# AN APPROACH TO VALIDATION PLANS FOR SATELLITE CONTROL GENERATED WITH PLANNERS SYSTEMS BASED AUTONOMOUS ARTIFICIAL INTELLIGENCE

**André A. de Souza Ivo[1], Mauricio G. Vieira Ferreira[1], Nilson Sant'Anna[1], Ana Maria Ambrosio[1]**

[1]Instituto Nacional de Pesquisas Espaciais (INPE), Brazil

andre.ivo@inpe.br, mauricio@ccs.inpe.br, nilson@lac.inpe.br, ana@dss.inpe.br

## 1. INTRODUCTION

The scheduling or autonomous planning is one of the areas of artificial intelligence (IA), which develops the process to define the sequence of actions to achieve the objectives established through computational methods.

Through an initiative of the ESA (European Space Agency) emerges the SPASS (Software Product Assurance for Autonomy Spacecraft on-board), a project that resulted in a job that recommends extensive testing and use of simulation techniques, with application of two software components, the plausibility checker and bag security, proposed by (Blanquart, 2004).

The purpose of this paper is to show a theoretical approach to implemente a plausibility checker and safety bag as described by (Blanquart, 2004).

## 2. SOFTWARES COMPONENTS

According to the recommendations proposed in the work "Software Safety Supervision On-board Autonomous Spacecraft" (Blanquart, 2004) two generic software components are defined:

- **Plausibility checker**

  The plausibility checker is a software component responsible for supporting the validation of plans in ground stations before they are sent to systems in space environment, as shown in architecture Figure 1.

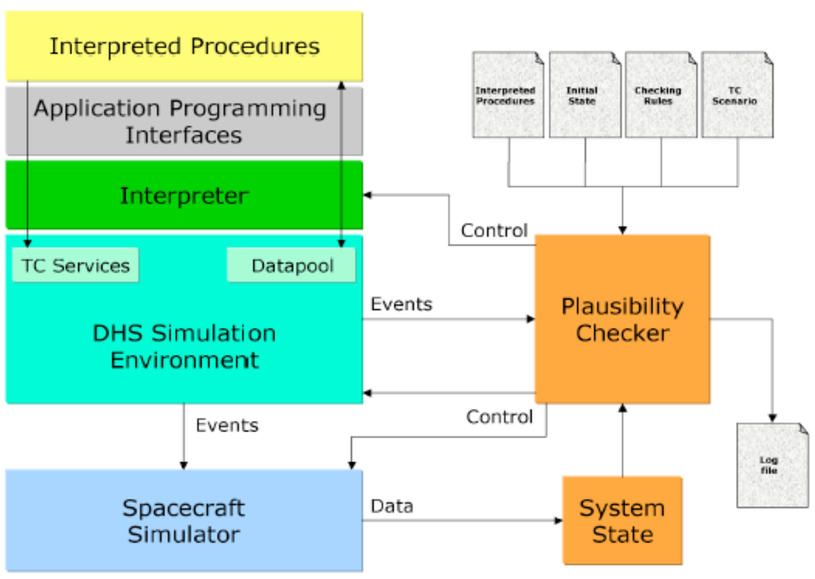


**Figure 1 - Architecture of plausibility checker (Blanquart, 2004)**

- **Safety Bag**

  This component will be responsible for supporting the decision of the onboard computer for execute the flight plan based on current conditions. architecture of the safety bag can be observed in Figure 2.

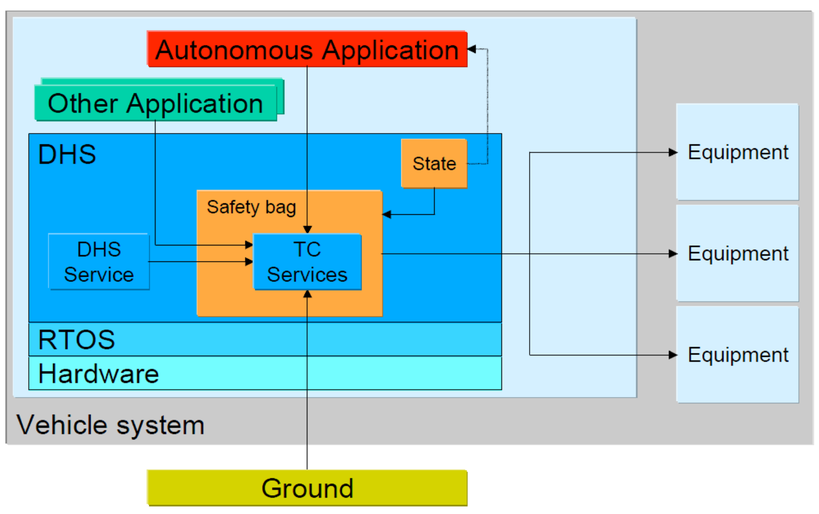


**Figure 2 - Architecture of safety bag (Blanquart, 2004)**

## 3. APPROACH FOR VALIDATION OF PLANS

The proposed makes use of Finite State Machines (FSM) to model the planning and its various possibilities of states and transitions.

The application of such techniques has focused on the validation of flight plans of satellites from INPE, however, for this article was used as a case study a classical problem of planning, The World of Blocks.

The proposed approach makes use of FSM techniques for modeling and generating the validation of plans generated by autonomous systems. The Figure 3 shows the beginning of the modeling to the problem of case study.

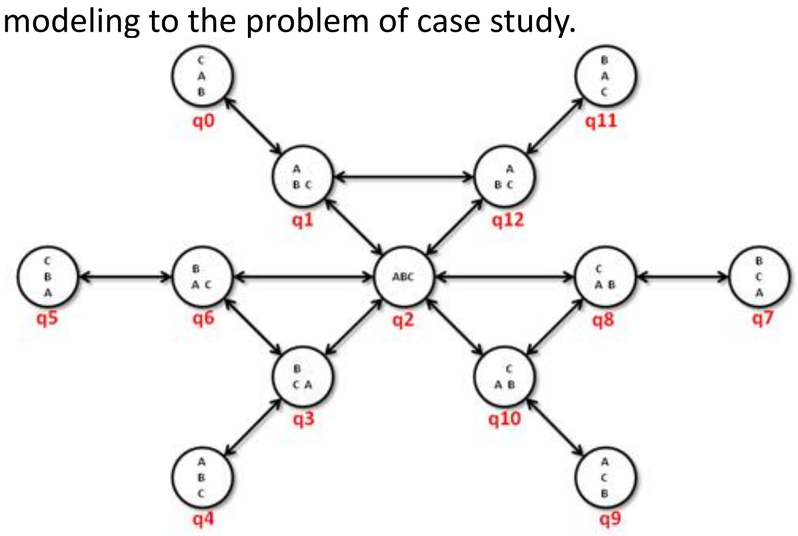


**Figure 3 - Graph of states of a plan to the problem of The World of Blocks**

In the example shown in Figure 3, the transition table generated can be observed in Table 1.

**Table 1 - Table of transition states**

| States / Events | #A | AB | AC | #B | BA | BC | #C | CA | CB |
|---|---|---|---|---|---|---|---|---|---|
| q0 | q0 | q0 | q0 | q0 | q0 | q0 | q1 | q0 | q0 |
| q1 | q2 | q1 | q12 | q1 | q1 | q1 | q1 | q0 | q1 |
| q2 | q2 | q1 | q12 | q2 | q6 | q3 | q2 | q8 | q10 |
| q3 | q3 | q4 | q3 | q2 | q6 | q3 | q3 | q3 | q3 |
| q4 | q3 | q4 | q4 | q4 | q4 | q4 | q4 | q4 | q4 |
| q5 | q5 | q5 | q5 | q5 | q5 | q5 | q6 | q5 | q5 |
| q6 | q6 | q6 | q6 | q2 | q6 | q3 | q6 | q6 | q5 |
| q7 | q7 | q7 | q7 | q8 | q7 | q7 | q7 | q7 | q7 |
| q8 | q8 | q8 | q8 | q8 | q8 | q7 | q2 | q8 | q10 |
| q9 | q10 | q9 | q9 | q9 | q9 | q9 | q9 | q9 | q9 |
| q10 | q10 | q10 | q9 | q10 | q10 | q10 | q2 | q8 | q10 |
| q11 | q11 | q11 | q11 | q12 | q11 | q11 | q11 | q11 | q11 |
| q12 | q2 | q1 | q12 | q12 | q11 | q12 | q12 | q12 | q12 |

From the transition table is constructed the state machine shown in Figure 4, using Mealy for representation.

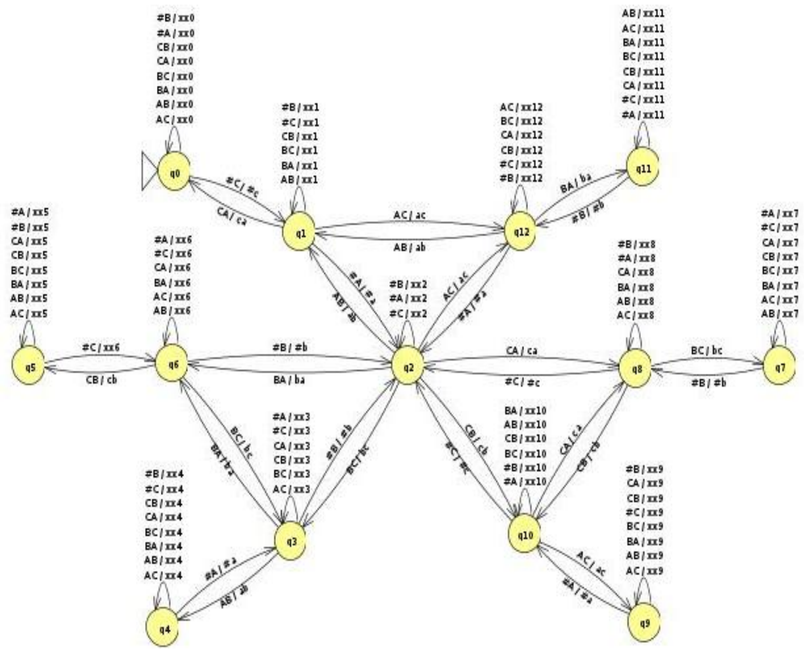


**Figure 4 - Mealy FSM representing the problem**

The generation of plans for validation has been performed with the tool JPlavis FSM. (Pinheiro, 2011).

The tool JPlavis FSM's main objective is the automatic generation of tests from a finite state machine. The generation is based on algorithms of reduction the FSM known such as method W (Chow, 1978), Wp (Fujiwara, Von Bochmann, Khendek, Amalou, & Ghedamsi, 1991), UIO (Sabnani & Dahbura, 1988) , HSI (Petrenko, Yevtushenko, Lebedev, & Das, 1993) e SPY (Simão, Petrenko, & Yevtushenko, 2009).

Given the initial state of the system the tool JPlavis FSM is capable of generating all possible plans for FSM described. The results obtained for the case of study, the World of blocks, can be observed in Table 2.

**Table 2 - Test results generated by JPlavis FSM**

| Method | Plans |
|---|---|
| W | 210 |
| HSI | 129 |
| SPY | 44 |
| UIO | 117 |

## 4. CONCLUSION

The algorithms generated all possible plans for FSM described, regardless of the end state. However to meet the main requirement of the planner (from an initial goal, shall be able to plan to achieve the final goal) it is necessary that the algorithms are changed including the restriction to be selected only the plans that include final goal.

The plausibility checker would be completed with this implementation and can be used to verify the plans on the ground stations.

The method for the plausibility checker can be used in the safety bag. However, the approach changes slightly due to the dynamic characteristic of the, the purpose of including a safety bag on-board computer is to make the actions of the plan are evaluated in real time, therefore the initial state is changed, assuming is position.

The validation should not be performed in the plan as a whole, but only on the next action.

To this should be evaluated if the next action that the system want to run is within a valid set of actions that should be checked against plans generated by the method.

## 5. BIBLIOGRAPHY

•Blanquart, J.-P. (2004). Software Safety Supervision On-board Autonomous Spacecraft. 2nd European Congress ERTS, EMBEDDED REAL TIME SOFTWARE - 21-23 January 2004, Toulouse, France .

•Cardoso, L. S. (2006). An Intelligent System for Generation of Automatic Flight Operation Plans for the Satellite Control Activities at INPE. 9th Conference on Space Operations (SpaceOps-2006) .

•Chow, T. S. (1978). Testing software design modeled by finite-state machines. IEEE Transactions on Software Engineering, v.4 .

•Fujiwara, S., Von Bochmann, G., Khendek, F., Amalou, M., & Ghedamsi, A. (1991). Test selection based on finit state models. EEE Transactions on Software Engineering, v.17 .

•Kucinskis, F. d. (2007). Alocação dinâmica de recursos computacionais para experimentos científicos com replanejamento automatizado a bordo de satélites. São José dos Campos, SP, Brasil: INPE.

•Petrenko, A., Yevtushenko, N., Lebedev, A., & Das, A. (1993). Nondeterministic state machines in protocol confance testing. Proceedings of the 6th International Workshop on Protocol Test systems .

•Pinheiro, A. C. (2011). Subsidies for the application of state machine based test case generation methods. São Paulo, SP, Brasil: Universidade de São Paulo (USP).

•Sabnani, K., & Dahbura, A. (1988). A protocol test generation procedure. Computer Networks and ISDN Sytems, v.15 .

•Simão, A., Petrenko, A., & Yevtushenko, N. (2009). Generation Reduced Test for FSMs with Extra States. Proceeding TESTCOM '09/FATES '09 Proceedings of the 21st IFIP WG 6.1 International Conference on Testing of Software and Communication Systems and 9th International FATES Workshop .

•Souza, P. B. (2011). Uma estratégia baseada em algoritmos de mineração de dados para validar plano de operação de voo a partir de predições de estados dos satélites do INPE. São José dos Campos, SP, Brasil: INPE.