Applying Autonomy to Distributed Satellite Systems

Carles Araguz, Elisenda Bou-Balust, Eduard Alarcón
**Autonomy**, a word the meaning of which is applied to several cutting-edge technologies:

- Smart grids; smart cities;
- Self-driving cars;
- Autonomous teams of robots, UAV's, etc.
- The Internet of Things: devices autonomously interacting;
- ...
What is autonomy?

→ A concept positioned at the intersection of philosophical, scientific and engineering interests:

→ There are several ways to understand “autonomy”:
  - Autonomous ≠ Automatic.
  - 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 it is their ultimate purpose.

Definition: **X is autonomous from Y for p**

Autonomy as an external attribute of the system.

- It’s not absolute: there are degrees of autonomy (a system can be less or more autonomous).
- Autonomy is not an immutable attribute either: context does affect how autonomous a system might be (with respect to something).
Why autonomous spacecraft?


→ Motivated by:
  ▪ Taking advantage of science opportunities. Improving science return (mission “performance”).
  ▪ Reducing spacecraft operational costs.
  ▪ Ensuring robust operations in the presence of uncertainties. Failure tolerance.

→ Implemented through systems with on-board planning capabilities, that strongly relied upon the actual state of the spacecraft.

→ Increasing autonomy through goal-oriented operations.
MPS has become an essential subsystem in spacecraft, specially in Earth Observation missions.

In monolithic satellites:
- Function: to autonomously manage internal resources and spacecraft mission tasks.
- For timeline-based MPS: commonalities identified in (Chien 2012):

**TASKS**
- Observational requests (set by user/operators or generated onboard).
- Maintenance tasks (e.g. reaction wheels de-spinning, recorder device management).
- Tasks are decomposed into detailed system states.

**CONSTRAINTS** (adapted from Chien 2012)
- Atomic resources: binary constraint.
- Depletable resources: integer constraint.
  - Can have intrinsic maximum/minimum values.
  - Commonly modelled resources/constraints:
    - Power, Energy, Memory (payload data storage), Thermal, Attitude...
    - Temporal and spatial constraints, synchronization constraints.
    - Visibility (with ground station or the object to observe).
    - System safety constraints (e.g. direct Sun exposure, SAR instrument mutual exposure).
    - Technological limitations (e.g. simultaneous accesses to GS).
    - Environmental conditions (e.g. cloud coverage, predicted humidity, sunlight).
Mission Planning and Scheduling systems

→ Multiple instances exist from which we can extract several design choices:

1. **Problem modelling:**
   - Typically modelled as Knapsack (Travelling Salesman Problem subset):
     \[
     \begin{align*}
     \text{Max.} \quad & f(X) = \sum_{i=1}^{n} w_i x_i \\
     \text{Subject to:} \quad & \sum_{i=1}^{n} r_i c_i \leq c
     \end{align*}
     \]
   - Problem encapsulation:
     - Several sub-problems and levels (constellation-, spacecraft-, subsystem-level).
   - Solution refinement:
     - NASA/JPL’s Earth Observing-1 ([Chien, 2010](#)) → Two parts: pre-computation *on-ground* (week planner) plus accurate refinement *on-board* (with shorter scheduling window).
   - Algorithms and metaheuristics:
     - Dynamic programming, negotiation-based approaches.
     - Ant Colony Optimization, Local Search, greedy algorithms, Tabu Search and **Genetic or evolutionary Algorithms** (most common).
Mission Planning and Scheduling systems

2 → **Optimization:**
- **Common:** multi-objective optimization (optimize the use of multiple resources simultaneously).
- **Optimization is not always targeted** → **distribution of user requests** (e.g. Bonnet 2008 or TerraSAR-X/TanDEM-X).

3 → **Execution and runtime characteristics:**
- **On-ground:** the execution is performed by the spacecraft.
- **On-board:** the execution is performed at the ground segment and delivered to the spacecraft (delays, uncertainty, unfeasible).
- **Reactive** (on-line): spacecraft actions are determined at every moment, depending on the current state of the inputs.
- **Deliberative** (off-line): plan of action generated for a given future scheduling window.
- **Hybrid** approaches: on-ground/on-board (e.g. Damiani 2005), reactive/deliberative (e.g. Beaumet 2011).
Autonomous Distributed Spacecraft

→ In monolithic spacecraft missions (Iacopino and Palmer, 2012):
  ▪ Flexibility and adaptability – mission goals change.
  ▪ System agility (or “responsiveness”) – environmental changes, internal failures.
  ▪ Complexity handling.

→ DSS: new mission architectures and paradigms.

→ Why autonomous Distributed Satellite Systems:
  ▪ Dynamic architectures – incremental deployment, graceful degradation.
  ▪ Networked modules with different architectural characteristics: FSS, swarms, constellations...
  ▪ Heterogeneous computational, communication and functional capabilities:
    • A variety of payload technologies and characteristics (i.e. spatial or spectral resolution).
    • Nano-satellite platforms (limited computational resources & constrained communication capabilities).
    • Hybridization of high-density nano-satellite components with high-performance spacecraft.
From monolithic to distributed autonomous systems

→ Traditional MPS (i.e. for monolithic satellites) can’t be applied or are insufficient:
  ▪ For on-board approaches:
    • Centralized metaheuristics are not distributable (efficiently) → Parallel metaheuristics need to be applied.
    • Bio-inspired approaches rely on the “environment” as a channel for indirect communication (i.e. stigmergy).
  ▪ In general:
    • Time is not always a common dimension → timeline based representations and solvers can’t be applied.

→ Current works – MPS for distributed satellite systems:
  ▪ In currently operative missions:
    • GMV’s S1MP for Sentinel-S1: on-ground, multiple-ground station optimization, complex constraints (Tejo, 2014).
    • TerraSAR-X/TanDEM-X: on-board, no optimization, complex instrument constraints (Lenzen, 2011).
  ▪ Negotiation-based approaches:
    • Based in Multi-Agent paradigm.
    • Better scalability and adaptability.
    • E.g. (HolmesParker, 2012) and (Damiani, 2005).
  ▪ Bio-inspired, self-organizing MPS: academic works.
    • Influenced by swarm intelligence and collective organization schemes.
    • Stigmergy (Tripp, 2010).
    • Ant Colony Optimization (Iacopino, 2014): collaborative tasks.
Applying autonomy in DSS (for Earth Observation)

→ Design of *top-level* MPS.

→ The importance of interactions: **network-aware** algorithms.
  
  • Networks strongly affected by **orbital characteristics**.
  • Optimization methods/algorithms that leverage on this characteristic.
    
    ▪ Aware of **computational** restrictions (i.e. nano-satellites).

→ Missing features that should be incorporated (potentially):
  
  ▪ Creation of **opportunistic networks**.
  ▪ In-orbit data **services**: relay, processing (e.g. compression), storage…
  ▪ **Exchange of resources**: e.g. energy (WPT), share spaceborne commodities (FSS approach).
  ▪ **Collective behaviours**: coordination, cooperation, deliberation, collaboration.
    
    • Multi-point measurements.
    • Enhanced distributed detection of science opportunities.
Final remarks

→ Autonomous DSS: an on-going research at UPC’s NanoSat Lab.

→ **Fundamental questions** critical for the design of new MPS for DSS:

  - **Architectural aspects:**
    - What should be the **size of a DSS** so as to fulfil the application’s requirements while providing **feasible** operational mechanisms (i.e. task scheduling, collaborative datatakes) and maintaining a good performance?
    - **Where** should decision-making be located: *distributed* among the spacecraft nodes or *centralized* on the ground segment?
    - Placing the autonomy barrier: what is the **level of autonomy** that DSS can *tolerate, implement* or *require*?

  - **Network aspects:**
    - What are the effects of small spacecraft’s communication constraints for mission planning and scheduling algorithms?
    - What are the effects of dynamic and opportunistic networks for mission planning and scheduling algorithms?
    - What are the necessary mechanisms to achieve self-healing and resilient DSS at the MPS level?
    - What are the effects of communication-impaired small spacecraft, for MPS?

  - **Computational aspects:**
    - What are the **computational resources** (or reasoning capabilities) required at the nodes?
    - Are **Multi-agent** approaches suitable for on-board autonomy systems? Are **bio-inspired** approaches suitable for DSS?
    - What are the effects of small satellites’ constrained computational resources, for MPS?
Thanks for your attention

Carles Araguz
carles.araguz@upc.edu

Elisenda Bou-Balust
elisenda.bou@upc.edu

Eduard Alarcón
eduard.alarcon@upc.edu