The Realization of Model-Based Spacecraft Mission Planning and Execution of High-Level Autonomy

Xinying Yan

Abstract At present, the demand for improving spacecraft autonomy is becoming increasingly urgent. Traditional operations relying on ground segment or on-board time-based commands are no longer able to meet the current needs of spacecrafts. According to the four levels of autonomy defined in the standard ECSS-E-ST-70-11C, the paper proposes a practical realization of the four levels autonomy within the mission planning and execution application filed. The paper illustrates a model-based hierarchical layered software architecture of mission planning and execution, and shows the methods used to realize the autonomy of each level. Based on the software architecture, the paper explains how to realize high-level autonomy in a geostationary (GEO) meteorological satellite. The realization is organized with definite border between two related levels of the software architecture and it is practical in engineering because of the decoupling of the realization details within each level. The principles extracted from this method can be regarded as irrelevant to specific application and can be widely used in spacecrafts which need high reliability and autonomy.

Keywords High-level autonomy · Mission planning and execution · Hierarchical layered software architecture

1 Introduction

At present, the demand for improving autonomy of spacecraft is becoming increasingly urgent [1]. Traditional operations relying on ground segment are no longer able to meet the current needs of spacecraft as follows: (1) The mission requirements of spacecraft, e.g., multiple mission stages, complex mission purposes, and complicated operation steps of spacecrafts for deep space exploration and space attack-defense, make it impossible to rely on ground operations solely; (2) Longtime continuous

X. Yan (✉)
Beijing Institute of Control Engineering, Beijing, China
e-mail: xinying_7@163.com

Y. Jia et al. (eds.), *Proceedings of 2023 Chinese Intelligent Systems Conference*,
Lecture Notes in Electrical Engineering 1091,
https://doi.org/10.1007/978-981-99-6886-2_30

Table 1 Spacecraft autonomy level definition [4]

Level	Descriptions	Functions
E1	Mission execution under ground control; limited on-board capability for safety issues	Real-time control from ground for nominal operations; execution of time-tagged commands for safety issues
E2	Execution of preplanned, ground-defined, mission operations on board	Capability to store time-based commands within an on-board scheduler
E3	Execution of adaptive mission operations on board	Execution of on-board operations control procedures
E4	Execution of goal-oriented mission operations on board	Goal-oriented mission replanning

operation of spacecraft relying on ground will inevitably increase the demand for resources and raise operating costs; (3) The requirement to enhance the autonomy of spacecraft to adapt to unknown operating environments is especially outstanding in deep space exploration. The autonomy of spacecraft is necessary due to the inability to adjust control strategies in real time after longtime latency of ground control. The attack-defense spacecrafts are required to respond rapidly to the actions of an attacked target; (4) In order to maximize the utilization of the spacecraft, such as remote sensing satellites, it is necessary to automatically analyze the satellite's observation coverage capabilities and lighting conditions to optimize the observation tasks and realize the continuous observation of the interested area; (5) Monitoring and handling of various types and levels of faults happened unpredictably in spacecraft need high-level autonomy.

With high-level autonomy, important requirements can be fulfilled, such as continuous mission product generation on board, realtime spacecraft control outside ground contact, maximization of mission objectives in relation to the available on-board resources, and robust operations in presence of on-board failures and context uncertainty [2]. Further drivers for increased on-board autonomy are the overall improvement of spacecraft availability and reliability [3].

There is no exact definition of autonomy of spacecraft as it varies with the different applications and different autonomy degrees. The ECSS-E-ST-70-11C [4] defines four levels of autonomy (Table 1).

From the above definition, we noticed that the level of autonomy depends on the distribution of operations between the ground segment and on-board segment. The higher spacecrafts are endowed with capabilities of understanding the situations and the goals, the more operations according to the situations and goals are realized on board, so that the spacecraft realizes high-level autonomy. Currently, most of spacecrafts are in E2 level which realize predefined sequences of on-board executable commands trigged by specific events or ground telecommands. A few spacecrafts for some earth observation missions and deep space exploration missions are in E3 level which realizes adaptive operations.

And autonomy can be categorized in general four application fields: intelligent sensing, planning and execution, fault protection and health management and distributed decision making [1].

In this paper, we propose a practical realization of the four levels autonomy within the mission planning and execution application filed. Firstly, Sect. 2 illustrates the model-based hierarchical layered software architecture of the autonomous system, and shows the methods to realize the four levels autonomy. Section 3 presents how these methods are used in a real spacecraft application and help realizing the mission planning and execution with high autonomy. Finally, Sect. 4 gives a conclusion and proposes work needed to be carried out in the future.

2 Autonomous System Software Architecture and Methods

2.1 Autonomous System Software Architecture

The On-Board Software (OBSW) plays an important role in an autonomous system. OBSW is more complicated as the autonomous degree becomes higher. The best way of handling increasing complexity is always the hierarchical layered architecture by decoupling the system according to the abstract function modules [2]. With respect to the hierarchical architecture proposed in [1], we proposed an enhanced hierarchical architecture with four layers in order to correspond to the four levels of autonomy here, as illustrated in Fig. 1.

(1) Decisional Level: this level aims to make plans for operational level according to distinguished models of the system after receiving the goals generated by telecommand, or sequences of the task, or reactions to the dynamic environment. The plans are the controlled sequences of configuration commands for subsystems of the spacecraft. For the purpose of coordinating different subsystems of the spacecraft, the decisional level also supervises the execution of the configuration command and maintains the resources and states of the whole spacecraft. In the decisional level, it needs the reasoning engine to obtain the configuration commands from current state to the desired state. Reasoning relies on the models of the system, while the reasoning method is domain independent.
(2) Operational Level: this level aims to decompose the configuration command into a controlled sequence of commands for operational mode transferring. This level can be interfered directly by ground telecommand.
(3) Function Level: this level takes in charge of controlling and supervising the status and actions of each subsystem in certain operational mode, both in normal and abnormal conditions. Function level supervises the actions, and if the actions are done incorrectly, it takes further solutions, e.g., does them again or gives feedback to upper level to do alternative strategy.
(4) Action Level: this level does the specific actions, which come from function level and conducted by certain components.

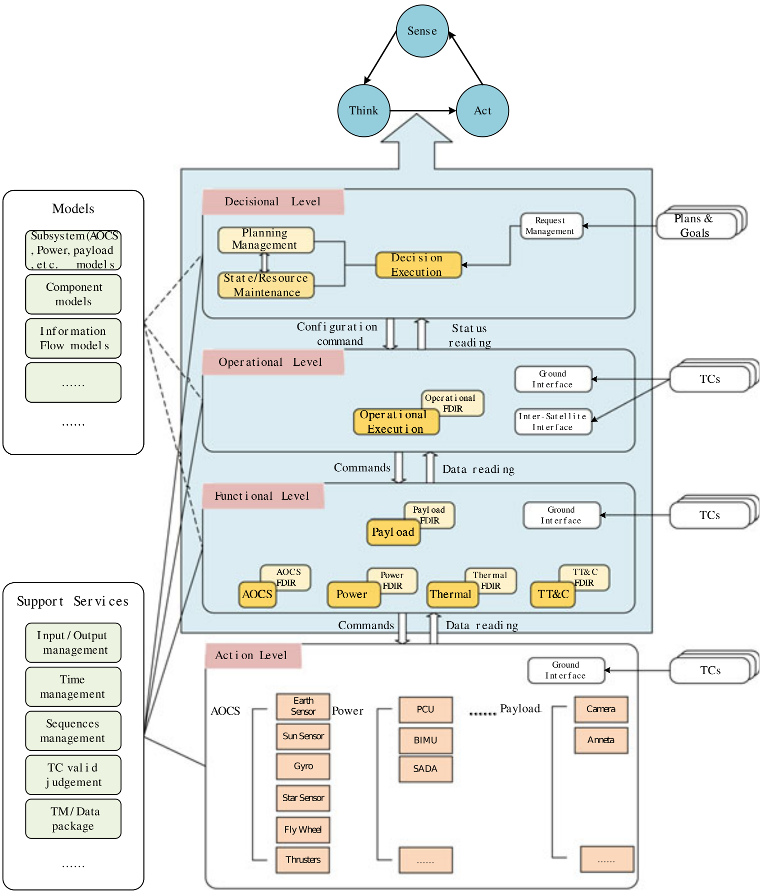

Fig. 1 The enhanced hierarchical architecture of OBSW of autonomous system

The upper three levels use sense-think-act [1] feedback control cycle to fulfill their respective functions and each of them has different degree of thinking. Sensing involves getting command from upper level or outside, and getting in-formation from lower level, and then mapping the command and the information to a representation that the software can use. Thinking involves making decisions about what to do according to the information form sensing. Meanwhile, acting involves carrying out actions relevant to each level [1]. The cycle is used to guarantee that the abnormal within each level can be handled by themselves. The OBSW also provides supported services such as input/output management, time management, telecommand and telemetry management etc., for all four levels.

2.2 Methods Used in Autonomous System

In this section, the methods used in each level to gain certain autonomy of mission planning and execution are introduced.

As described in Table 1, for E1 level whose mission execution is under ground control, Flight control Procedures (FcPs) [5] are used, in which the commands are directly provided to action level from ground, and telemetry data is downlinked to ground for human checking whether they are conducted correctly. This method is usually used in geostationary satellite while the satellite is visible all the time and telecommand and telemetry latency is extremely small. In E2 level, the plans are prestored, or ground defined and uplinked to the spacecraft in terms of time-tagged commands. It can be called Mission Time Line (MTL) [5]. The time-tagged commands are executed automatically by OBSW as time-tag expired. MTL is useful for predictable operations, but it cannot handle the abnormal situation in which the commands are not executed correctly and the sequence commands may not be appropriate thereafter. In order to solve this problem, the On-Board Control Procedures (OBCP) method is introduced which can be used for realizing E3 level autonomy [2]. Papers [2] and [6] illustrate how the OBCP method is realized in detail. The OBCP method uses the state machine to characterize the mission plan abstractly, and it is irrelevant to specific application. From users' perspectives, OBCP can be classified into two broad categories, that is to say On-Board Operations Procedures (OBOP, developed by the ground operations team) and On-Board Application Procedures (OBAP, developed by the spacecraft manufacturer in response to system requirements) [6] which is the abstract part irrelevant to the mission. The OBCP methods are widely used in deep space exploration such as Rosetta, Venus Express [7] and earth observation satellite such as Meteosat Third Generation Satellite [6]. Meanwhile, a method presented in paper [8] shows how to established an interactive control model to realize an E3 level autonomous control software in a Satellite Internet.

The E4 level is the highest level autonomy which realizes the goal-oriented mission planning and is also the most challenging. The methods for achieving E4 level autonomy is not a quite explored topic at both research and industrial areas [2] while there are already some efforts have been done. The MDP (Markov Decision Process) -based approach is a kind of model-based method used to implement highly autonomous reconfiguration for spacecraft mission plan. The theoretical basis of MDP-based method is the Markov decision theory. The space-craft configurations are treated as different states in the whole state space, the transition from one configuration to the other can be regarded as the Markov process, and the decision process used for handle the uncertainty in the outcome of actions is the Markov Decision Process. In practice, the most sophisticated part is to define the gain of actions and state transition probabilities which values most, such as the highest reliability or the highest performance. Because of the large scale of the whole state space, the BDP-based method used for mission planning can be calculated off-line on ground in certain degree, especially in some Fault Detection Isolation and Recovery (FDIR) strategies, it can be used to evaluate whether the process of recovery actions is optimal.

There are also some other model-based methods for mission planning and execution of Level E4 autonomy: Remote Agent Method for NASA's Deep Space One technology validation mission (DS1) [9], the autonomous reasoning engine (ARE) method proposed in paper [10], the multi agent system used for mission planning in deep space explorer mission [11] etc. Above all, the model-based method relies on the model of the spacecraft, either deterministic model or stochastic model, to handle the uncertainties in the environments or spacecraft itself and achieve the goal satisfying all the constrains of the system. Besides model-based methods, the artificial intelligence methods which do not need the model of the spacecraft can also be used [12]. But how to guarantee the high reliability of spacecraft needs more work to be done.

3 Realization of Mission Planning and Execution of High-Level Autonomy

In this section, we present how the methods proposed in the above section are used in a geostationary (GEO) meteorological satellite to realize the continuing observation with high-level autonomy without ground interference. Because of the similarity of the launch and early orbit phase (LEOP) of most GEO satellites, we only focus on the geostationary phase operational mode management which is customized based on the requirements of the GEO meteorological satellite.

3.1 Requirements of GEO Meteorological Satellite

The most benefit of GEO meteorological satellite is the high time resolution of observation. Continuing observation requires the on-board operation without interruption by ground because of which may not only influence the continuity of the observation but also introduce the probability of human error.

Because of the cooling need of the high resolution payloads of the GEO meteorological satellite, the satellite is designed with only one solar array on one side of Y direction of the satellite, and the other side is used for radiative cooling of the payloads, as illustrated in Fig. 2, which is the configuration of Chinese GEO meteorological satellite [13], for example. And the new generation of GEO meteorological satellite of the United States GOES-R has the same configuration. As a result, the sun pressure accumulation is extremely high and frequent unloading of fly wheels is necessary. The unloading frequency depends on the angular momentum envelope of the group of the fly wheels. On the other hand, GEO satellite needs station keeping to maintain in a fixed zone on GEO orbit not only avoid influencing other satellites on the same orbit but also provide a relevant stationary observation to the target on earth. Because unloading of wheels and station keeping operations needs propulsion,

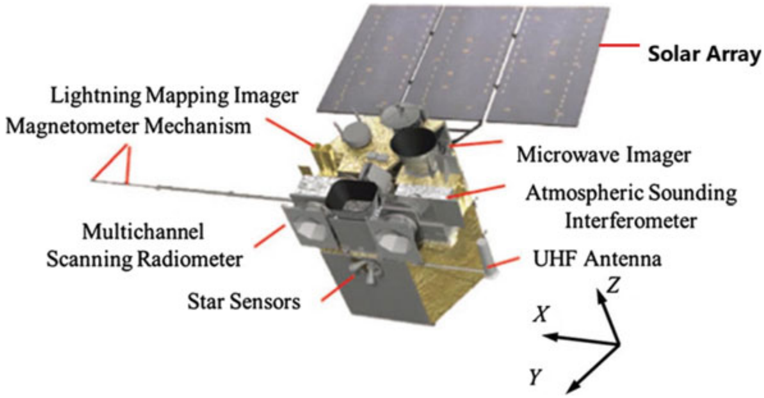

Fig. 2 The configuration of Chinese GEO meteorological satellite [13]

the design of the satellite must be sophisticated to reduce the influence of propulsion on the attitude which needs to be stable and accurate enough to satisfy the payloads observation requirements.

The most direct way to solve this problem is to use the small propulsion thruster, like electrical thruster, whose impact on the attitude is small enough to avoid negative effect on observation. As the thruster's force becomes smaller, the time takes longer in order to provide enough impulse for station keeping and enough angular momentum for flywheel unloading. Long time continuous operation needs high-level autonomy since ground operation is always not practicable because of high maintenance costs and probable human error.

3.2 Satellite Operational Mode Design and Transition

During the geostationary phase of satellite, the operational modes, which are shown in Table 2, are closely relevant to both the Attitude and Orbit Control System (AOCS) and payload mode throughout the mission lifecycle.

In Nominal Mode (NM), the AOCS works under Fine Pointing Mode (FPM) which realizes the high stability and high accuracy pointing along with +Z axis of

Table 2 Satellite, AOCS and payload mode combination during GEO phase (any = any operation, non-op=no operation)

Satellite operational mode	Nominal mode				Safe mode	
AOCS mode	FPM	FPM & Unloading	FPM & Station keeping	Yaw turning mode	SAM	ESM
Payload	Any	Any	Any	Non-op	Non-op	Non-op

the satellite to earth, while the satellite and payloads are fully operated. During FPM, unloading of flywheel and station keeping are realized simultaneously under certain frequency according to the strategies made by the software. During FPM mode, unloading or station keeping has no influence on payloads, and the changes among all these operations are all automatic. Because of cooling requirement of payloads, Yaw Turing Mode (YTM) is conducted at every equinox to make the cooling area of the satellite avoid sunlight exposure. During this period, the payloads stop working. Turing around the yaw axis can be conducted automatically because it is a predictable operation at the definite time.

The Safe Mode of the satellite is used to guarantee the satellite's energy or fuel or structure or payload in safe to avoid permanent termination of the mission of the satellite. In the Safe mode, there are two corresponding modes, Sun Acquisition Mode (SAM) and Emergency Stop Mode (ESM). The SAM realizes searching for the sun using thrusters and stable pointing to the sun using thrusters or fly-wheels. When pointing to the sun, the solar array can be directed to the sun along the definite direction to guarantee the energy of the satellite. The ESM stops all the control force and torque of the AOCS to make sure that the angular speed stops increasing, and the fuel or structure is safe in this situation. During the safe mode of the satellite, the payloads can be turned down or oriented to safe direction to avoid sunlight into the field of view of the payloads. As illustrated in Fig. 3, the safe mode can only be turned back to nominal mode by ground telecommand while the satellite FDIR software automatically monitor the status of the satellite and turned it into SAM or ESM.

3.3 Realization of Autonomous Operational Mode Management

Based on the hierarchical architecture illustrated in Sect. 2.1, the autonomous operational mode management of GEO meteorological satellite is realized as follows:

(1) Goal-oriented design of decisional level (E4 level of autonomy)

The goal of mission planning of GEO meteorological satellite is to maintain the high stability and pointing accuracy satisfying payloads needs during flywheel unloading and station keeping while thrusters are working. Because the precise model of thruster is hard to be established on ground and it varies as the working conditions e.g. temperature and pressure changes, the sense-think-act cycle acts an important role in realizing the autonomy of the mission planning:

Sense: the purpose of sensing is to evaluate the effect of the last act and provide information for "think" part to adjust the strategy or parameters for the following actions. Thus, during unloading, the flywheels' speed and attitude are recorded in order to adjust the controller parameters such as thrusters' torque, compensation torque and compensation time etc. Although these parameters may not be decoupled

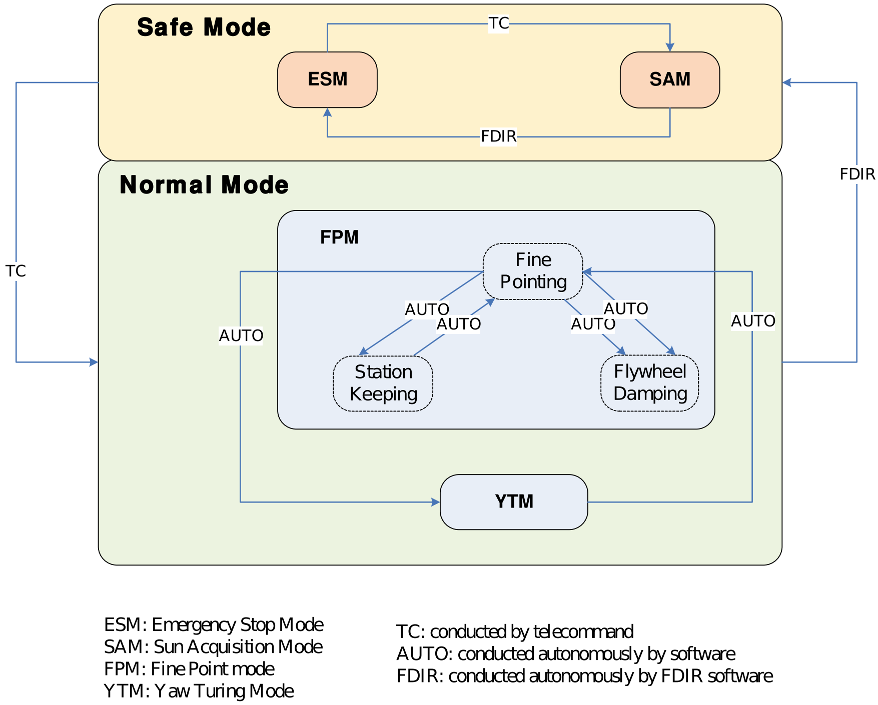

Fig. 3 Transition of the operational modes during GEO phase

from each other, the parameters can be adjusted automatically to achieve a comprehensive effect. And station keeping is realized in the same way, in which, the orbit and the attitude are used to evaluate whether the thrusters' propulsion are appropriate.

Think: the nature of "think" is to adjust the strategies to satisfy the goals, e.g., goal-oriented. It relies on the models of the system and can be achieved by Markov Decision Process, which is realized either off-line or on-line, depending on the realization complexity and on-board computer capability. The gain of state transition probabilities which values most are the highest control performance while flywheel unloading and orbit transferring. Another important role of thinking is to coordinate different subsystems of the satellite. For example, the electric thrusters cannot be used during earth shadow period because of high power consumption, and the mission planning strategy should be adjusted during the midnight around equinox time. Meanwhile, if the FDIR of certain subsystem is acting, the mission planning has to be adjusted or aborted according to the fault phenomenon.

Act: the act of decisional level refers to the configuration act rather than the specific commands conducted on the acting level. The strategy made by "think" part is formed in a sequence of configuration commands, such as flywheel unloading at midnight or start station keeping at the ascending node. "Act" part provides these configuration commands to the lower operational level for further actions.

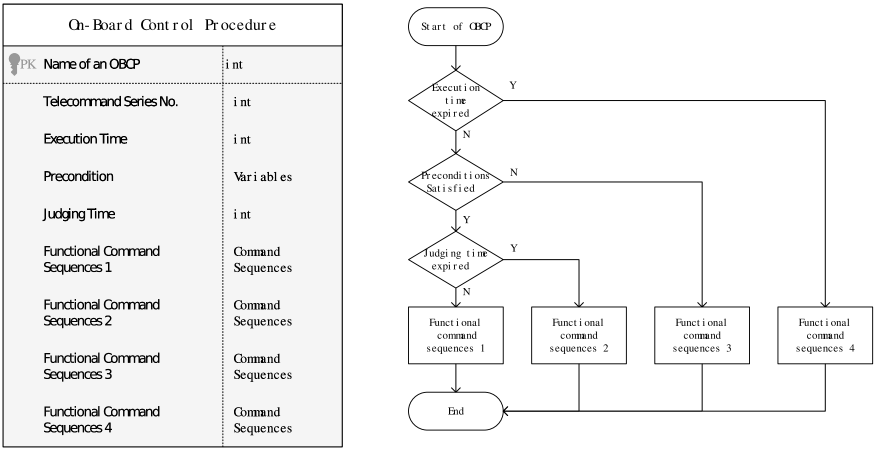

Fig. 4 The on-board control procedure flow map

(2) On-board control procedures (OBCP) design of Functional level (E3 level of autonomy)

The purpose of OBCP of operational level is to decompose the configuration command which is always relevant to the operational mode of different subsystems into functional commands which are used to control the specific function within the subsystem, and to make sure the decomposing procedure is entirely under control in normal and abnormal conditions. If the procedure cannot be conducted or is not conducted correctly, the OBCP should provide alternative solutions to handle this situation. As illustrated in Fig. 4, the attributes of the OBCP include: (a) procedure name; (b) corresponding telecommand number; (c) execution time which sets the expired time if the procedure is not conducted correctly or completely; (d) execution preconditions without which the procedure cannot be conducted; (e) judging time which sets the expired time for judging preconditions; (f) different kinds of execution sequences in different branches of the procedures. The execution states are continuously checked during the whole procedure and corresponding command sequences are provided so that it can also be regarded as a "sense-think-act" feedback loop cycle.

The OBCP can be activated by the configuration command and become inactive after execution (whether successful or not).

(3) Mission Time Line (MTL) design of Operational level (E2 level of autonomy)

Functional command sequences from upper level are commands within the subsystem to realize a specific function which can be realized by a time-tagged sequence of commands conducted by certain components, which can be regarded as Mission Time Line. For example, the function of starting firing of electric thruster can be decomposed into a sequence of time-tagged commands in series of opening the valve of supplying propellants, driving pressure regulation module, driving flow regulation module, and starting the electric excitation for electric thruster.

According to the model of the subsystem, the function of each subsystem is definite and can be stored before launch as the form of time-tagged command sequences. In the operational level, each command is conducted under supervisory of information from the components. If done incorrectly, the commands can be repeated for definite times which can be regarded as the feedback cycle of the same level. Meanwhile, the on-board software supervises the whole command sequences' state. If the sequence is not done correctly, the FDIR strategy recovers the state or just turns the subsystem into a safe state (which is another MTL), and gives feedback to upper level to make alternative strategies in operational level or decisional level.

The MTL can be conducted both by functional commands from upper level and telecommands from ground.

(4) Action level design (E1 level of autonomy)

Action level is the final fundamental level conducting the specific commands by specific component. After the action, the component provides its information back to upper level. In the action level, the command is conducted directly without supervisor. Therefore, the E1 level of autonomy is only within this level, and the feedback cycle is between the satellite and the ground.

4 Conclusion

This paper proposes a model-based enhanced hierarchical architecture of OBSW of autonomous system. Based on the proposed architecture, we present a method for realizing the mission planning and execution with four different levels of autonomy, and show how this method is used in a specific application. Thanks to the hierarchical architecture of software, the realization is organized with definite border between two related levels so that it is practical in engineering because of the decoupling of the realization details within each level. The method can be regarded as irrelevant to specific application and can be widely used in spacecrafts which need high reliability and autonomy.

We also noticed that the functions of the FDIR can also be organized as four levels in the architecture, and it supervises the execution of the mission plan in each level. The FDIR functions have great influence on the main stream of mission planning in abnormal conditions, and the recovery actions would be upgraded if done unsuccessfully, which means that the upper level would involve in the recovery if the lower level actions fail. It makes the whole mission planning and execution more complicated and further researches need to be done.

References

1. Jónsson, A.K., Morris, R.A., Pedersen, L.: Autonomy in space: current capabilities and future challenge. AI Mag. **28**, 27–42 (2007). https://doi.org/10.1609/aimag.v28i4.2066
2. Tipaldi, M., Glielmo, L.: A survey on model-based mission planning and execution for autonomous spacecraft. IEEE Syst. J. **12**, 3893–3905 (2018). https://doi.org/10.1109/JSYST.2017.2720682
3. Frost, C.R.: Challenges and opportunities for autonomous systems in space. In: National Academy Engineering U.S. Frontiers of Engineering Symposium, pp. 1–17 (2010)
4. Space Engineering-Space Segment Operability, European Cooperation for Space Standardization Standard ECSS-E-ST-70-11C (2008)
5. Tipaldi, M., Bruenjes, B.: Survey on fault detection, isolation, and recovery strategies in the space domain. J. Aerosp. Inf. Syst. **12**, 235–256 (2015). https://doi.org/10.1109/INDUSCON51756.2021.9529495
6. Tipaldi, M., Ferraguto, M., Ogando, T., Camatto, G., Wittrock, T., Bruenjes, B., Glielmo, L.: Spacecraft autonomy and reliability in MTG satellite via on-board control procedures. In: 2015 IEEE Metrology for Aerospace (MetroAeroSpace), pp. 155–159 (2015). https://doi.org/10.1109/MetroAeroSpace.2015.7180645
7. Steiger, C.N., Furnell, R., Morales, J.M.: On-Board Control Procedures for ESA's Deep Space Missions Rosetta and Venus Express (2005)
8. Jianglei, G., Xiaodong, H., Jinhui, Z., Bing, D., Kaifei, Y., Xiang, L.: Design and application of satellite autonomous control software based on model configuration. In: Aerospace Control and Application (2020). https://doi.org/10.3969/j.issn.1674-1579.2020.03.006
9. Bernard, D.E., Dorais, G., Fry, C., Gamble, E.B., Kanefsky, B., Kurien, J., Millar, W., Muscettola, N., Nayak, P.P., Pell, B., Rajan, K., Rouquette, N.F., Smith, B., Williams, B.C.: Design of the Remote Agent experiment for spacecraft autonomy. In: 1998 IEEE Aerospace Conference Proceedings (Cat. No.98TH8339), vol. 2, pp. 259–281 (1998). https://doi.org/10.1109/AERO.1998.687914
10. Bozzano, M., Cimatti, A., Guiotto, A., Martelli, A., Roveri, M., Tchaltsev, A., Yushtein, Y.A., Kessler, F.B.: On-Board Autonomy Via Symbolic Model-Based Reasoning(2008)
11. Rui, X., Pingyuan, C., Xiaofei, X.: Realization of multi-agent planning system for autonomous spacecraft. Adv. Eng. Softw. **36**, 266–272 (2005). https://doi.org/10.1016/j.advengsoft.2004.10.003
12. Zhao, P., Chen, Z.-M.: An adapted genetic algorithm applied to satellite autonomous task scheduling. Chin. Space Sci. Technol. **36**(6), 47–54 (2016). https://doi.org/10.16708/j.cnki.1000-758X.2016.0064
13. Liang, T.E., Shoulei, C., Kai, W., Yiwu, L.: Fengyun-4 attitude control system design and its in-flight performance. J. Spacecr. Rockets (2019). https://doi.org/10.2514/1.A34226