Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-5 (2016) 120–125

Language and Toolset for Visual Construction of Programs for Intelligent Autonomous Spacecraft Control

Andrey A. Tyugashev*.

**ITMO University, Saint Petersburg, 197101*
Russia (Tel: +79818214752; e-mail: tau@corp.ifmo.ru).

Abstract: The paper describes approach to Autonomous Fault Tolerant Intelligent Control of Spacecraft based on usage of Onboard Real-Time Interpreter of Integrated Control Programs, including special Diagnostic Routine. Rules of the Autonomous Control Program could be added or refined from Earth in operative manner by radio channel. Specially designed Visual Domain Specific Language allowing Control Logic Designers check, analyze and construct Rules in user friendly graphical environment excluding necessity to involve Software Developers. Proposed approach allows reducing of costs and labor consuming of Space Mission because of reducing of efforts needed for common-style Flight Control Software coding, multi-stage testing and support. Special Software Engineering Toolset that including Visualizer and Graphical Constructor of these autonomous control programs presented as well as the principles of its design and development. The Prototype of the Toolset has been successfully introduced at JSC Information Satellite Systems, Krasnoyarsk Region, Russia.

Keywords: Autonomous Control, Satellite Control, Diagnostic Programs, Software Tools, Software Engineering, Domain Specific Language, Visual Software Construction.

1. INTRODUCTION

The modern spacecraft/satellite is a technical complex integrating several kinds of systems such as Motion Control System, Power Supply System, Telemetry System, Thermal Control System, etc. Each of system, in turn, consists of devices, sensors, aggregates.Thus, it is not a wonder that many faults, failures and emergencies happens in real Space Missions, caused both by hardware and software errors and bugs. The cost of such an error is unacceptable high. We can have loss of very expensive satellites, results of efforts of many scientists, engineers, technicians (who can represent different countries and participate in a Space Project for years). Thus, the maximum efforts are in use to avoid these catastrophes.

1.1 Role of Flight Software in Fault Tolerance of Space Missions

The possible abnormal situations are a subject of detailed pre-mission engineering analysis, and the Control Logic for the Satellite Recovery in case of analyzed Abnormal Situations should be specified in the corresponding documentation. In manned missions, these documents must be carefully learned by cosmonauts. For an Unmanned Automatic Satellites, the role of Onboard Autonomous Control is extremely important. In case of abnormal situation, Ground Personnel could not have enough time to understand the situation, make right decisions and transmit correct commands to a Satellite. This is a good reason for advancing in development of autonomous control means.

Nowadays, even the nanosatellites with the mass less than 10 kilogram and overall dimensions less than one meter, have an Onboard Computer(s) (see Eickhoff 2012). The Onboard Control System is one of the most complicated Onboard Systems of the Satellite. Herewith, Satellite Control Logic is being implemented by Onboard Software. Thus, there is straight dependence between correctness and fault tolerance of the Onboard Software and overall Space Mission's success. Meanwhile, as a result, when we evaluate labor and time consumption, and total costs, the typical proportion between hardware and software of the Onboard Control System can be characterized as 1:10. In aerospace projects (as it was noticed decades ago, for example see Koczela L.I. and Burnett G.I., 1968 and Tomayko 1988, 1994 and 2000), the processes of design, development and verification of Onboard Software became a 'critical path' on network schedule embracing all works connected with a design and manufacturing of Rocket/Space System as a whole. Any software module must go through many formally specified stages which are obligatory for development of Mission Critical Software. These stages include specification, unit testing, integrate/system testing, acceptance testing on special Integrated Test bed complex, preparation of Program Documentation. Any changes lead to repeat of the named stages. The total cost of Onboard Software Lifecycle dramatically grows because of required software maintenance efforts (see Krasner and Bernard, 1997). Another problem is correspondence between the version of program and its documentation.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.100

1.2 State of the Art in Intelligent Autonomous Satellite Control

The serious problem in Mission Critical Software development is a misunderstanding between participants of the Project – operations engineers, specialists of certain onboard equipment, Control Logic Designers, and programmers. Moreover, it is not a trivial thing to find a concrete operator(s) in program written in C or Java programming language which implements particular logic. We can decrease level of influence of 'Broken Telephone' effect by excluding programmers from a process (James Martin, 1982).

The very promising way to do this is an attempt to realize some sort of "Onboard Intelligence" with the flexible responding to the actual situations and decision making. The specially designed Intelligent Diagnosis Software may be considered as a part of the Onboard Intelligence providing Fault Tolerance Autonomous Control (Hayes-Roth 1995). There are several known approaches to do this, including neural networks, fuzzy decision making systems, knowledge bases and expert systems (Grabot et al, 1993).

The problem with usage of neural networks for Onboard Intelligence is necessity of machine learning process. This process could require many "trial-and-error" attempts. But the Satellite Control is a typical Mission Critical System. Satellite can be an element of Global Commercial or Government Communication System or even has Military purposes. We cannot allow any errors at the exploitation stage. We can practically consider pre-flight neural network learning only. But in this case we still have the insufficient level of confidence to control because of lack of well-defined, documented procedure of decision making. The problem also connected with «dispersion» of decision making Rules among the network, leading to impersonality in responsibility. But in Space Mission the issue of personal responsibility in case of emergency has a high value (for example, see Finn 2008).Pattern-based principles of neural network learning are not enough corresponds to this problem domain at all because a number of patterns used for learning are restricted. But the typical expected lifetime of the modern Satellite is 10-15 years. Thus, there is a non-zero probability of appearing a Abnormal Situation which was not used as a pattern. We need a means that allow refinement of the Onboard Control Rules during the flight. The similar issues can be addressed to fuzzy logic based systems.

The Real-Time Expert Systems (also named 'dynamic expert systems') potentially look more promising. There are several commercial products developed in this area – RTworks (Talarian), G2 (Gensym), FAIN (Nippon Steel), RTXPS (ESS). G2 even incorporates idea of Visual Language for description of Business Processes. Some of them have a success stories in Mission Critical Applications But as far as author know (we need consider security/secrecy issues actual for Space Industry), these systems used in Control Systems on Earth and there are no cases of use onboard in Space.

As it was specified before, the knowledge about Abnormal Situations and needed actions is owned by Space System Engineers and Control Logic Designers. We need means for

a. Convenient collection of this knowledge;

b. Flexible implementation in Onboard Software.

This problem is a particular case of very important and well-known problem in Intelligent Control and Expert Systems design – problem of Knowledge Acquisition (see Chassiakos, 2005, Grabot 1996 and 2014, Lugerand and Stubblefield 1989, Ruiz et al., 2014, Shadbolt et al, 1996). The work of Smith, Rajan, and Muscettola, 1998 is directly related to an Autonomous Spacecraft Control.

At the JSC Information Satellite Systems, the very advanced and flexible technology with elements of intellect has been successfully implemented in this problem domain (Khartov, 2006, Koltashev, 2006).Every several seconds, Onboard Dispatcher runs special Diagnostic Program that compares particular State Vectors of Satellite with the predefined patterns (Kochura, 2011).

Diagnostic routine named "DKD" (Russian "Duty Control and Diagnostics") must:

a. Detect the Abnormal Situation.

b. Execute a set of correctly synchronized actions to parry a fault or at less to provide switching the Satellite into a 'safe mode' without catastrophic consequences.

The "DKD" program organized as a set of Rules. Each Rule joins a State Vector and needed Actions. Herewith, the specially designed Domain-Specific Language (DSL) is being used to specify Rules. This language is not similar to C, Fortran or Java and could be easily understand by non-programmers. The construction of the rules is an interactive process supported by special "REAL" Programming System presented in Fig.1. Of course, in this case, we need a special Software Tool - translator from the DSL into language acceptable to Satellite's Onboard Computer. In JSC Information Satellite Systems, the translator implemented as Onboard Real-Time Interpreter. The program that must be interpreted and executed, can be uploaded onboard in very flexible operative manner.

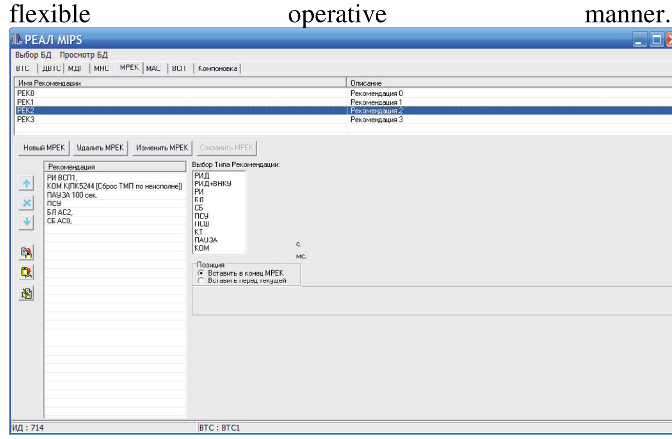

Fig. 1. Screenshot of "REAL" Programming System.