On use of Satisfiability Modulo Theories approach for evaluation of Real-Time spacecraft control logic

A.A. Tyugashev[1]

[1]Samara State Transport University, Svobody street 2V, Samara, Russia, 443066

Abstract. Use of SMT solvers is a very promising modern approach successful in different application domains. The paper is devoted to study how the functionality provided by SMT solver can be utilized for evaluation of key parameters of spacecraft's control logic. The modern spacecraft is a complicated complex of technical complexes should function in consistent matter like an 'orchestra'. Use of SMT in this problem domain is based on formal system of the real-time control and the semantic model of real-time control logic. A Formal specification of real-time control could be feasible or non-feasible on the defined basis of functional tasks (dependable on the parameters of the task, including duration). The feasibility can be checked using SMT approach. As an example, use of Z3 SMT Solver in specially developed Java application through API is discussed.

Keywords: Satisfiability modulo theories, spacecraft, Real-Time control, control logic, onboard device

1. Introduction

The modern technical object such as airplane, submarine, spacecraft, nuclear power station can be reviewed as ‘system of the systems’ which includes a lot of subsystems, actuators, sensors, devices. All of these devices should co-function in harmonic manner to produce a useful outcome like an orchestra playing symphony. Each instrument must start play at a right time. In orchestra, the control functions are executed by a conductor. In modern complex technical complexes, the control system should provide that functionality. The human could be involved in the process in case of automated control, or not be involved in case of automatic system. Moreover, in according to Ahby’s Law of Requisite Variety [1], “Variety absorbs variety”, so the complexity of control system should be adequate to a complexity of controlled object. Control algorithms must implement the right ‘control logic’, i.e. coordinated functioning of the all units. The ‘coordinated’ word means both a semantic coordination related to physical restrictions and logic of actions and coordination in time. The time characteristics of the control logic should be adequate to the speed of ongoing physical processes associated with the controlled technical complex [2-5].

The very important problem for control logic of complex technical object is evaluation of its parameters and checking if these values are correspond to existing physical and technological constraints. This problem is quite actual both at design stage when the key question is feasibility of the required system and even during operation of the existing technical object when we need to analyze

performance, for example. This paper is focused on timing (synchronization) parameters, and use of accessible resources (level of workload/overload). The problem has an additional importance due to its straight connection to dependability/safety issues of spaceflights.

Today, as a rule, the evaluation process is being performed by a human. Unfortunately, the number of parameters of real modern spacecraft must be analyzed, for example, can be very big and exceed the human opportunities.

So, it is a potentially useful approach to apply existing automation tools to provide some assistance to specialists who are responsible for control logic evaluation [6-7]. The very popular and promising technology today is Satisfiability Modulo Theories (SMT) approach supported by a lot of commercial and free solvers such as ABSolver, Alt-Ergo, Barcelogic, MathSAT, CVC, OpenSMT, Simplify, STeP, Yices, Z3, etc. We can specify the existing constraints using smt-lib formal language, and then get the answer if the system satisfies (`sat`) the constraints, or not (`unsat`). The system even can calculate the values of the variables which provide satisfiability.

2. Models and Method

2.1. Mathematical model of Real-Time Control Logic

In previous papers [3,6,7] author proposed the semantic model for real-time control algorithms. The model represents control actions by the set of following:

$$RTCL = \{ < f_i, t_i, \tau_i, l_i > \}, i=1..N \quad (1)$$

f_i represents an ID of functional process (FP) to be executed, and: t_i – time of f_i begin (non-negative integer), τ_i – its duration of the (non-negative integer). l_i is a 'logical vector' defining if the process should be executed. The logical vector is formed by the logical variables with its values: ($\alpha_1=0$, $\alpha_2=1$, $\alpha_3=0$, $\alpha_4=H$, $\alpha_5=H$). Herewith, 1 and 0 corresponds to `True` and `False`, and '`H`' value means that execution of the process is not depends on value of this logical variable. The presence of logical variables in the model allows specifying a set of options of implementation of the algorithm (including normal and abnormal situations).

Some parameters can be specified by a known constants, some be initially unknown and stated as variables.

The constraints and requirements for the Real-Time control logic can be specified using language of CA formal theory (calculus of real-time control algorithms) proposed by A.A. Kalentyev [3-4]. The extended version of this theory developed by author [6] contains the following operators allowed for use in specification (see Table 1). There are also 'soft' bindings: <. << and <>. Special operator <*l*> means logical incompatibility of actions, i.e. the processes cannot be found in the same case of execution. This is means that the same logical variables has 1 value in one vector, and 0 in another.

Table 1. Operations of the extended algebraic model of real-time control algorithms.

Name	Mean	Signature
CH	'begin-begin' synchronization	$(UA_1, UA_2) \rightarrow UA$
CK	'end-end' synchronization	$(UA_1, UA_2) \rightarrow UA$
$\rightarrow$	direct following	$(UA_1, UA_2) \rightarrow UA$
H	parameterized overlay	$(UA_1, UA_2,$ integer$) \rightarrow UA$
3A	parameterized following	$(UA_1, UA_2,$ integer$) \rightarrow UA$
@	absolute time binding	(UA, integer) $\rightarrow$ UA
$\Rightarrow$	qualification by the logical condition	(condition, UA) $\rightarrow$ UA

These formal calculi are strong associated with algebraic models or real-time control algorithms [6]. As is was presented in [7], in some cases with the preset values of the model variables, wanted specification of the control actions may be feasible, but the same set of requirements and constraints may be unfeasible with the other values.

Example 1. For the following synchronization requirements:
$f_1\ CH\ f_2$
$f_1 \rightarrow f_3$
$f_4\ CK\ f_5$
$f_3 \rightarrow f_4$
$f_2 \rightarrow f_5$,
and values $\tau_1 = 20$, $\tau_2 = 100$, $\tau_3 = 200$, $\tau_4 = 10$, $\tau_5 = 50$, the specification is not feasible due to impossibility of fulfillment of $f_2 \rightarrow f_5$ formulae, this fact can be visually verified in Figure 1.

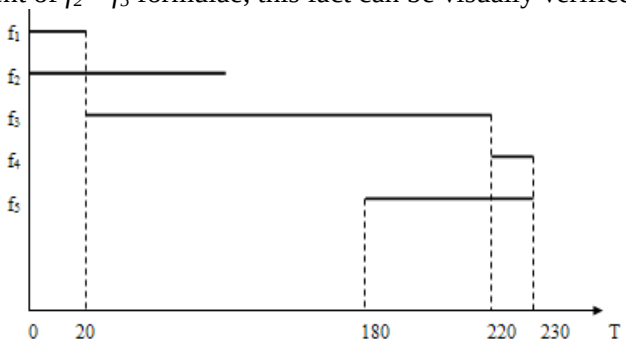

Figure 1. The first example of feasibility checking.

But if we have the another parameters, for example, $\tau_1 = 100$, $\tau_2 = 150$, $\tau_3 = 70$, $\tau_4 = 10$, $\tau_5 = 50$, specification becomes feasible (see Figure 2).

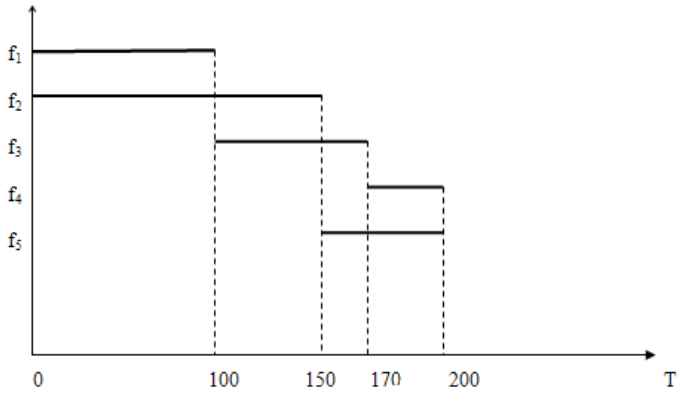

Figure 2. The other combination of values allows feasibility.

The very important point is that the model above can be applied not only for real-time onboard flight control software it was initially developed for, but for representation of any sort of activity/processes performed by human, technical devices, processors, mechanisms, etc. Indeed, the presented model is independent on nature of implementer of functional process. On the other hand, the model allows representing adequately complexity of Real-Time control actions in 'time space' and 'logical space'.

2.2. Ways of utilization of SMT solvers functionality

The most common of mathematical objects and constructs such as rational and real numbers, vectors, and matrix are already supported by existing SMT solvers. So, if we will transform requirements to real-time control logic into requirements to these kinds of entities, we will have possibility to utilize functionality of SMT solver.

To do this we may use following way from the formulas and relations between the functional processes to equations and inequalities on the numbers.

Table 2. Associations between the control logic requirements and numbers related inequalities and equations.

RTCL formulae	entails	comment
$f_i\ CH\ f_j$	$t_i = t_j$	equation of numbers
$f_i\ CK\ f_j$	$t_i + \tau_i = t_j + \tau_j$	equation of numbers
$fi \rightarrow f_j$	$t_i + \tau_i = t_j$	equation of numbers
$3A(fi, fj, \delta)$	$t_i + \tau_i + \delta = t_j$	equation of numbers
$H(fi, fj, \delta)$	$t_i + \delta = t_i$	equation of numbers
$fi < fj$	$t_i < t_j$	inequality of numbers
$fi << f$	$t_i + \tau_i < t_i$	inequality of numbers
$f_i <> f_i$	$t_i + \tau_i < t_j \vee t_j + \tau_j < t_i$	disjunction of inequalities
$f_i <l> f_j$	set of boolean equations	logical incompatibilities of FPs (see above)

This way allows us to formulate in notions of smt-lib language.

2.3. Program implementation

Many of the free accessible SMT solvers provide API for integration with user developed software. Moreover, Z3, for example, can be executed online via Internet. It is allowed us to utilize SMT solver functionality for practical goals in control logic problem domain. Special program prototype was developed to validate applicability of the describing approach.

The prototype written using Java 8 and has intuitive graphical user interface (see Figure 3, in Russian).

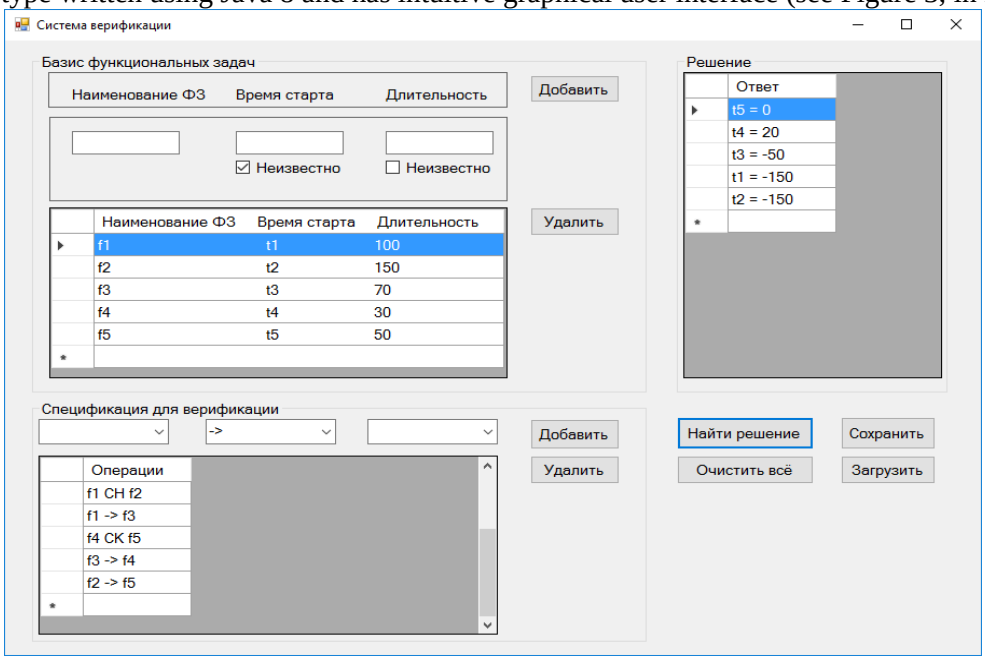

Figure 3. Screenshot of developed program prototype.

User can set known values of model variables in dedicated editor fields. There are also special field on form for specification of requirements in terms of control logic, buttons for solving (trying to evaluation of feasibility – `sat` or `unsat`, and finding parameters which are make the set of requirements feasible).

3. Conclusion

The paper describes how the Satisfiability Modulo Theories approach in couple with mathematical models of real-time control logic based on previously developed of A.A. Kalentyev and author formal theories and algebraic models of real-time control algorithms, can be applied in problem domain of complex technical object control. The method of transition from the formal requirements to functional

processes in terms of RTCL formulas to SMT compatible equations and inequalities is being described. The prototype of specially developed program tool presented as well. This method can allow automate evaluation of the key characteristics of spacecraft flight control logic at the design phase and check some safety issues of existing control logic.

4. References

[1] Ashby, W.R. Introduction to Cybernetics – NY: Chapman & Hall, 1956.

[2] Sollogub, A.V. Principles of the Earth Observation Satellites Control in Contingencies / R.N. Akhmetov, V.P. Makarov, A.V. Sollogub // Information and Control Systems – 2012 / Vol. 1, 2012, PP. 16-22.

[3] Tyugashev, A.A. Integrated environment for designing real-time control algorithms / A.A. Tyugashev // Journal of Computer and Systems Sciences international – 2006 / Vol. 2(45), P. 287-300.

[4] Tiugashev, A.A. Application of CALS technologies in Lifecycle of Complex Control Software / A.A. Kalentyev, A.A. Tiugashev – Samara: Samara Centre of RAS Publishing, 2006 – 266p. – (in Russian).

[5] Tyugashev, A.A. Ways to improve quality and reliability of software in aerospace industry / A. Tyugashev, I. Ilyin, I. Ermakov // Large-Scale Systems Control – 2012 / – Vol. 39. – P. 288–299 – (in Russian).

[6] Tyugashev, A. Use of graph-based and algebraic models in lifecycle of real-time flight control software // Proc. the Mathematical Modeling Session at the International Conference Information Technology and Nanotechnology (MM-ITNT 2017) Samara, Russia, 24-27 April, 2017., Edt. by S. Sazhin, D. Kudryashov et al. [Electronic resource]. — Access mode: http:// http://ceur-ws.org/Vol-1904/ (19.11.2017).

[7] Tiugashev, A. Build and evaluation of real-time control algorithms in case of incomplete information about functional processes' parameters // Proc. 2017 XX IEEE International Conference on Soft Computing and Measurements (SCM’2017), Saint Petersburg, Russia, 24-26 May, 2017, PP 179-185.