

Space Autonomy as Migration of Functionality: The Mars case

Tim Grant
Science & Technology b.v.
tgrant@cs.up.ac.za

André Bos
Science & Technology b.v.
bos@science-and-technology.nl

Mark Neerinx
TNO Human Factors
neerinx@tm.tno.nl

Antonio Olmedo Soler
OK-Systems S.A.
aos@olmedo.com

Uwe Brauer
EADS Space Transportation
uwe.brauer@space.eads.net

Mikael Wolff
ESA ESTEC
mikael.wolff@esa.int

Abstract

This paper develops Grandjean and Lecouat's insight that spacecraft autonomy can be seen as the migration of functionality from the ground segment to the space segment. Their insight is extended to manned planetary exploration missions and applied to an IT-based crew assistant for supporting manned and unmanned Martian operations.

The Mission Crew Execution Assistant (MECA) system is seen as a distributed, personal ePartner in a ubiquitous computing environment, designed to amplify the cognitive capacities of human-machine teams during planetary exploration missions to cope autonomously with unexpected, complex and potentially hazardous situations. The paper shows that the autonomy requirements have implications for the MECA system's architecture and functionality.

The research reported here has been performed by a consortium led by TNO Human Factors (NL) in Phase 1 of the MECA project funded by the European Space Agency (ESA) under contract number 19149/05/NL/JA.

1. Introduction

1.1. Background

Over the past decade there has been extensive debate about the needs for and expected benefits, complications and implementation of spacecraft autonomy. The flights of NASA's Deep Space One and ESA's PROBA spacecraft have been milestones in

the development of autonomy. Drivers for autonomy are seen as interruptions and delays in the Earth-space communications link, spacecraft survival in the presence of on-board faults, and cost reduction in the ground segment. The consensus is that autonomy involves building additional functionality into the spacecraft.

Spacecraft autonomy brings with it complications for ground-based control. Controllers have all the usual problems of monitoring a distant system over a communications link with limited bandwidth, of assessing the on-board situation, of determining the appropriate course of action, and of generating and uplinking telecommand-sequences. In addition, they have to contend with tracking and understanding the decisions taken by the spacecraft's autonomous systems and intervening where necessary (and possible).

For longer duration manned missions, such as the human exploration of Mars [3] [4] [5], the concept of crew autonomy versus ground control is likely to undergo drastic changes. The crew has to be closely involved in the decision-making process and in resource management. For cost reasons, staffing for mission control will be reduced by comparison with present-day Shuttle and ISS missions. Crew motivation and satisfaction needs for long duration missions are additional rationales for their increased involvement in decision-making.

For day-to-day operations on Mars, the distance between the planetary explorers and Earth-bound mission controllers prohibits the controllers' involvement in real-time decisions. Mission control

will take part in strategic and tactical planning and mission evaluation activities, leaving the crew responsible for execution-level planning and mission execution. To this end, the crew will need IT-based support.

1.2. MECA project

The main objective of the MECA project is to develop the user and software requirements and an architectural baseline for a IT-based system that will enable autonomous operation by human-machine teams during planetary exploration missions. The MECA system is seen as a distributed, personal ePartner in a ubiquitous computing environment (see Figure 1), designed to amplify the cognitive capabilities of human-machine teams to cope autonomously with unexpected, complex and potentially hazardous situations.

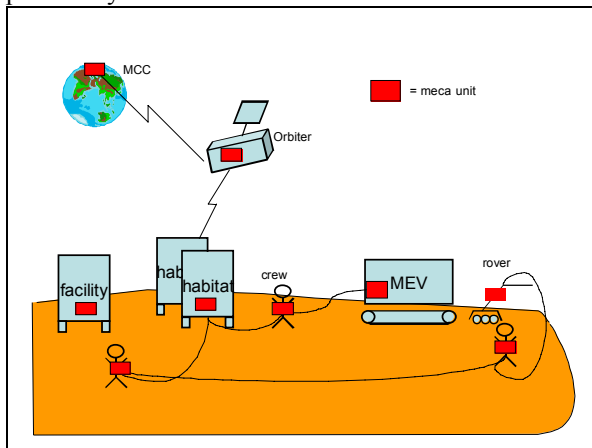


Figure 1. MECA as ubiquitous ePartner.

MECA is required [6] to:

- Allow the crew to schedule and make decisions about short- and mid-term activities.
- Be pro-active w.r.t. health and status monitoring of equipment and facilities.
- Adjust monitoring and supervision algorithms as equipment and facilities evolve.
- Support collaboration for support teams and explorers separated in time and space [7].
- Provide “what-if” scenario evaluation.
- Provide crew with updated procedural knowledge [8].
- Provide crew with advice (potentially unsolicited) about possible alternatives and their effects on all available resources.
- Execute tasks delegated to it by the crew.

The MECA system is especially important when the communications link with Earth is absent.

The MECA project is divided into two phases. An initial version of the user and software requirements have been defined in Phase 1, based on use-cases derived from four candidate mission scenarios and on a survey of research topics and emerging technologies. The requirements will be refined in Phase 2 by defining an architectural baseline and by building and evaluating a proof-of-concept demonstrator of the MECA system using present-day technologies. The end-deliverables of the MECA project will be the refined Requirements Baseline (RB) and Interface Requirements Document (IRD), compliant with the ECSS-E-40 standard.

The MECA project began in August 2005. Phase 1 was completed with a System Requirements Review in February 2006. Phase 2 is expected to last 12 months from May 2006.

The Phase 1 survey covered operational¹, human factors², and technical³ topics. The results of surveying each topic were documented in a white paper, with the conclusions and recommendations being collected together in a Technical Note. The recommendations were incorporated as requirements in the initial RB. This paper summarises the Phase 1 results from the MECA white paper on autonomy [9].

1.3. Purpose and scope of paper

The purpose of this paper is to present a development of Grandjean and Lecouat’s [1] insight that spacecraft autonomy can be seen as the migration of functionality from the ground segment to the space segment.

In this paper, we do not intend to propose a specific autonomous mission design for Martian exploration. Rather we use Martian exploration as an illustrative case for regarding autonomy as functionality migration.

1.4. Structure of paper

This paper is divided into six sections. Section 1 gives the background, overviews the MECA project, and defines the purpose, scope, and structure of the paper. Section 2 defines autonomy and describes the

¹ Autonomy, risk management and high reliability organisations, operational process models, and human supervisory control.

² Cognitive task load, collaboration and co-operation, crew resource management, naturalistic and rational decision-making, situation awareness and sensemaking, and trust and emotions.

³ Distributed intelligence and agent architectures, automated planning and scheduling, model-based reasoning, system health management, modelling simulation and gaming, visualisation human-computer interfaces and augmented reality, and comparable systems.

drivers for applying autonomy in space missions. Section 3 summarises a number of approaches to autonomy, describes the migration of functionality in more detail, and extends it to planetary exploration. Section 4 applies the extended approach to the Martian reference mission. Section 5 draws out the implications for the MECA system. Section 6 lists references.

2. Autonomy and its drivers

2.1. Defining autonomy

In the space literature, autonomy is rarely defined rigorously. Informally, spacecraft autonomy means making spacecraft less dependent on ground-based control.

Easter & Staehle [10] define spacecraft autonomy as “the independence of the man/machine flight system from direct, real-time control by the ground over a specified period of time” (p.2-1). The problem with this definition is the specified period of time. In MECA it may not be possible to be specific about the period of time over which the resources on the Martian surface have to be independent. The period will be at least as long as the Mars-Earth round-trip time at light speed.

Castelfranchi [11] distinguishes two types of autonomy: autonomy from an agent’s physical context; and autonomy from the agent’s social context. The first is related to homeostasis, i.e. the ability of an agent to compensate for changes in its environment. The second means that an agent will not unconditionally follow the goals that other agents attempt to impose on it. In the MECA context, the (command) relationship between Earth-based control and the astronauts on the Martian surface is a social one. Therefore, it is the second meaning of autonomy that is relevant to MECA.

Wan, Braspenning & Vreeswijk [2] point out that the concept of spacecraft autonomy was intended to enable spacecraft to continue with their mission during a *temporary* loss of contact with ground control. They warn that ground controllers will have to accept that truly autonomous spacecraft will have to have the independence to refuse to execute ground-generated telecommands. Finally, they aver that truly autonomous spacecraft will necessarily have to evolve.

2.2. Autonomy as additional functionality

The consensus is that autonomy involves building additional functionality into the spacecraft. The literature talks about “autonomous spacecraft” and “on-board autonomy.” Typical autonomous functions include guidance, navigation and (attitude and orbit

control (GNC), fault detection, isolation and recovery (FDIR), and planning and scheduling (P&S).

2.3. Autonomy drivers

Traditionally, the drivers for autonomy are seen as interruptions and delays in the Earth-space communications link, spacecraft survival in the presence of on-board faults, and cost reduction in the ground segment. Communications may be interrupted because there is only one ground station serving the spacecraft, because another body intervenes in the link, or because the link is unreliable.

In manned planetary missions an additional driver is that the crew has to be closely involved in decision making and in resource management. During missions with durations measured in years, giving the crew autonomy in execution-level planning, mission execution, and decision-making will add to their motivation and satisfaction.

2.4. Approaches to autonomy

In Phase 1 of the MECA project several approaches to autonomy were identified [9]. One approach is to view autonomy as an extension of the spectrum of operating modes in the human supervisory control literature [13]. Another is to view autonomy as movement up the command spectrum [14] towards control-free command. In this paper, autonomy is viewed as functionality migration.

3. Migration of functionality

3.1. Grandjean & Lecouat’s insight

Grandjean and Lecouat make a perceptive observation in the last sentence of the introduction to their paper [1]. They state that: “as ground segment automation is a delegation of some decisions from operators to the ground system, on-board autonomy is a delegation of decisions from the ground segment to the space segment and similar issues have to be solved.” They note that decisions traditionally taken by operators include mission planning, task scheduling, schedule execution, navigation and orbit control, and fault detection and recovery.

3.2. Generalising the insight

Grandjean and Lecouat [1] regard a space mission as being split into two levels: a space segment and a ground segment. Borrowing from the human supervisory control literature [13], the space segment is

the Process Under Control (PUC) that interacts with the task environment. The ground segment is the Controlling Process (CP) in which plans are developed for execution by the PUC. In organisational terms, a space mission is a two-level hierarchy.

This simple picture is inadequate for manned missions, because it does not recognise astronauts as anything more than execution agents. In reality, astronauts are entirely capable of supervisory functions [13], such as planning, programming software-intensive systems, delegating control, monitoring, intervening, and learning from experience. Considering manned activities on the ISS, for example, we see astronauts performing supervisory tasks, such as monitoring the automated execution of experiments in payloads and reprogramming the payload after a failure.

This example of manned activities on the ISS shows that the space segment is organisationally more complex than the simple model would allow. Within the space segment, there is both a process under control (i.e. the payload) and a controlling process (i.e. the supervising astronaut). Hence, manned operations on the ISS involves a three-level hierarchy. At the top level, the ground segment is a CP, with the space segment as its PUC. The space segment is itself a control system dividing into a CP at the second level and a PUC at the third level.

In planetary exploration missions involving human-machine teams there will be more than three levels in the organisational hierarchy. Starting at the bottom of the hierarchy, present-day Martian rovers, like Spirit and Opportunity, already embody two levels: an execution-level PUC that (e.g.) moves the rover forward over the terrain, and a software-based CP that monitors and controls the movement to avoid large rocks. If such rovers were operating under the control of an astronaut, the astronaut would represent a third level. The astronaut-rover team would probably itself be monitored by the captain, e.g. from the Home Base. The captain would then represent a fourth level, with the Earth-based mission control centre being a fifth level. At the top of the hierarchy is the mission director.

We can usefully generalise Grandjean and Lecouat's observation to say that each step by which a decision-making function is migrated down a level in the organisational hierarchy yields an increase in autonomy at the lower level.

3.3. Mission control functions

A present-day mission control system (MCS) monitors the health of a spacecraft by checking the telemetry data sent from the spacecraft and controls it

by sending it telecommands [15]. The MCS offers user interfaces to operations staff in the form of various telemetry and telecommand screens. Depending on the mission, the MCS can also have certain dedicated components, such as a mission planning system (MPS) and a data disposition system to allow distribution of payload data to end users.

Within the mission control centre there may well also be a software simulator, modelling the spacecraft and the ground station. Ground systems and operations engineers use the simulator for testing the ground system, including the MCS, and for training operations staff.

A flight dynamics team will track the spacecraft's orbit and the health of the GNC system, providing inputs to the operations staff as necessary.

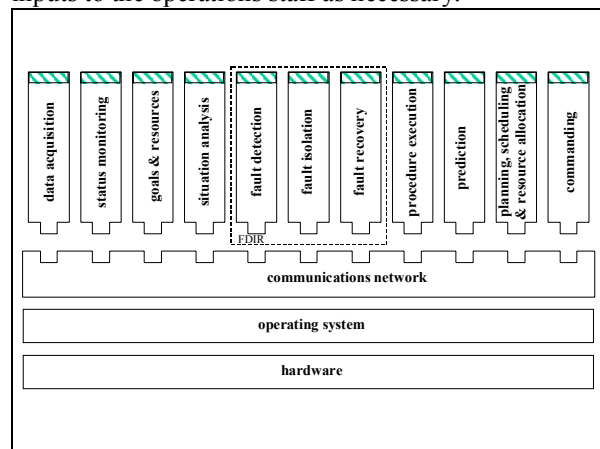


Figure 2. COSPOL functions in the control loop.

We shall use the Crew/Operator Support POLicy (COSPOL) architecture [12] as the set of all mission control functions⁴. The COSPOL functions in the control loop⁵ are (see Figure 2):

- *Data acquisition (DA)*. DA acquires the telemetry data from the PUC. It is responsible for interfacing to the communications network. It has no domain-specific knowledge or intelligence, and simply performs an information-handling function. It extracts the data from incoming data packets or streams and calibrates the data for use by the other applications.
- *Status monitoring (SM)*. SM maintains a database of the current status of the objects of interest in the task environment, in the process under control, and in the control system itself. The current status

⁴ It is unlikely that any given MCS will implement all of the COSPOL functions.

⁵ Support functions, such as the Event Logger (EL), operator training facilities, and editors are not described further in this paper.

is displayed to the operator, typically as a mimic, schematic, or geographical diagram. SM performs a resource management function, where the objects of interest are the resources. It does not maintain a history of events⁶.

- *Goals & resources (GR)*. GR is only applicable to nested control systems, i.e. where this control system receives its goals and resource allocations from another control system at a higher level in the organizational hierarchy. GR checks the feasibility of a new instruction, which may contain new goals, new constraints, a new resource allocation, or a combination of these. The new goals, constraints, and resource allocations are checked for compatibility with the existing set of goals, constraints, and resources.
- *Situation analysis (SA)*. SA maintains a database of the currently-recognised situation, identifying situations at a higher cognitive level from the information it receives from SM. Identification involves combining information about the change in status of an object of interest, with information about the current status of other objects of interest and, if necessary, with historical information⁷. The combined information is matched against templates representing information about prototypical situations.
- *Fault Detection (FD)*. FD detects abnormal situations. Abnormal situations can arise from events in the task environment (e.g. cosmic rays), in the process under control (e.g. sensor failures), or in the control system itself (e.g. communications failures, lack of domain knowledge). Detection can be done on the basis of violations of threshold values, by trending, and by detecting discrepancies between expectations (i.e. planned or predicted events) and actual events.
- *Fault Isolation (FI)*. FI brings the PUC to a safe state as quickly as possible, minimising propagation of the adverse effects of the abnormal situation. FI is often implemented within the PUC itself, as “failure detection & correction,” redundancy or diversity.
- *Fault Recovery (FR)*. FR diagnoses the abnormal situation from the symptoms to find the root cause of failure (RCOF). Having identified the cause, FR either generates one or more instructions, or it may prescribe the execution of a procedure designed to bring the PUC back to an operational state (possibly degraded).

- *Procedure Execution (PE)*. PE selects, instantiates, and runs standard operating procedures (SOPs), as well as displaying them to the operator. The SOPs represent courses of action to be taken in normal or abnormal situations. When an SOP is running, PE selects the next action, checks that its preconditions are satisfied, obtains the operator's approval to execute the action (if that is required), and generates the corresponding instruction, sending it to Commanding (see below) for transmission to the PUC. When a change in the status of an object of interest (from SM) shows that the action has succeeded, then PE selects the next action in the SOP.
- *Planning & Scheduling (PS)*. PS displays pre-planned activities to the operator, and enables the operator to insert, modify and remove activities in a plan and to allocate resources and start times to activities. PS may have additional functionality to generate plans (i.e. courses of action) and to schedule them automatically.
- *Prediction (PR)*. PR projects the future changes of state/situation in the process under control and its task environment, taking into account any planned actions (obtained from PS). PR may be implemented using statistical or simulation technology. The predictions can be displayed to the operator, e.g. by animating the predicted course of events so that the operator can evaluate or rehearse an underlying plan.
- *Commanding (CO)*. CO enables instructions (c.f. telecommands) to be sent directly to objects within the PUC. Instructions can be generated by the operator or by other applications. Operator commanding may be implemented by providing direct-manipulation facilities in the mimic, schematic or geographical diagram used to display status information (see SM). In a mirror image of DA, CO encapsulates the instructions in the outgoing data stream or packets, dispatching them to the process under control via the communications network.

Only a subset of the COSPOL functions can be migrated. Each level in the hierarchy must have at least the functionality of acquiring data from and issuing instructions to the level below. Moreover, the status of the lower level must be monitored. Thus, DA, SM, and CO is the basic set of functionality at each level. GR is only present at a particular level if there is an organisational level above it. There is no COSPOL functionality at the execution level, nor at the top level.

The COSPOL functions that can be migrated are SA, FD, FI, FR, PR, PS, and PE. FI and FR can only be migrated if FD is also at the lower level.

⁶ This is done by EL.

⁷ Obtained from EL.

4. Application to Martian exploration

4.1. Migration cases

In our analysis of various lunar and Martian mission scenarios, we observed that manned planetary exploration is invariably preceded by unmanned planetary operations. For example, in the Mars reference mission, facilities and rovers will be pre-positioned on the Martian surface. Before the human crew can be launched, these facilities and rovers will have to autonomously deploy and perform check-out. Prior to the crew's arrival, the rovers will perform substantial robotic operations, both tele-operated and autonomous.

During unmanned operations, subsets of the MECA functionality are needed on the planetary surface, e.g. to schedule and make decisions about short-term activities, for health and status monitoring, and to execute delegated tasks. Therefore, we considered two cases for functionality migration:

- *Unmanned operation of facilities and rovers.* This involves a four-level hierarchy: execution system of facility/rover, control system of facility/rover, MCS, mission director.
- *Operation by human-machine teams.* This involves a six-level hierarchy: execution system of facility/rover, control system of facility/rover, astronaut, captain, MCS, mission director.

The starting point in each case is with all COSPOL functions in the MCS.

4.2. Migration in unmanned operations

Only one step of migration is possible in unmanned operations: from the MCS to the control systems of the facilities and rovers. Various permutations of the COSPOL functions SA, FD, FI, FR, PR, PS, and PR are possible.

4.3. Migration in manned operations

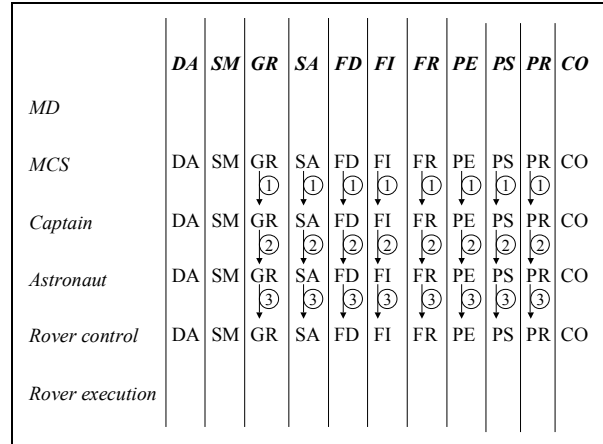


Figure 3. Three steps of functionality migration.

Three steps of migration are possible in the operation of human-machine teams (see Figure 3): from the MCS to the captain, from the captain to individual astronauts, and from astronauts to the control systems of the facilities and rovers. Within each migration step, various permutations of the COSPOL functions SA, FD, FI, FR, PR, PS, and PR are possible.

5. Implications for MECA

The implications for MECA are as follows:

- MECA should support *autonomy management*, i.e. tracking which functions have been migrated and to what organisational level.
- MECA should have a *modular architecture* to enable migration by individual functions. An obvious modularity principle to adopt is the multi-agent systems approach.
- MECA units should be designed as *adaptive, functionally accurate agents*. They need to be adaptive to enable monitoring and supervisory algorithms to adjust as equipment and facilities evolve over the duration of the mission. They need to be functionally accurate to be tolerant of errors in other agents.

6. References

[1] P. Grandjean and F. Lecouat, "Scheduling and Plan Execution: from Ground Segment Automation to Autonomous Spacecraft Operation Concepts", Proceedings of the On-Autonomy Workshop, ESTEC, Noordwijk, Netherlands, 17-19 October 2001.

[2] A.D.M. Wan, P.J. Braspenning and G.A.W. Vreeswijk, "Limits to Ground Control in Autonomous Spacecraft",

Proceedings of the 10th Goddard Conference on Space Applications of Artificial Intelligence, NASA Goddard Space Flight Center, Greenbelt, Maryland, USA, May 1995.

[3] Hoffman, S.J., and D.I. Kaplan (eds), *Human Exploration of Mars: The reference mission of the NASA Mars Exploration Study Team*, NASA Johnson Space Center, NASA SP-6107, July 1997.

[4] Hoffman, S.J. (ed), *The Mars Surface Mission: A description of human and robotic surface activities*, NASA Johnson Space Center, NASA TP-2001-209371, Dec 2001.

[5] Kminek, G., *Human Mars Mission Project: Human surface operations on Mars*, ESA/Aurora/GK/EE/004.04, issue 1, revision 1, June 2004.

[6] ESA Directorate of Technical and Quality Management, *Statement of Work: Mission Execution Crew Assistant*, ESA SOW TRP ESD-023, Issue 1.0, 24 May 2004.

[7] M. Neerinx, J. Lindenberg, N. Smets, A. Bos, U. Brauer, T.J. Grant, A. Olmedo Soler and M. Wolff, "Cognitive engineering for long duration missions: Human-machine cooperation on Mars", Proceedings of the 2nd Space Mission Challenges for Information Technology (SMC-IT 2006), Pasadena, California, USA, 2006.

[8] A. Bos, L. Breebart, T.J. Grant, M. Neerinx, A. Olmedo Soler, U. Brauer, and M. Wolff, "Supporting Complex Astronaut Tasks: The right advice at the right time", Proceedings of the 2nd Space Mission Challenges for Information Technology (SMC-IT 2006), Pasadena, California, USA, 2006.

[9] Grant, T.J. *Autonomy*, white paper, Mission Execution Crew Assistant (MECA) project, Science & Technology b.v., Delft, Netherlands, 11 February 2006.

[10] Easter, R.W. and R.L. Staehle, *Space Platforms and Autonomy*, Technical Report JPL D-1973, Jet Propulsion Laboratory, Technology and Space Program Development, NASA/California Institute of Technology, Pasadena, California, USA, 1984.

[11] C. Castelfranchi, "Guarantees for autonomy in cognitive agent architecture", in Wooldridge, M.J. and N.R. Jennings (eds), Proceedings of the 1994 Workshop on Agent Theories, Architectures, and Languages. John Wiley & Sons, Chichester, UK, 1994.

[12] T.J. Grant, "A Domain-Independent Architecture for Decision Support Applications", International Journal of Advanced Manufacturing Systems, 5, 2, 2002, pp.20-46.

[13] T.B. Sheridan, "Supervisory Control", chapter 9.6 in Salvendy, G. (ed), *Handbook of Human Factors*, John Wiley & Sons, 1987, pp.1243-1268.

[14] Alberts, D.S., and R.E. Hayes, *Power to the Edge: Command Control in the information age*, US Department of

Defense Command & Control Research Program, Washington DC, USA, 2003.

[15] ESA Spacecraft operations, "About Mission Data Systems", http://www.esrin.esa.it/spacecraftops/ESOC-Article-art_print_friendly_1069167510692.html, accessed 13 January 2006.