/TanDEM-X planning system is the fact that it was designed to schedule thousands of observational requests per day, without human supervision.

On the other hand, the planners presented in [36, 65, 68, 71, 74, 75] have been designed for complex satellite systems encompassing, in most of them, tens or hundreds of heterogeneous satellites in multiple configurations (i.e. GEO, LEO, several orbit planes, etc.) Because of that, these works tend to simplify the modelling of subsystems and instruments. Instead of considering the modes of operation that would be needed to operate very-high-resolution imagers, or complex attitude dynamics, these approaches focus on the complexity of the architecture and define simplified arbitrary components.

4.1.3 Multi-agent-based MPS

A recent study for the European Space Agency [50] has explored and validated the benefits of MAS technologies in the context of Earth Observation constellations. The Distributed Agents for Autonomy study (DAFA) demonstrated how MAS-oriented frameworks can improve several system qualities and presented negotiation-based self-organizing mechanisms as means to improve the autonomous capabilities of a system. Aligned with this vision, some works have addressed MPS for DSS that could be regarded as multi-agent systems [36, 65, 73–75]. Whereas traditional heuristic approaches are fine-tuned and optimized for a known number of tasks and satellites, multi-agent approaches allow for an adaptability and scalability that is not comparable [65]. The system proposed in [74], defines nine types of agents that allow to model as many elements of the architecture as the mission requires. The authors have designed three cooperative agents that interact to find a solution to the scheduling problem: the *satellite*, *request* and *mesh* agents. In addition, three active agents and three passive agents allow them to model observational constraints (cloud coverage, solar ephemeris, ground stations) and the system resources (battery, attitude, memory), respectively. This configuration allowed to achieve optimized results and to adapt to environmental circumstances and structural characteristics. Noteworthy, [73] presented a MPS that was tested against scalability, showing reasonable results for a constellation of 300 CubeSat-like satellites.

4.1.4 Bio-inspired, self-organizing MPS

Aside from heuristic and MAS-based designs, a few MPS have been influenced by swarm intelligence and collective organization schemes and have been based upon bio-inspired algorithms. On the one hand, Tripp and Palmer presented a mission planning scheme grounded on stigmergy [68]. In this simulation-based approach, the satellites were precluded of any direct communication capability (i.e. Inter-Satellite Links) and were forced to self-organize through indirect interactions. This lack of capability was owed to and justified by the inherent technological limitations of nano-satellites. A similar proposal was published later by Iacopino et al. [71]. In their work, the authors presented a mission planning system based on the Ant Colony Optimization metaheuristic, inspired by pheromone trails left by ants and termites and extensively employed in other optimization contexts. The proposal is instantiated in real scenarios that take the Disaster Monitoring Constellation as case scenario. They represent the planning problem as a binary chain where each task corresponds to the node of a graph (i.e. the colony) and satellites can choose whether to schedule the (potential) observation requests or not. Tasks are generated with the predicted ground tracks; each satellite has its own orbit and can propagate its position and determine which observation requests correspond to their future geographical location. This binary chain encoding not only allowed them to solve the problem as an ACO but also provided organic coordination mechanisms: tasks that present conflicts (e.g. because can be scheduled by two satellites) are modeled as shared nodes between two colonies. Furthermore, their encoding also allowed to define dynamic scenarios in which new observations are inserted in the binary chain at any given moment, in order to represent disaster events (e.g. fires, volcanic eruptions) or weather updates. While it is unclear whether their current approach could be executed on-board, the main drawback is that the binary encoding of observation requests and ACO algorithm does not account for other structural functions in DSS (e.g. exchange of resources, multi-angular observations.)

4.2 Problem division

Similarly to the architectural separation of components mentioned above, some works try to reduce the computational complexity of the problem by dividing it into several sub-problems or levels of definition. On the one hand, the approach presented in [76] has two separated modules with two different goals. The weekly module, executed on ground, takes advantage of the computational resources of large servers and generates weekly schedules 3-5 days in advance. These schedules are refined several times until they become available to the mission control center. A second part of the problem is solved on-board, where the spacecraft executes a replacement scheduling algorithm. Much more constrained than the weekly phase, the second stage takes the weekly schedule as an input and replaces it with a more accurate solution with a time horizon of only 8 hours. This system, implemented by researchers at NASA JPL, allowed to port their MPS to the IPEX CubeSat mission [77]. On the other hand, [75, 78] also present algorithms that split the problem into many sub-problems that are then recombined. This very idea is also implemented in [36, 79] where the problem is divided into a global part and a local part. Global problems take into account the whole constellation and all the observation requests, while local problems are those limited to one spacecraft. In both cases, at the time of solving the problem the global part of the algorithm is unaware of the actual state and resource capacities of the spacecraft. The difference between these two approaches lays in the algorithm used in the global part. The global algorithm in [36] is executed on ground with global, real-time information from all the observation requests and the environment. Once the observation requests are distributed among the satellites in the constellation, each satellite uses this plan as a recommendation and generates its own local schedule taking into account its resources. On the other hand, the approach presented in [79] is always executed in the space segment. A coordinating satellite receives the set of observation requests and distributes all of them to the rest. Then, local schedulers provide n sorted sub-solutions that are merged by the coordinating satellite with a combinatorial optimization algorithm [80].

4.3 Timeline representation

The modelling of a scheduling problem can be done in a variety of ways. One of such ways is using timeline representations. This approach is used in most mission planning systems for monolithic spacecraft (e.g. [64, 76, 78, 81]) as well as in some satellite constellation missions (e.g. [66] and [71][2]). Timeline-based modelling is grounded on the definition of each system variable in the time domain (i.e. resources, instrument modes, constraints, activities). Thus, the plan generated by the scheduling algorithm is composed of several timelines the values of which are generated during the planning process through system models that reflect their time dynamics as illustrated in Figure

[2] Their task chains are described by the authors as a *discrete* timeline representation.

5. This commonality was identified by Chien et al. in a survey paper published in 2012 that was devoted to automated mission operations [18]. There, they explored the state-of-the-art in MPS and highlighted the common types of constraints, resources, tasks and dependencies. Interestingly enough, task decomposition is commented as a common feature in most MPS. However, the decomposition of high-level activities into lower-level ones is referred to post- and pre-activities that need to be performed when a given instrument or subsystem has to be operated (e.g. thermal stabilization of an instrument). Hierarchical Task Networks (HTN) are a known and suitable method to represent and model such task decompositions. Noteworthy, the very same concept could be applied for distributed applications where complex and synchronized activities describe collaborative tasks[3]. Furthermore, [18] also notes that dependencies usually exist between actions and state variables (e.g. a SAR instrument should not be enabled until the satellite is in a safe attitude mode). While some of the matters discussed in their paper will be used as a reference in the following sections, the reader is directed to the survey for detailed analysis on the functionalities and characteristics presented by timeline-based MPS.

4.4 Optimization

Another aspect of MPS is whether they perform some kind of optimization and which are the optimized objectives. The algorithms used in [36, 69, 71, 72, 74, 81–83] all solve optimization problems, usually with multiple objectives. The Capacity Analysis and Mission Planning Tool developed by DEIMOS [81] calculates the most optimum sequence of observations and allows the end-user to decide the type of figure of merit that feeds the sequencer in order to fully suit the mission needs. Similarly, the MPS for the Sentinel-S1 constellation has been designed to optimize both the usage of the recording device and the ground station network usage [69].

Although there are very few planning systems not optimizing their outcomes, some MPS simply perform task distribution and try to optimize a given figure of merit with abstract constraints (e.g.

[3] It should be possible to include such type of interactions in the binary chains used in [71].

minimize repeated tasks). This kind of optimization is the one observed in [68, 75] and in the ground-segment process of [36]. Similarly, negotiation-based algorithms in multi-agent approaches, present optimization algorithms that are computed in a collective and decentralized manner (e.g. [36, 65, 74]).

4.5 Execution and runtime characteristics

4.5.1 On-board/on-ground

One of the abovementioned open questions when designing autonomy systems for satellites is whether their intelligence has to be placed on-board or on-ground. Traditionally, deep space missions have been commanded with On-Board Control Procedures (OBCP): automated programs or complex routines that are decomposed into time-tagged commands on-board, and which implement specific control actions. These actions need be executed without human supervision and generally encompass some kind of fault management to prevent loss of contact with the spacecraft [84]. OBCP are the solution to critically long propagation delays in which human operators can not operate their spacecraft reliably. For similar reasons, one could argue that missions with long propagation delays would also be inclined to issue autonomous decisions on-board. While this could be true for these cases, the solution can be debatable for Earth observation satellites that are frequently in contact with ground operators without delays.

Recent developments [30, 68, 77] leverage on the benefits exploited in previous on-board systems [36, 75, 76] and are designed to run in the space segment. These systems take advantage of having real updated data with which they can predict their schedule more accurately. On-board MPS for DSS can utilize inter-satellite communication capabilities (ISL) to perform truly decentralized and distributed mission planning, even without direct visibility with ground control centers. On the other hand, they are limited by the inevitable computational and network constraints. These limitations need also be considered along their lack of real-time operational goals. The system has its own variables updated in real-time and can react to internal and environmental changes with agility. However if a change in mission goals occurs, an on-board system needs this information to be propagated to all the nodes.

On the contrary, ground-based MPS [66, 69, 71, 73, 74, 81–83] can be executed in powerful computers but present a reduced responsiveness for similar reasons. One could say that both on-board and on-ground alternatives can be used to provide autonomy to the system, since its definition does not force the algorithm to be executed in any particular physical location. However, one could also argue that the adaptability claimed by MAS-based schemes is worsened as long as the system is executed in partially updated environments. The truth is that some developments tend to divide the system in two parts: one executed on-ground and the other on-board [36, 76]. In our point of view, there is a clear trade-off between the advantages and drawbacks presented by both on-board, on-ground and hybrid approaches.

4.5.2 Reactive control, deliberative reasoning and hybrid approaches

On-line mission planners are those which are constantly producing an output (i.e. the actions to be performed). Conversely, *off-line* schedulers are executed during a certain amount of time and generate a plan of actions to be performed until a given future time horizon (Figure 6). A similar distinction can also be made between schedulers that perform *reactive control* or *deliberative reasoning* about the task to perform. While the first ones react to inputs from the environment to perform the scheduling function (i.e. on-line), deliberative planners predict future states and plan the tasks accordingly (i.e. off-line). Traditionally, Mission Planning Systems have been deliberative [66, 78, 81–83]. The accuracy and feasibility of their solutions relies on the fidelity of their system models, which is proportional to their complexity. Because of that, some off-line MPS have reported to take up to 2 days of runtime to output a schedule [82]. On-line approaches [68, 72], present shorter execution times to produce a valid output but are unable to predict future states. This condition may prevent them from fully optimizing the operations, specially in a DSS context.

For this characteristic, hybrid approaches exist as well, yielding better results and balancing the benefits of both approaches [17, 36, 71, 78]. Upon the detection of an inconsistency, a system failure or the arrival of new science data (i.e. reactive behavior), the planners described in [17, 36, 64, 71] activate a deliberative algorithm which re-computes or modifies an existing schedule. The fact that the system is designed to modify an existing solution could be advantageous in terms of computational costs, given that the algorithm does not start with the same initial conditions. An interesting approach is that of [64], in which the system can operate solely with a reactive algorithm that produces instantaneous solutions based on system rules. Given the runtime required by a deliberative algorithm, the system is constantly controlled by the reactive module and receives deliberations in an asynchronous manner. Therefore, the deliberative layer only enhances the solutions and acts as an *advisor*.

The mission planners presented in [66, 78], could also be considered hybrid in these terms. In these works, the deliberative planner is executed periodically, allowing the algorithm to re-plan part of the spacecraft activities with updated system values. Such modes of operation normally present interleaved scheduling windows as shown in Figure 7.

4.6 Problem modeling, types of algorithms and metaheuristics

In general, mission planning problems can be formulated as multi-dimensional Knapsack problems like Equation (1) where X is a vector of decision variables $x_i \in \{0, 1\}\,, \quad i = 1, \ldots, n$ that indicate if request i has been scheduled ($x_i = 1$) or not ($x_i = 0$). Equation (1b) expresses arbitrary resource constraints. Most schedulers have to deal with m resources, yielding at least m constraints. Additionally, scheduling constraints that express relative timing or ordering of activities can be added [18]. Allen's interval algebra defines up to seven different time relations that could be applied in different cases (Table 1). Many MPS explicitly formulate the problems as a Knapsack [36, 71, 83, 85] which are, in fact, a subset of the NP-Hard Traveling Salesman Problem (TSP).

$$\text{Max.} \quad f(X) = \sum_{i=1}^{n} w_i x_i \tag{1a}$$

$$\text{Subject to:} \quad \sum_{i=1}^{n} r_i x_i \leqslant c \tag{1b}$$

Other characteristics in the way these problems are tackled are the algorithms used to solve them. Dynamic Programming has been used in Earth observation missions [36] as well as negotiation-based approaches [74, 75]. In [65] the authors implement a reinforcement learning strategy to solve this problem, implementing it with neural networks encoded in chromosome. However, the most common approach to solving these problems is to implement some kind of metaheuristic or combination of metaheuristics. While [71] uses Ant Colony Optimization, [86] combines ACO with a mechanism of Local Search. Classical metaheuristics like Local Search [17] or Greedy algorithms [64, 66] are also common when producing sub-optimal solutions is not critical. Tabu Search [87] and Simulated Annealing [88] have also been used for EO missions, although, above all methods, the most common ones are Genetic Algorithms [68, 72, 81–83, 89].

4.7 Tasks

Tasks (or *activities*) are the basic inputs for Mission Planning Systems. In EO applications, tasks are usually mapped to observational requests. They are represented as objects which, upon their execution, consume a given resource. As mentioned above, datatake activities can actually be decomposed into several tasks or activities [18] (linked with Allen's interval algebra relations, Table 1). Depending on the framework and mission, the tasks and their level of abstraction varies. Nonetheless, it is common to find tasks related to data acquisition, attitude control or data download (e.g. [17, 36, 72, 75, 81, 83]). Some works actually map each system state to a task, as in the DEIMOS-2 MPS [81], where the five high-level system modes of operation are mapped to tasks that can be repeated, namely: Sun-pointing for housekeeping, nadir-pointing, image acquisition, orbit maintenance and control, and download.

4.8 Resources and constraints

The diversity in the modelling of resources is almost as large as the diversity of EO applications. Resources are abstract objects that encapsulate a system constraint, usually represented as a capacity. Researchers at NASA JPL analysed them thoroughly [18] and described the following characteristics, which have been used in other works as well [72]:

1. *Atomic resources:* tasks or activities that can not overlap, will require resources with binary capacity. Thus if a task is scheduled for a given period of time, the atomic resource to which it is linked will have capacity 0 and will prevent other tasks from being scheduled at the same time.

2. *Depletable resources:* on the contrary, resources that can take an integer capacity, will be those which can be consumed by more than one activity at a time (if and only if the sum of task consumptions is lower or equal to the resource capacity). This resources are also called "consumable" in [72], where their capacity C_i is formulated as a normalized real number ($0 < C_i \leqslant 1$).

3. *Intrinsic maximum/minimum values:* Given that resources are, in some cases, representing physical variables, some of them will present intrinsic restrictions that have to be modelled accordingly. When modelling the resource *energy*, the designer has to take into account that the devices that store energy (e.g. batteries) can store up to a maximum amount of charge; when the state-of-charge of a battery reaches 100%, energy in excess will be lost.

Typically, satellite MPS encompass resource models to represent power, energy and memory (data storage) (e.g. [17, 36, 74, 75, 81, 83, 90]). Some missions have been seen to include complex models to represent thermal [90] or attitude [74, 81] as a resource.

Finally, most missions include temporal and spatial constraints to reduce the solution space of the problem and represent regions of unavailability. Complex operations constraints can include:

instrument and subsystem timing and synchronization; thermal; power; data volume; antenna visibility (i.e. line-of-sight and/or distance); spacecraft pointing; and priority. Among the common constraints that can be found in the literature, visibility with the ground station is one of the most common one. In addition, safety constraints are also defined to prevent damage to the spacecraft subsystems (e.g. direct Sun exposure [17], sensor mutual exposure [90]) to impose technological restrictions (e.g. multiple simultaneous downloads to the same ground station [90]) or to reflect environmental conditions (e.g. cloud coverage [83], sunlight [64]).

4.9 Network considerations

Very few Mission Planning Systems oriented to DSS consider network topologies, the dynamism influenced by orbit trajectories or their lossy nature (especially for missions that rely on small spacecraft technologies). Figure 8 tries to emphasize the fact that node interactions (i.e. transfer of data, coordination messages, etc.) happen in an intermittent manner. In addition to that, different nodes in the constellation may be characterized by different ISL bandwidths, ranges or even technologies (i.e. optical, RF bands). Nevertheless, many of the consulted references propose MPS that do not rely or consider these network features: they are executed on-ground and then send the solution to the network of satellites. A notable exception is the one published in [37, 75]. These researchers from the French CNES and ONERA propose a cooperative framework based on a negotiation approach that actually takes advantage of the fact that orbits can be considered quasi-periodic. With this primitive assumption, they designed a decentralized, on-board planning scheme where satellites predict future interactions based on orbit period information. At every satellite encounter, spacecraft exchange information in order to allow the negotiation scheme to converge to a feasible planning solution. They implemented an epidemic inter-satellite communication protocol (known to propagate a given information unit in $O(\log n)$ communication rounds) to allow for indirect communication between the satellites. Their simulations show how, on average, a satellite constellation with 3 satellites is able to absorb more than 70% of the requests in less than 48 hours (with the constellation initialized with 150 requests and two blocks of 200 requests triggered during the simulation). Nonetheless, their work does neither analyze the network overhead nor show the actual convergence time for this algorithm, making it difficult to evaluate the suitability for demanding contexts requiring short revisit times or deadlines. Moreover, despite centralized, the algorithm they presented assumes a fixed and known constellation, precluding dynamic or opportunistic concepts such as FSS.

Novel approaches such as the one in [68], propose hierarchical networks of indirectly-communicated satellites in order to provide a scale-free solution to the planning problem in huge constellations or satellite swarms. This idea, however, was not explored in their published work and does not seem to be developed in further publications of the authors.

Finally, it is interesting to mention that the MPS presented in [36] was validated for a complex network consisting of several ground stations, 3 GEO satellites to provide data rely capabilities and 12 LEO satellites performing Earth Observation datatakes. Interestingly, despite dating from 2005, this work seems to capture the network complexity of future distributed satellite systems, at least in a very primitive way (i.e. small constellation, unconstrained ISL, etc).

5 Conclusions

This paper has tried to lay out the potential benefits of autonomy in Distributed Satellite Systems by revisiting and structuring its traditional motivations and the current prospects. Autonomy is presented as an integral characteristic of DSS that could enable coordination of spacecraft in distributed Earth observation endeavors. Particularly, the need for autonomous DSS has been discussed in the context of innovative space systems; those that foresee re-usable, increasingly complex and large-scale systems, composed of interconnected spacecraft with heterogeneous capabilities. As such, autonomy is not only targeted at solving task scheduling problems but rather at becoming an all-pervasive operational scheme that can contribute to the achievement of many system qualities.

In line with this, the paper highlighted that operating DSS autonomously could improve responsiveness of the system, and even bring about other qualities like resiliency, adaptability and in-orbit exchange of resources.

Autonomous operations can be designed as part of next-generation Mission Planning and Scheduling frameworks. Accordingly, the paper devoted several sections to present an exhaustive knowledge classification that has aimed at identifying the architectural, modeling and execution characteristics of current MPS solutions. Among the explored characteristics of MPS, several coordination and organization approaches towards increasing autonomy have been presented, namely: bio-inspired and self-organizing systems; MAS-based designs; and MPS that generate mission plans in batch. Problem division, the type and modeling of resources, and the timeline representation have been presented as some of the design traits of multiple MPS. Additionally, their runtime characteristics have been classified into deliberative, reactive, and hybrid approaches. With the former being common for MPS that generate mission plans in batch, reactive decision making has been found to be one of the commonalities of negotiation-based and bio-inspired solutions. Similarly, this classification has also attempted to distinguish the segmentation between on-board and on-ground mission planning designs. Moreover, the paper has also explored many of the metaheuristic families used in the solving of DSS scheduling problems when these are formulated as a multi-dimensional Knapsack problem.

Albeit some works have addressed autonomous operations in regards to flight-formation, failure detection and mitigation, or the generation of observational requests (through processing of on-board data), achieving autonomous operations at all levels is still an open field of research. Several fundamental questions remain open and should be addressed in future works. Undoubtedly thought-provoking, one of such questions continues to revolve around what is the most appropriate organizational paradigm to implement autonomous DSS (i.e. multi-agent-based, bio-inspired, based on heuristics and optimization techniques, based on negotiation algorithms, hybrid constitution.) Related to that, future studies may also inquire further into the actual tradeoffs of decentralized on-board decision making in contrast to centralized on-ground alternatives. Furthermore, attaining the system qualities of a DSS through autonomous operations has still not been demonstrated. Future research is expected to tackle these aspects, which may also comprehend the ability to provide in-orbit data services or, more generally put, the exchange of on-board resources. Assessing the impact of small spacecraft technologies and their limitations in regards to autonomous operations is equally pivotal. In particular, new satellite coordination mechanisms need to circumvent the constraints enforced by the low datarate inter-satellite links inherent to small satellite designs. Ultimately, it is important to highlight the need to put all these questions in the context of disruptive DSS designs: large-scale, dynamic networks in which heterogeneous satellites cooperatively perform distributed Earth observation.

6 Acknowledgments

The work presented in this paper has been carried out in the frame of the Operational Network of Individual Observation Nodes (ONION) project, and has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement 687490, coordinated by Thales Alenia Space France.

References

[1] Owen Brown and Paul Eremenko. "Application of value-centric design to space architectures: the case of fractionated spacecraft." In: *AIAA SPACE Conferences and Exposition*. 2008.

[2] Alessandro Golkar and Ignasi Lluch. "The Federated Satellite Systems paradigm: Concept and business case evaluation." In: *Acta Astronautica* 111 (2015), pp. 230–248.

[3] Ignasi Lluch and Alessandro Golkar. "The Value Proposition of Novel Distributed Space Systems: A Literature Survey on Ilities." In: *3^{rd} Federated Satellite Systems Workshop*. Ithaca, NY, 2015.

[4] Maria Daniela Graziano. "Overview of distributed missions." In: *Distributed Space Missions for Earth System Monitoring*. Ed. by Marco D'Errico. Vol. 31. Springer Science & Business Media, 2012. Chap. 12, pp. 375–386.

[5] M. R. Schoeberl. "The afternoon constellation: a formation of Earth observing systems for the atmosphere and hydrosphere." In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*. Vol. 1. 2002, pp. 354–356.

[6] Manfred Zink, Gerhard Krieger, Hauke Fiedler, Irena Hajnsek, and Alberto Moreira. "The TanDEM-X mission concept." In: *Proceedings of the 7^{th} European Conference on Synthetic Aperture Radar*. 2008, pp. 1–4.

[7] Sebastián Muñoz, Richard W. Hornbuckle, and E. Glenn Lightsey. "FASTRAC Early Flight Results." In: *Journal of Small Satellites* 1.2 (2012).

[8] Daniel Hastings, Paul La Tour, and Benjamin L. Putbrese. "Value-Driven Analysis of New Paradigms in Space Architectures: An Ilities-Based Approach." In: *AIAA SPACE Conferences and Exposition*. 2014.

[9] Owen Brown and Paul Eremenko. "The value proposition for fractionated space architectures." In: *SPACE Conferences and Exposition*. 2006, pp. 1–22.

[10] Allison K. Porter, Dustin J. Alinger, Raymond J. Sedwick, John Merk, Roedolph A. Opperman, Alexander Buck, Gregory Eslinger, Peter Fisher, David W. Miller, and Elisenda Bou. "Demonstration of Electromagnetic Formation Flight and Wireless Power Transfer." In: *Journal of Spacecraft and Rockets* 51.6 (2014), pp. 1914–1923.

[11] Christopher Boshuizen, James Mason, Pete Klupar, and Shannon Spanhake. "Results from the planet labs flock constellation." In: *28th Anual AIAA/USU Conference on Small Satellite*. 2014.

[12] Xiao-li Ding, Zhi-wei Li, Jian-jun Zhu, Guang-cai Feng, and Jiang-ping Long. "Atmospheric effects on InSAR measurements and their mitigation." In: *Sensors* 8.9 (2008), pp. 5426–5448.

[13] Frederik Tack, Rudi Goossens, and Gurcan Buyuksalih. "Assessment of a Photogrammetric Approach for Urban DSM Extraction from Tri-Stereoscopic Satellite Imagery." In: *The Photogrammetric Record* 27.139 (2012), pp. 293–310.

[14] Wolfgang Lucht, Crystal Barker Schaaf, and Alan H. Strahler. "An Algorithm for the Retrieval of Albedo from Space Using Semiempirical BRDF Models." In: 38 (2 2000), pp. 977–998. DOI: 10.1109/36.841980.

[15] Sreeja Nag, Charles Gatebe, and Thomas Hilker. "Simulation of Multiangular Remote Sensing Products Using Small Satellite Formations." In: *IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing* 10 (2 2017), pp. 638–653. DOI: 10.1109/JSTARS.2016.2570683.

[16] Sreeja Nag, Charles K Gatebe, David W. Miller, and Oliver de Weck. "Effect of satellite formations and imaging modes on global albedo estimation." In: *Acta Astronautica* 126 (2016), pp. 77–97. DOI: 10.1016/j.actaastro.2016.04.004.

[17] Steve Chien, Rob Sherwood, Daniel Tran, Benjamin Cichy, Gregg Rabideau, Rebecca Castano, Ashley Davis, Dan Mandl, Bruce Trout, Shulman Seth, and Darrell Boyer. "Using Autonomy Flight Software to Improve Science Return on Earth Observing One." In: *Journal of Aerospace Computing, Information, and Communication* 2 (4 2005), pp. 196–216. DOI: 10.2514/1.12923.

[18] Steve Chien, Mark Johnston, Jeremy Frank, Mark Giuliano, Alicia Kavelaars, Christoph Lenzen, Nicola Policella, and Gerald Verfailie. "A generalized timeline representation, ser-

vices, and interface for automating space mission operations." In: *Space Operations Symposium*. Stockholm, Sweden, 2012.

[19] Daniel Selva and David Krejci. "A survey and assessment of the capabilities of Cubesats for Earth observation." In: *Acta Astronautica* 74 (2012), pp. 50–68.

[20] Kirk Woellert, Pascale Ehrenfreund, Antonio J. Ricco, and Henry Hertzfeld. "Cubesats: Cost-effective science and technology platforms for emerging and developing nations." In: *Advances in Space Research* 47.4 (2011), pp. 663–684.

[21] Jasper Bouwmeester and J Guo. "Survey of worldwide pico-and nanosatellite missions, distributions and subsystem technology." In: *Acta Astronautica* 67.7 (2010), pp. 854–862.

[22] Armen Poghosyan and Alessandro Golkar. "CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions." In: *Progress in Aerospace Sciences* 88 (2017), pp. 59–83. DOI: 10.1016/j.paerosci.2016.11.002.

[23] Sreeja Nag, Jacqueline LeMoigne, and Olivier de Weck. "Cost and risk analysis of small satellite constellations for earth observation." In: *2014 IEEE Aerospace Conference*. 2014, pp. 1–16.

[24] Rainer Sandau. "Status and trends of small satellite missions for Earth observation." In: *Acta Astronautica* 66.1 (2010), pp. 1–12.

[25] David J. Barnhart, Tanya Vladimirova, and Martin N Sweeting. "Very-small-satellite design for distributed space missions." In: *Journal of Spacecraft and Rockets* 44.6 (2007), pp. 1294–1306.

[26] Carles Araguz, David Llaveria, Estefany Lancheros, Elisenda Bou-Balust, Adriano Camps, Eduard Alarcón, Ignasi Lluch, Hripsime Matevosyan, Alessandro Golkar, Stefania Tonetti, Stefania Cornara, Judith Cote, Stephane Pierotti, Pedro Rodríguez, Angel Alvaro, Mateusz Sochacki, and Janusz Narkiewicz. "Optimized model-based design space exploration of distributed multi-orbit multi-platform Earth observation spacecraft architectures." In: *2018 IEEE Aerospace Conference*. Submitted for publication. 2018.

[27] Randy Rose, Chris Ruf, Debi Rose, Marissa Brummitt, and Aaron Ridley. “The CYGNSS Flight Segment; A Major NASA Science Mission Enabled by Micro-Satellite Technology.” In: *Proceedings of the 2013 IEEE Aerospace Conference*. 2013, pp. 1–13.

[28] Eberhard Gill, Prem Sundaramoorthy, Jasper Bouwmeester, B. Zandbergen, and R. Reinhard. “Formation flying within a constellation of nano-satellites: The QB50 mission.” In: *Acta Astronautica* 82.1 (2013), pp. 110–117.

[29] Edward B Gamble Jr and Reid Simmons. “The impact of autonomy technology on spacecraft software architecture: A case study.” In: *IEEE Intelligent Systems* 5 (1998), pp. 69–75.

[30] Harald Wojtkowiak, Oleskii Balagurin, Gerhard Fellinger, and Hakan Kayal. “ASAP: Autonomy through on-board planning.” In: *Proceedings of the 2013 6th International Conference on Recent Advances in Space Technologies*. IEEE. 2013, pp. 377–381.

[31] Massimo Tipaldi and Luigi Glielmo. “A Survey on Model-based Mission Planning and Execution for Autonomous Spacecraft.” In: *IEEE Systems Journal* (2017). Accepted for publication, pp. 1–13. DOI: 10.1109/JSYST.2017.2720682.

[32] *Space segment operability*. Standard. European Cooperation for Space Standardization, 2008.

[33] Ari Jónsson, Robert A. Morris, and Liam Pedersen. “Autonomy in Space. Current Capabilities and Future Challenges.” In: *AI Magazine* 28 (4 2007). DOI: 10.1609/aimag.v28i4.2066.

[34] Kyriakos G. Vamvoudakis, Panos J. Antsaklis, Warren E. Dixon, João P. Hespanha, Frank L. Lewis, Hamidreza Modares, and Bahare Kiumarsi. “Autonomy and machine intelligence in complex systems: A tutorial.” In: *Proceedings of the 2015 American Control Conference*. IEEE. 2015, pp. 5062–5079.

[35] Claudio Iacopino and Phil Palmer. “Autonomy.” In: *Distributed Space Missions for Earth System Monitoring*. Ed. by Marco D’Errico. Vol. 31. Springer Science & Business Media, 2012. Chap. 10, pp. 309–329.

[36] Sylvain Damiani, Gérard Verfaillie, and Marie-Claire Charmeau. "An earth watching satellite constellation: How to manage a team of watching agents with limited communications." In: *Proceedings of the 4^{th} International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM. 2005, pp. 455–462.

[37] Grégory Bonnet and Catherine Tessier. "Collaboration among a satellite swarm." In: *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM. 2007, p. 54.

[38] Emil Vassev, Roy Sterritt, Christopher Rouff, and Mike Hinchey. "Swarm technology at NASA: building resilient systems." In: *IT Professional Magazine* 14.2 (2012), p. 36.

[39] Roy Sterritt, Christopher A Rouff, Michael G Hinchey, James L Rash, and Walt Truszkowski. "Next generation system and software architectures: Challenges from future NASA exploration missions." In: *Science of Computer Programming* 61.1 (2006), pp. 48–57.

[40] Oliver Montenbruck, Ralph Kahle, Simone D'Amico, and Jean-Sebastien Ardaens. "Navigation and control of the TanDEM-X formation." In: *The Journal of the Astronautical Sciences* 56.3 (2008), pp. 341–357. DOI: 10.1007/BF03256557.

[41] Eberhard Gill, Oliver Montenbruck, and Simone D'Amico. "Autonomous formation flying for the PRISMA mission." In: *Journal of Spacecraft and Rockets* 44.3 (2007), pp. 671–681.

[42] J. Salvador Llorente, Alfredo Agenjo, Carmelo Carrascosa, Cristina de Negueruela, Agnes Mestreau-Garreau, Alexander Cropp, and Andrea Santovincenzo. "PROBA-3: Precise formation flying demonstration mission." In: *Acta Astronautica* 82.1 (2013), pp. 38–46.

[43] Paul T. Grogan, Alessandro Golkar, Seiko Shirasaka, and Oliver de Weck. "Multi-stakeholder interactive simulation for federated satellite systems." In: *2014 IEEE Aerospace Conference*. Big Sky, MT, US, 2014. DOI: 10.1109/AERO.2014.6836253.

[44] Paul Muri and Janise McNair. "A survey of communication sub-systems for intersatellite linked systems and cubesat missions." In: *Journal of Communications* 7.4 (2012), pp. 290–308.

[45] Lloyd Wood, Will Ivancic, Dave Stewart, James Northam, Chris Jackson, and Alex da Silva Curiel. "IPv6 and IPsec on a satellite in space." In: *conference paper B*. Vol. 2. 2007.

[46] Ignasi Lluch, Paul T Grogan, Udrivolf Pica, and Alessandro Golkar. "Simulating a proactive ad-hoc network protocol for Federated Satellite Systems." In: *2015 IEEE Aerospace Conference*. IEEE. Big Sky, MT, US, 2015, pp. 1–16.

[47] Will Ivancic, Wesley M Eddy, Dave Stewart, Lloyd Wood, James Northam, and Chris Jackson. "Experience with Delay-Tolerant Networking from orbit." In: *International Journal of Satellite Communications and Networking* 28.5-6 (2010), pp. 335–351.

[48] Todd S Rose, Siegfried Janson, W LaLumondiere, N Werner, D. H. Hinkley, Darren W. Rowen, R. A. Fields, and R. P. Welle. "LEO to ground optical communications from a small satellite platform." In: vol. 9354. 93540I. 2015, pp. 1–9. DOI: 10.1117/12.2087275.

[49] Katia P. Sycara. "Multiagent systems." In: *AI magazine* 19.2 (1998), p. 79.

[50] Jorge Ocón, E. Rivero, A. Sanchez Montero, A. Cesta, and R. Rasconi. "Multi-agent frameworks for space applications." In: *Proceedings of the 11th International Conference on Space Operations*. Huntsville, AL, 2010.

[51] Simon Garnier, Jacques Gautrais, and Guy Theraulaz. "The biological principles of swarm intelligence." In: *Swarm Intelligence* 1.1 (2007), pp. 3–31.

[52] Luca Tummolini, Marco Mirolli, and Cristiano Castelfranchi. "Stigmergic cues and their uses in coordination: An evolutionary approach." In: *Agents, Simulation and Applications*. Ed. by A Uhrmacher and D Weyns. CRC Press, 2009. Chap. 1, pp. 243–265.

[53] ND Monekosso and P Remagnino. "A collective can do better." In: *Design of Intelligent Multi-Agent Systems*. Springer, 2005. Chap. 7, pp. 215–238.

[54] Yongsheng Ding, Yanling Jin, Lihong Ren, and Kuangrong Hao. “An Intelligent Self-Organization Scheme for the Internet of Things.” In: *IEEE Computational Intelligence Magazine* 8.3 (2013), pp. 41–53.

[55] Frederick Ducatelle, Gianni A. Di Caro, Carlo Pinciroli, and Luca M. Gambardella. “Self-organized cooperation between robotic swarms.” In: *Swarm Intelligence* 5.2 (2011), pp. 73–96.

[56] Lynne E. Parker. “Distributed intelligence: Overview of the field and its application in multi-robot systems.” In: *Journal of Physical Agents* 2.1 (2008), pp. 5–14.

[57] Robert Johansson and Alessandro Saffiotti. “Navigating by stigmergy: A realization on an RFID floor for minimalistic robots.” In: *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*. 2009, pp. 245–252.

[58] Justin Werfel and Radhika Nagpal. “Extended stigmergy in collective construction.” In: *IEEE Intelligent Systems* 21.2 (2006), pp. 20–28.

[59] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. “Designing collective behavior in a termite-inspired robot construction team.” In: *Science* 343.6172 (2014), pp. 754–758.

[60] Diana Tsankova, Velichka Georgieva, Frantisek Zezulka, and Zdenek Bradac. “Immune network control for stigmergy based foraging behaviour of autonomous mobile robots.” In: *International Journal of Adaptive Control and Signal Processing* 21.2-3 (2007), pp. 265–286.

[61] Alejandro Cornejo, Anna Dornhaus, Nancy Lynch, and Radhika Nagpal. “Task allocation in ant colonies.” In: *Distributed computing*. Springer, 2014, pp. 46–60.

[62] Grosan Crina and Abraham Ajith. “Stigmergic Optimization.” In: Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. Chap. Stigmergic Optimization: Inspiration, Technologies and Perspectives, pp. 1–24.

[63] Claudio Iacopino, S Harrison, and Alessandro Brewer. "Mission Planning Systems for Commercial Small-Sat Earth Observation Constellations." In: *Proceedings of the 2015 International Workshop on Planning and Scheduling for Space*. Buenos Aires, Argentina, 2015.

[64] Grégory Beaumet, Gérard Verfaillie, and Marie-Claire Charmeau. "Feasibility of autonomous decision making on board an agile Earth-observing satellite." In: *Computational Intelligence* 27.1 (2011), pp. 123–139. DOI: 10.1111/j.1467-8640.2010.00375.x.

[65] Chong Wang, Jun Li, Ning Jing, Jun Wang, and Hao Chen. "A Distributed Cooperative Dynamic Task Planning Algorithm for Multiple Satellites Based on Multi-agent Hybrid Learning." In: *Chinese Journal of Aeronautics* 24.4 (2011), pp. 493–505. ISSN: 1000-9361. DOI: 10.1016/S1000-9361(11)60057-5.

[66] Christoph Lenzen, Maria Th. Wörle, Falk Mrowka, Michael P. Geyer, and Rüdiger Klaehn. "Automated scheduling for TerraSAR-X/TanDEM-X." In: *Proceedings of the 2011 International Workshop on Planning and Scheduling for Space*. 2011.

[67] Ellis Miller, Phillip Paulsen, and Michael Pasciuto. "Autonomous satellite operations via secure virtual mission operations center." In: *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*. IEEE. 2010, pp. 2607–2610.

[68] Howard Tripp and Phil Palmer. "Stigmergy based behavioural coordination for satellite clusters." In: *Acta Astronautica* 66.7 (2010), pp. 1052–1071.

[69] Juan A Tejo, Amparo Garrigues Rua, and Juan Pablo Arregui. "Planning the operations for Sentinel-1 satellite: how to fit a complex puzzle." In: *SpaceOps 2014 Conference*. 2014, p. 1727.

[70] Jose Antonio González, Oscar Gonzalez, Ángel Monge, Belén Cillero, Fabrizio Pirondini, and Alvaro Ortiz. "DEIMOS-2 Advanced Mission Planning Capabilities." In: *SpaceOps 2014 Conference*. Pasadena, CA, US: American Institute of Aeronautics and Astronautics, 2014.

[71] Claudio Iacopino, Phil Palmer, Nicola Policella, Alessandro Donati, and Andy Brewer. "How Ants Can Manage Your Satellites." In: *Acta Futura* 9 (2014), pp. 59–72. DOI: 10.2420/AF09.2014.59.

[72] Andrew Carrel and Phil Palmer. "An evolutionary algorithm for near-optimal autonomous resource management." In: *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*. 2005.

[73] Chris HolmesParker, A Agigino, and Kagan Tumer. "Evolving distributed resource sharing for cubesat constellations." In: *Proceedings of the Genetic and Evolutionary Computation Conference*. Philadelphia, PA, 2012.

[74] Jonathan Bonnet, Marie-Pierre Gleizes, Elsy Kaddoum, Serge Rainjonneau, and Gregory Flandin. "Multi-satellite Mission Planning Using a Self-Adaptive Multi-agent System." In: *Proceedings of the 2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. 2015, pp. 11–20. DOI: 10.1109/SASO.2015.9.

[75] Grégory Bonnet and Catherine Tessier. "Coordination despite constrained communications: a satellite constellation case." In: *3rd National Conference on Control Architectures of Robots*. 2008, pp. 89–100.

[76] Steve Chien, Daniel Tran, Gregg Rabideau, Steve Schaffer, Daniel Mandl, and Stuart Frye. "Improving the operations of the Earth Observing One mission via automated mission planning." In: *Proceedings of the 11th International Conference on Space Operations*. 2010.

[77] Steve Chien, Joshua Doubleday, Kevin Ortega, Daniel Tran, John Bellardo, Austin Williams, Jordi Piug-Suari, Gary Crum, and Thomas Flatley. "Onboard Autonomy and ground operations automation for the Intelligent Payload Experiment (IPEX) CubeSat mission." In: *Proceedings of the International Symposium on Artificial Intelligence, Robotics, and Automation for Space*. 2012.

[78] Antonio Ceballos, Saddek Bensalem, Amedeo Cesta, L. De Silva, Simone Fratini, Félix Ingrand, J. Ocon, Andrea Orlandini, Frederic Py, Kanna Rajan, R. Rasconi, and M. van Winnendael. "A goal-oriented autonomous controller for space exploration." In: *Proceedings of 11th Symposium on Advanced Space Technologies in Robotics and Automation*. Vol. 11. 2011.

[79] Carles Araguz, Angel Alvaro, Íñigo Del Portillo, Kenny Root, Eduard Alarcón, and Elisenda Bou-Balust. "On autonomous software architectures for distributed spacecraft: A Local-Global Policy." In: *2015 IEEE Aerospace Conference*. IEEE. Big Sky, MT, 2015, pp. 1–9.

[80] Santiago Rodrigo Muñoz. "A scalable distributed autonomy system for fractionated satellite missions." MA thesis. Barcelona: Technical University of Catalonia – UPC BarcelonaTech, 2016.

[81] Stefania Tonetti, Stefania Cornara, and F Pirondini. "Fully Automated Mission Planning and Capacity Analysis Tool for the DEIMOS-2 Agile Satellite." In: *Proceedings of the 13^{th} International Conference on Space Operations*. Pasadena, CA: American Institute of Aeronautics and Astronautics, 2014.

[82] Jacob A Englander, Bruce A Conway, and Trevor Williams. "Automated mission planning via evolutionary algorithms." In: *Journal of Guidance, Control, and Dynamics* 35.6 (2012), pp. 1878–1887.

[83] Seung-woo Baek, Sun-mi Han, Kyeum-rae Cho, Dae-woo Lee, Jang-sik Yang, Peter M Bainum, and Hae-dong Kim. "Development of a scheduling algorithm and GUI for autonomous satellite missions." In: *Acta Astronautica* 68.7 (2011), pp. 1396–1402.

[84] Christoph Steiger, Robert Furnell, and Jose Morales. "On-board control procedures for ESA's Deep space missions Rosetta and Venus Express." In: *Proceedings of the Data Systems in Aerospace Conference*. Vol. 60. 2005.

[85] Romain Grasset-Bourdel, Gérard Verfaillie, and Antoine Flipo. "Planning and replanning for a constellation of agile Earth observation satellites." In: *Proceedings of the ICAPS-11 Workshop on Scheduling and Planning Applications*. Freiburg, Germany, 2011.

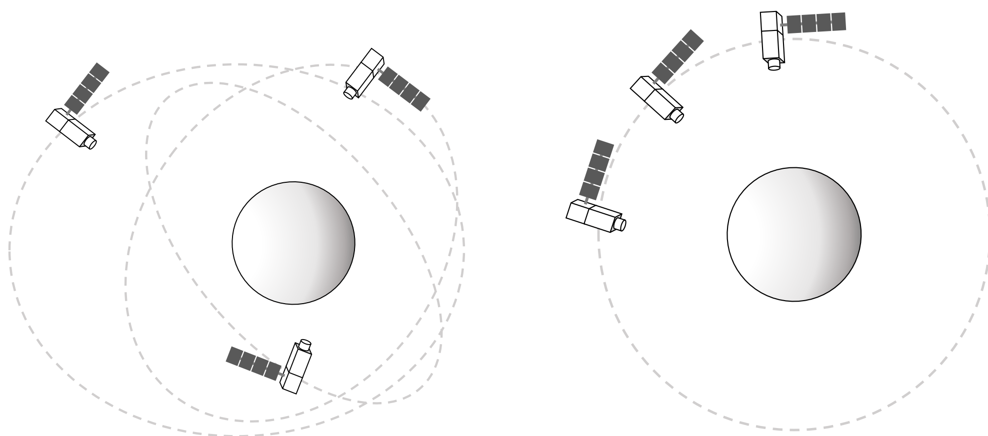
[86] Guohua Wu, Jin Liu, Manhao Ma, and Dishan Qiu. "A two-phase scheduling method with the consideration of task clustering for earth observing satellites." In: *Computers & Operations Research* 40.7 (2013), pp. 1884–1894.

[87] Nicola Bianchessi, Jean-Francois Cordeau, Jacques Desrosiers, Gilbert Laporte, and Vincent Raymond. "A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites." In: *European Journal of Operational Research* 177.2 (2007), pp. 750–762.

[88] Guohua Wu, Witold Pedrycz, Haifeng Li, Manhao Ma, and Jin Liu. "Coordinated Planning of Heterogeneous Earth Observation Resources." In: *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* 46.1 (2016), pp. 109–125.

[89] Mohamed AA Mansour and Maged M Dessouky. "A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite." In: *Computers & Industrial Engineering* 58.3 (2010), pp. 509–520.

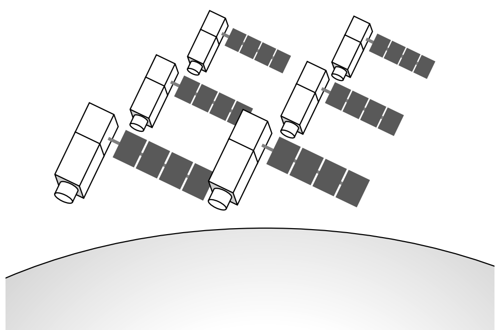
[90] Michael P Geyer, Falk Mrowka, and Christoph Lenzen. "TerraSAR-X/TanDEM-X Mission Planning–Handling Satellites in Close Formation." In: *Proceedings of the AIAA International Conference on Space Operations*. 2010.



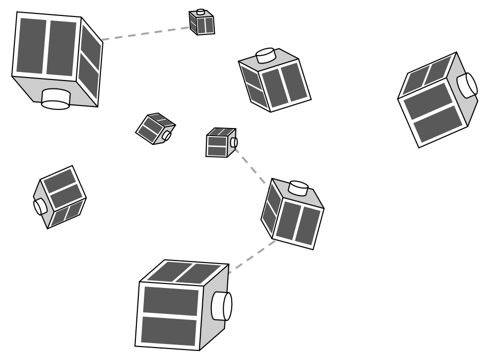
(a) Satellite constellation. Each satellite is in a strategic orbit (usually to guarantee global coverage. Communication between the satellites is scarce or not performed at all (Sat-Com. are an exception, such as Iridium). Each satellite is composed of all the necessary subsystems, including their payload. The size of the spacecraft tends to be big ($\backsim$ 700–800 kg.)

(b) Satellite train. Several spacecraft follow the same trajectory (i.e. they are in the same orbit). The satellites include all the necessary subsystems but their payloads are usually different to allow measurements of multiple magnitudes with negligible delays between them.

Figure 1: Distributed Satellite Systems taxonomy (I): constellation and train.

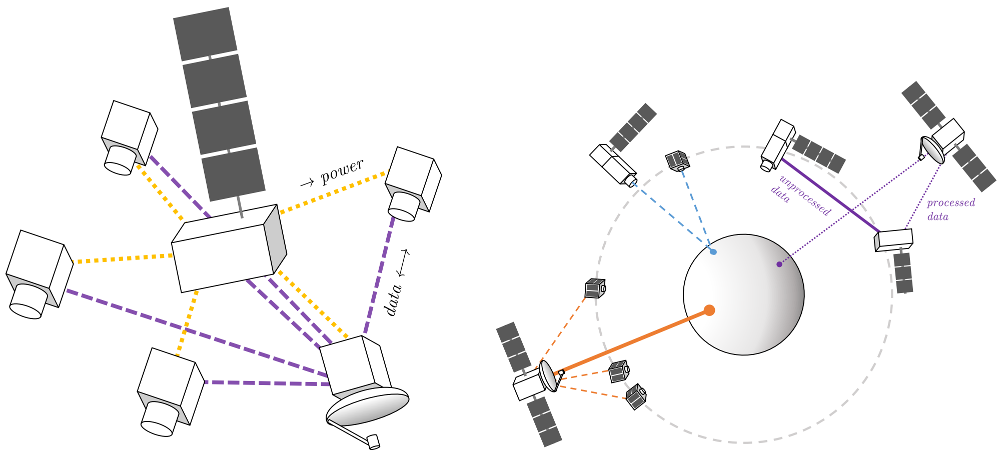


(a) Cluster. The spacecraft are maintaing tight flight formation. Therefore, they need to communicate and exchange, at least, attitude control and orbital state.



(b) Swarm. A cloud of several satellites (e.g. 10, 50, 100...) Each of the satellites is independent: it has its own goals and control mechanisms. Nonetheless, they may eventually comunicate with others in order to avoid collisions or engage in coordinated datakes.

Figure 2: Distributed Satellite Systems taxonomy (II): swarm and cluster.



(a) Fractionated spacecraft. The satellite is divided in several physically-detached modules. Spacecraft functionalities are disagregated on physically-detached modules. Each module is specialized in one or several functions (e.g. energy, instrument, communications...) Data and power are shared wirelessly in close formation.

(b) Federated Satellite System. Composed of a network of satellites formed opportunistically. Satellites may engage in cooperative measurements, foreign data relay and processing activities or similar collaborations. Nonetheless, each satellite includes all their necessary subsystems and their own payload: when they are not part of the FSS, they have their own mission goals.

Figure 3: Distributed Satellite Systems taxonomy (III): fractionated spacecraft and Federated Satellite System (FSS).

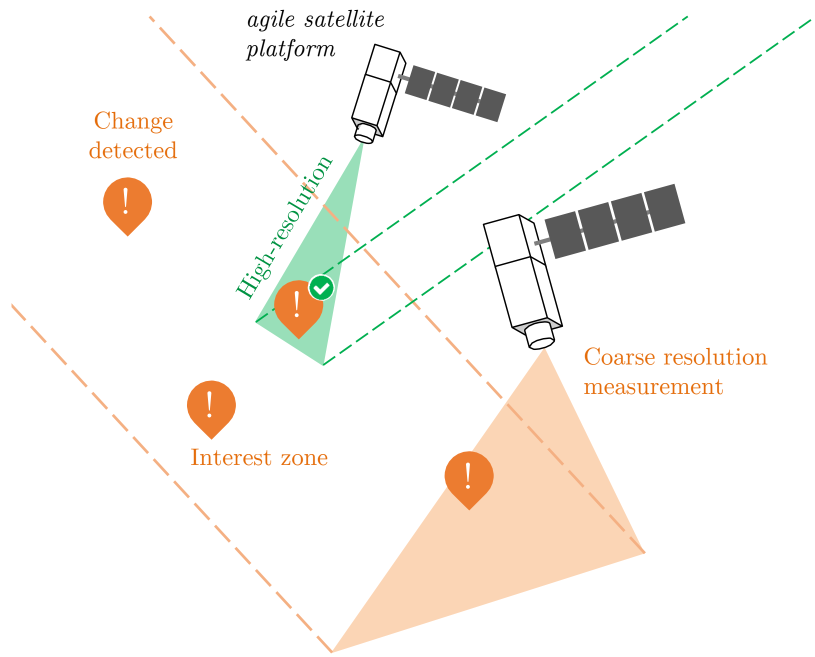


Figure 4: Coordinated multi-point measurements. DSS missions allow to have several coordinated spacecraft capturing data from the same geographical regions. One of such approaches could be the one illustrated in this figure, where one spacecraft with a wide instrument swath, acquires images in low-resolution and detects areas of interest that are later acquired by a secondary spacecraft with narrower swath and better resolution.

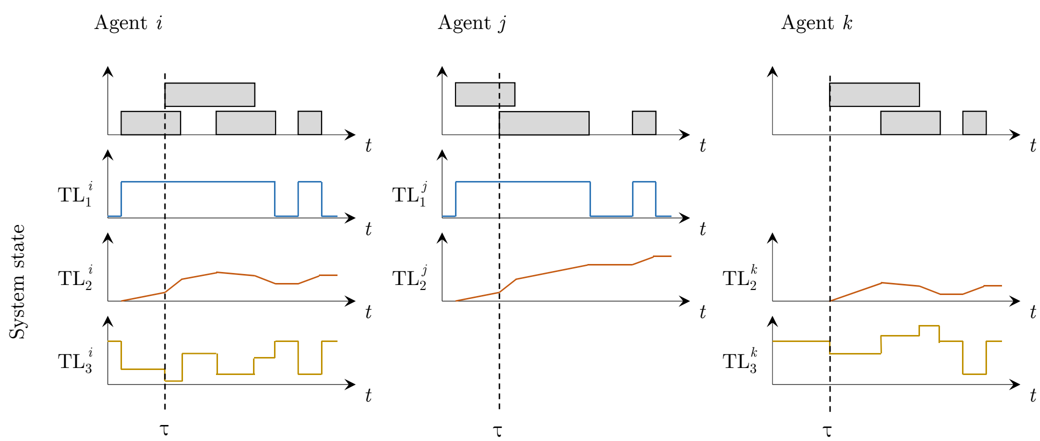


Figure 5: Timeline representations model system states and variables and visualize them together with the timeline of the scheduled tasks (top timelines). While some variables can be represented by integer values as in TL_2, others (e.g. instrument states) can only be represented with a finite set of values (TL_3). Timelines can also represent constraints, usually as maximum or minimum values or delays. Abstract constraints, such as mutual exclusion, can also be modelled as a resource timeline the values of which are binary (e.g. TL_1). MPS that control several spacecraft, would necessitate to represent a set of timelines for each satellite (in the figure represented as Agents i, j and k).

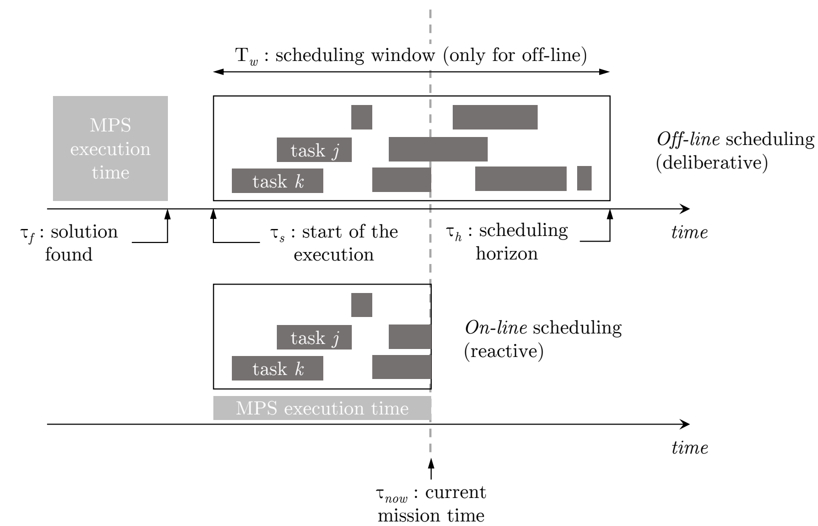


Figure 6: Whereas off-line (i.e. deliberative) systems allocate all tasks within a time window that is in the future, reactive systems constantly determine which tasks have to be enabled. Combinations of both approaches can exist, in which the on-line execution simply corrects or refines a previosly-generated schedule. These usually reduces the computational complexity of the reactive part, while improving the system reactiveness and schedule uncertainly.

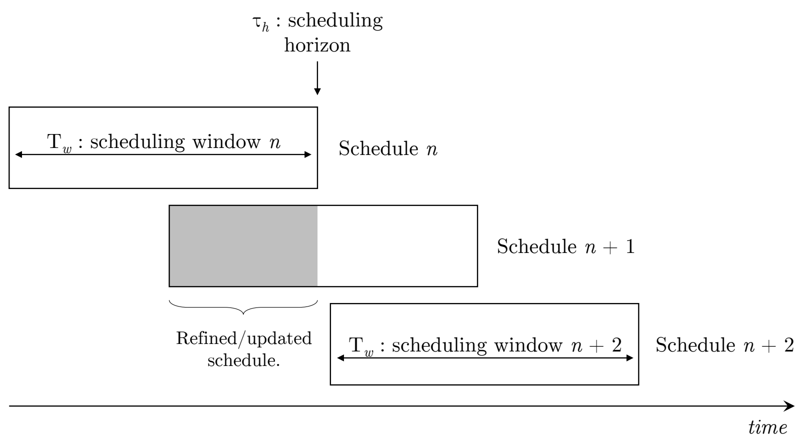


Figure 7: There are cases of off-line MPS in which their scheduling windows are interleaved, yielding to valid parts of a previously-generated schedule to be re-calculated. This approach tries to solve one of the main drawbacks of purely off-line/deliberative systems: because models are not perfect and faults can occur, the later in time a task has been scheduled the more uncertainty is accumulated from all the previous actions. Thus, when the scheduling window is fixed (e.g. a given number of satellite orbits), interleaved schedule windows allows to reduce the undertainty levels for late tasks.

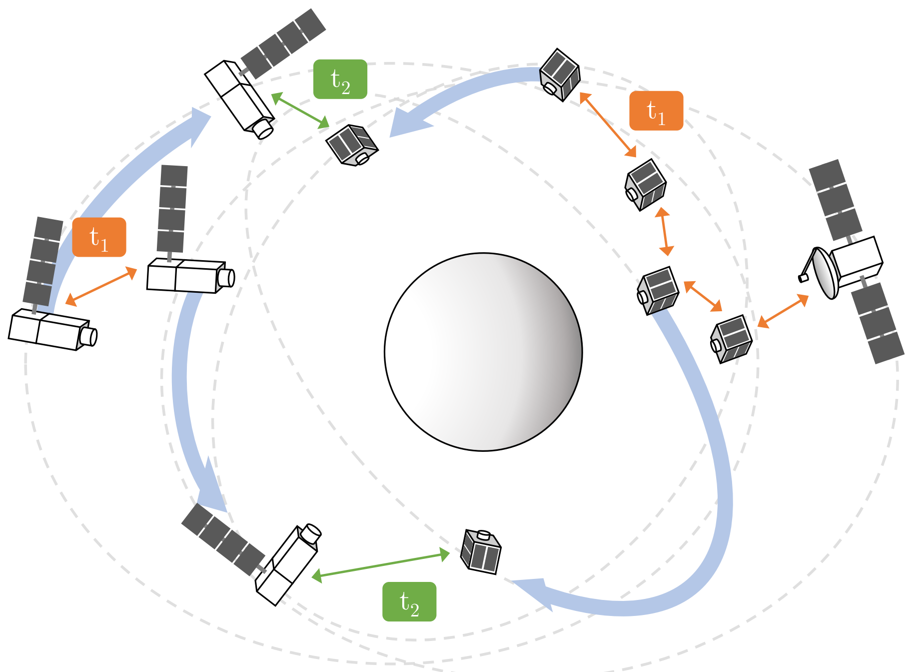
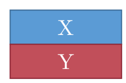
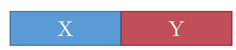
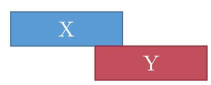
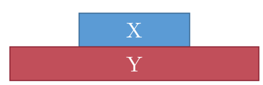
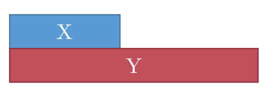


Figure 8: Networks implemented through Inter-Satellite Links present a dynamic structure mostly subjugated to the orbits of the satellites.

Table 1: Allen's Interval algebra relations

Relation	*Representation*
X before Y	
X equal Y	
X meets Y	
X overlaps Y	
X during Y	
X starts Y	
X ends Y