



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-14072-PRE/9241

**AN INTELLIGENT SYSTEM FOR GENERATION OF
AUTOMATIC FLIGHT OPERATION PLANS FOR THE
SATELLITE CONTROL ACTIVITIES AT INPE**

Luciana Seda Cardoso
Maurício Gonçalves Vieira Ferreira

Article Published in the Proceedings of the 9th Conference on Space Operations
(SpaceOps-2006) 19 – 23 June, 2006, Rome, Italy.

An Intelligent System for Generation of Automatic Flight Operation Plans for the Satellite Control Activities at INPE

Luciana S. Cardoso, Mauricio G. V. Ferreira, and Valcir Orlando

National Institute for Space Research – INPE, São José dos Campos, São Paulo, 12227-010, Brazil

Email: luciana@dss.inpe.br, {mauricio, valcir}@ccs.inpe.br

Research of automatic solutions for space operations is a real need for all space agencies in order to reduce space mission costs. Nowadays, a significant parcel of satellite operation activities at the National Institute for Space Research (INPE) is still performed manually. Thus, finding automated alternatives for the satellite operation activities at INPE is of capital importance, in order to maintain the currently satisfactory performance of these activities, despite the scarcity of financial resources. This paper proposes the architecture of an Intelligent Planning System for the automatic generation of satellite flight operation plans (PlanIPOV). The proposed system employs temporal planning techniques of artificial intelligence (IA) in the automatic flight operation plans (FOP) generation for a satellite routine operational phase, with the aim of opening the way toward a higher degree of automation for satellite operation activities of INPE. The main reason for the application of the planning system in the routine operational phase of the satellite lifespan is that this phase is composed of very repetitive and well defined tasks which have lower programming costs. In addition, this phase is the longest comprising practically the entire lifespan of the satellite. The PlanIPOV system uses the Planning Domain Definition Language (PDDL2.2) to model the knowledge base of INPE satellite operations. It is based on the automatic generation of problem files, i.e., the initial state of the satellite control environment and the goal to be reached by executing the generated FOP timeline. The system uses information extracted from the following files: tracking knowledge domain; prediction of future satellite passes of the involved ground stations; and configuration parameters of the satellites and the ground stations. This paper also presents a prototype of the proposed planning system. This prototype was implemented using the PDDL2.2 language and LPG-TD planner (Local Search for Planning Graphs – Time Initial Literal and Derived Predicates). It was tested for the tracking domain of the satellites currently being controlled by INPE (SCD1, SCD2 and CBERS2). These results are presented along with the limitations that have been observed in the planning technique application. The solutions adopted to overcome these limitations are outlined. The obtained results may be considered satisfactory.

I. Introduction

One of the main aims of the National Institute for Space Research (INPE) is the domain of space technology through the development and operation of artificial satellites. Due to the extension of the Brazilian territory, satellites technology constitute an important tool for some applications of great interest for the country as, for example, the monitoring of native vegetation, agriculture, rivers, climate; survey of natural resources; and telecommunications.

At the moment, INPE is controlling three satellites: SCD1, SCD2 (Data Collecting Satellites) and CBERS-2 (China-Brazil Earth Observation Satellites). For the next years the following satellite launches are foreseen: two scientific micro-satellites (EQUARS -Equatorial Atmosphere Research Satellite; and MIRAX - X-Ray Monitoring and Imaging), two remote sensing satellites (SSR-1 and SSR-2), that will use a multi-mission platform currently being development at INPE. Moreover, the agreement with China was renewed for the development and launch of three more remote earth observation satellites (CBERS-2B, CBERS-3 and CBERS-4). As a consequence of these future missions a great additional charge in terms of responsibility new activities will be imposed to the satellite operation

people. However, a significant amount of the satellite operation activities at INPE is still performed manually. Taking into account the scarcity of financial resources and the increase of the number of satellites to be controlled, automated alternatives for the satellite operation activities is becoming a particular very attractive option, in order to maintain the currently satisfactory performance of these activities.

This paper presents the developed research of an intelligent planning system for automatic generation of satellite flight operation plans which was called PlanIPOV. It is intended that this research be one of the first steps in direction to the automation of the satellite operation activities at INPE. This developed intelligent planning system performs the automatic generation of flight operation plans (FOP) for the routine operational phase of the satellites operations. The main reason for applying the proposed intelligent planning system in the routine phase is that the operation activities, in this phase, are composed of very repetitive and well defined tasks. As a consequence, these tasks are more feasible to be predicted, and automatically programmed. Another positive factor presented by the automation is that the routine activities are more susceptible to occurrence of human operation errors, and the automation can greatly reduce, or eliminate occurrences of this kind of errors.

The PlanIPOV employs the technology of planning, which is an Artificial Intelligence (AI) branch, in the automatic FOP generation. Its knowledge base of the satellite operation domain is implemented in the Planning Domain Definition Language (PDDL2.2).

Section 2 presents the PlanIPOV architecture. The prototype implementation, based on the PlanIPOV architecture, is described in Section 3. The prototype test was performed, by using real data of the satellites tracked by INPE. The executed tests and corresponding obtained results are presented in Section 4. Section 5 shows the limitations that have been observed in the utilization of the planning technique, and highlights the solutions adopted to overcome these limitations. The conclusion is presented in Section 6.

II. Architecture of the Intelligent Planning System for Flight Operation Plans

The Intelligent Planning System for Flight Operation Plans (PlanIPOV)¹ is a very promising result of a research work, currently being carried out at INPE, on the application of the technology of planning² in the automatic generation of FOPs for the routine phase of the satellite missions.

The functional requirements which guided the generation of the solution were the following ones: (i) The FOP generation process shall take into account all the time and resources restrictions imposed for the planning problem; (ii) It shall be possible that the user can choose a complete one week sequence of satellite passes over a specific ground station, or any subset of it, for the FOP generation process; (iii) It shall be possible that the operator can display the generated FOPs for any desired pass; (iv) The generated FOPs files shall be editable, in order to make possible that them can be directly modified by the user; and; (v) It shall be possible that the operation configuration of PlanIPOV can be modified during FOP generation processes, and that the performed changes can be applied to a specific satellite pass of a ground station, or to any desired set of the satellite passes involved in the current simulation interval.

Besides the functional requirements just presented above, it was also considered the following design constraints: (i) The architecture of a planning agent shall be inserted in the global architecture of the system³; (ii) The PDDL 2.2 language⁴ shall be used to build the knowledge base of the satellite tracking domain, since FOP generation is a problem of temporal planning with resource and time restrictions; and; (iii) The planner to be used for FOP generation shall be able to work with the PDDL 2.2 language and planning problems which include unconditional, deterministic and exogenous events.

The PlanIPOV system architecture is illustrated in Figure 1. It is composed of five modules: Configuration, Problem Generator, Planner, FOP Visualization, and Graphic User Interface.

The satellite passes prediction files (called here of PVP files) provides the input data for the system. Each PVP file contains the pass prediction of a satellite over a specific ground station, during a time interval specified by the user. For each pass described in the PVP files the system generates one output file containing the related FOP.

The internal files used by the developed FOP generation system are the following ones: the Domain file, the Problem files and the Configuration database. The Domain file contains the knowledge base of the satellite operation domain, coded in the PDDL 2.2 language. It is created manually based on information provided by satellite control experts. The Problem file is automatically generated by the system for each pass to be tracked. This file contains the pass environment initial state and the goal to be reached by the FOP. The Configuration database contains all the satellite configuration parameters, as well as the list of the ground stations involved in the satellite tracking activities.

The Configuration module is responsible for the task of Configuration database edition. It allows the user to edit: (i) The configuration parameters related to the ground stations to be considered in the process; (ii) The satellite

configuration parameters, including the telecommand ones; and; (iii) The relationships between ground stations and satellites, containing specific operation scenarios for each satellite - ground station set.

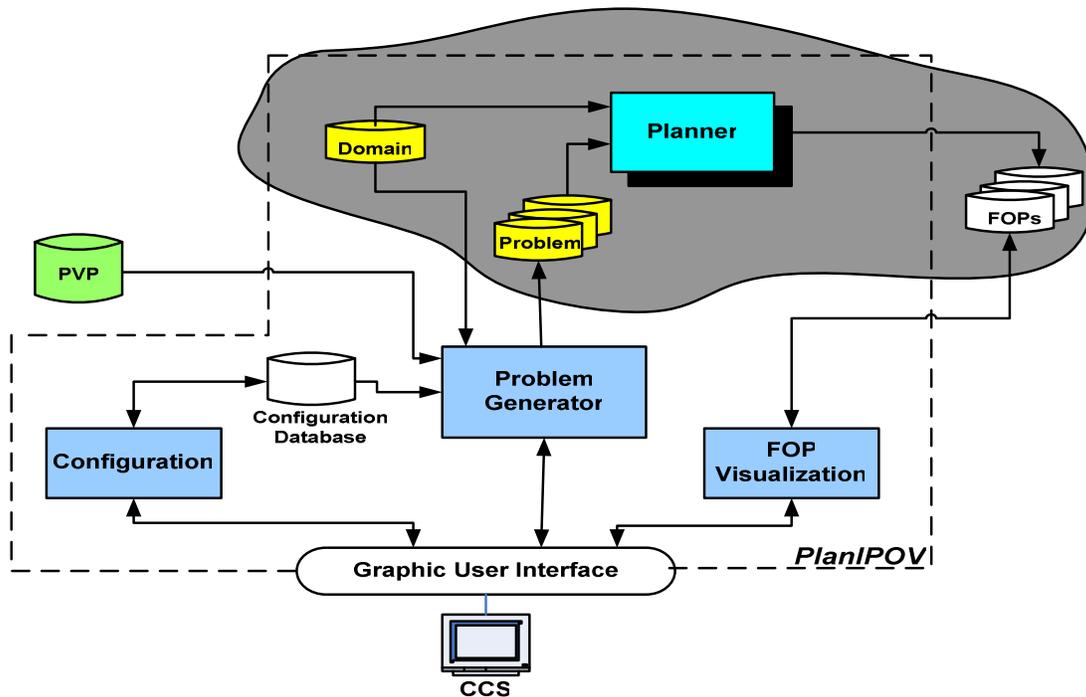


Figure 1 – PlanIPOV Architecture

The Problem Generator module is responsible for generating the Problems files, coded in the PDDL 2.2 language, for each pass. This module needs the following input data: the PVP file, the Configuration database and the Domain file. From this input data, this module generates the environment state at the beginning of each satellite pass contained in the involved PVP; the events that must occur in pre-defined times in the FOP (unconditional, deterministic and exogenous events); and the goals to be achieved at the end of the pass.

The Planner module is responsible for the FOP generation. This module is able to find a sequence of actions, using the knowledge base (Domain file), and considering the problem instance (initial environment states, exogenous events and goal) defined in the Problem file. The output files generated by the Planner are the satellite FOP files for each one of the passes specified in the Problem files. As can be observed in the darker part of Figure 1, the Planner module and the files of Domain, Problems and FOP represent the architecture of a planning agent.

The FOP Visualization module has the function of getting the FOP files generated by the Planner module and making its content available for visualization by the operator. It also allows that the operator can modify the contents of FOP file, before its execution.

Using the Graphical User Interface module the user can edit the Configuration database, and generate / visualize the FOPs.

III. Prototype Implementation

A prototype was developed for testing the proposed architecture for the PlanIPOV system. Table 1 presents the correspondence between the implemented or used prototype artifacts, and the modules, which have been considered in the PlanIPOV architecture. The tools and software languages used in the development of each artifact are presented as well.

Table 1 – Prototype Artifacts x PlanIPOV Modules

Artifact	PlanIPOV Module	Tool/Language
POVGenerator.exe	Graphical Interface Problem Generator POV Visualization	Microsoft Visual Studio 6.0 C++ Language
LPG-TD-1.0.exe	Planner	C Language
POVConf.mdb	Configuration	Microsoft Access 2003

Figure 2 presents the component diagram of the developed prototype. It shows the needed prototype configuring and running environment.

The following inputs files are used by the POVGenerator component: (i) nnn.pvp, which is, as already mentioned above, a file that contains the satellite pass prediction plan (nnn, is the name of PVP file generated in an existing external process the INPE’s Satellite Control Center); and; (ii) POVMEDZSRRTCDomain.pddl, which is a file that contains the satellite operation domain, codified in PDDL 2.2.

The POVGenerator activates the LPG-TD-1.0 program^{5, 6}, which is the planner responsible for FOP files generation.

The POVGenerator uses an ODBC driver to access the configuration database (POVConf.mdb), in order to get the configuration parameters. This component also provides to the user the option of activating the Microsoft Access 2003 software, which works as a tool to help him in the task of editing the POVConf.mdb database.

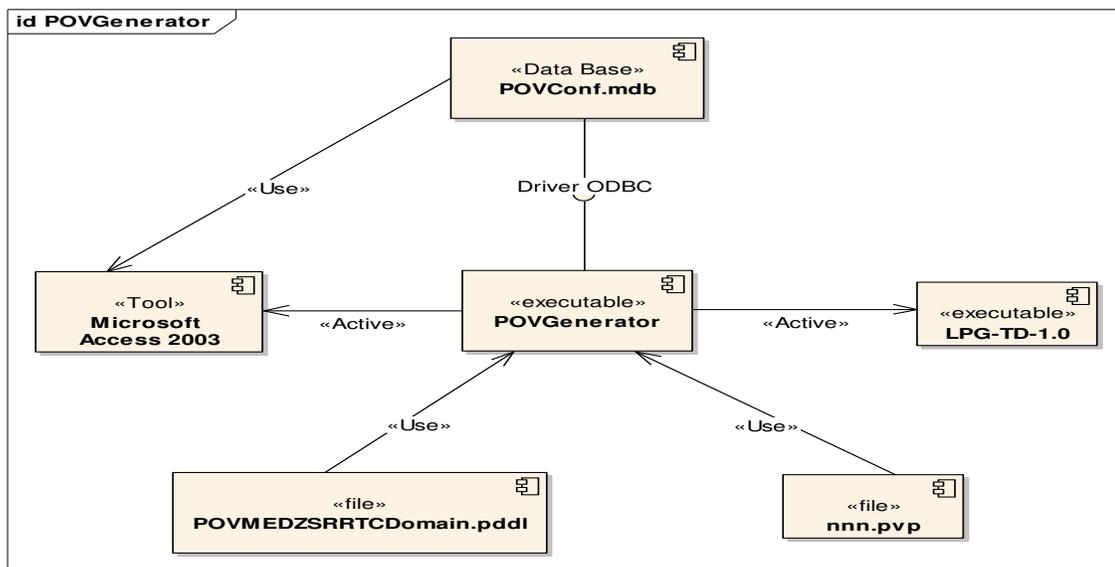


Figure 2 – Component Diagram of the Developed Prototype

The graphical user interface that was developed for the system is illustrated in Figure 3. Through options presented in the main screen of the prototype, the user can execute one of three kinds of functions: configuration of the POVConf database, generation of FOP’s and visualization of FOP’s.

If the control operator chooses the configuration function he will be able to include, update or delete parameters of the configuration database.

If the service of FOP generation is chosen, the user will have to choose the PVP file that will be used as input for the FOP generation process. Once the PVP file was chosen, all the existing passes in this file will be extracted and displayed. The user will have the option of selecting both one, all, or a set of the displayed passes as input. This option provides flexibility, allowing the user to modify the operational scenario for one or more passes by changing the Configuration database. Then the program provides automatic generation of problems files by using the PVP file and the passes, which were selected by the user. After that, the planner is activated in order that the FOP files can be generated, from the created Problems and Domain files. If the planner does not find a possible solution for the planning problem, an explanation message is displayed.

The option to visualize the FOP files will be available after their generation. The user will have the option to choose between the generated FOP files, the one he wish to visualize. The chosen FOP file content will be displayed, and can be printed, if needed. Additionally, the user has the option of insert, by direct edition, any alteration the find to be necessary.

Figure 3 presents the main screen of the prototype with the content of a generated FOP file for the 60484.7 pass of the SCD1 satellite over the Cuiabá ground station. Each line of the generated plan is composed of following fields: (1) nominal time when an action must be executed; (2) the action to be executed; and (3) duration of action execution.

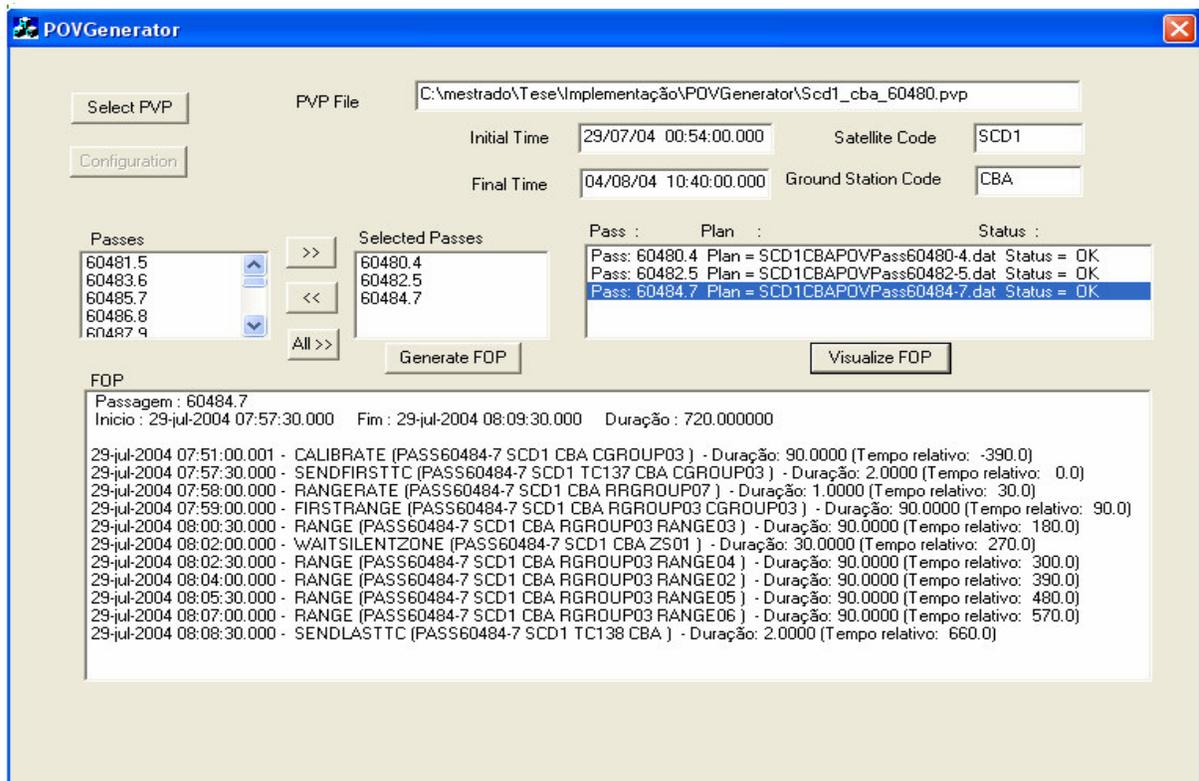


Figure 3 – Main Screen of Prototype

IV. Test and Results

The developed prototype was tested considering the generation of the FOP files for the satellites which are currently controlled by INPE: SCD1, SCD2 and CBERS2.

A set of PVP (pass prediction files) files, covering a period of seven days, which was generated by the real satellite operation process had been supplied, in order to be used in the developed system tests.

The tests had been executed for each one of the satellites tracked by INPE, using the Configuration database, PVP files and the Satellite Operation Domain file.

A program, which automatically compares the content of the FOP files generated by the prototype with the ones generated by the real satellite control center process, has been developed. Table 2 presents the results.

Table 2 – Results of comparison between the prototype FOP files and the ones used in the real satellite operation

Satellite Code	PVP File			Results		
	Pass Code	Period	Number of passes	Equal	Different	Without Solution
SCD1	60480	29/07/04 a 04/08/04	57	36 (64%)	14 (24%)	07 (12%)
SCD2	30444	29/07/04 a 04/08/04	57	46 (81%)	11 (19%)	00
CBERS2	9668	24/08/05 a 31/08/05	28	22 (78%)	06 (22%)	00

Three types of results had been found during the comparison of the set of FOPs generated from PVP files for each satellite:

- 1) Prototype and real FOPs with **equal** contents - All comparison criteria had been satisfied;
- 2) Prototype and real FOPs with **different** contents - The differences in the FOP timeline were, basically related to the relative position in time of the executions of ranging and range-rate, and also, to the amount of ranging executions. These differences, however, are not relevant since all the important actions involving putting the satellite in the needed configuration during the pass were accurately preserved;
- 3) Generation of problem files with **no solution** - The planner could not find a plan for the problem files. The reason of this is that the goals inserted in the problem files exceeded the time interval comprised by the pass.

V. Limitation Found in the Use of PDDL 2.2 and Solutions Adopted

During the process of development and validation of the knowledge database of the satellite operation domain, and of the prototype of PlanIPOV system, some limitations of PDDL 2.2 language had been noticed and solutions had been adopted to overcome them.

The first limitation was the great effort that is necessary for generating the problems files by hand. In the case of the satellite tracking domain, due to the environment changes that occur at each pass, it is necessary to reprogram the initial events, the exogenous events, and the goals to be attained in the problem file based on the PVP file. The manual generation of the problems files has a very high cost in terms of time and effort consumption.

This limitation was solved with the implementation of the Configuration and Problem Generator modules of the PlanIPOV system. These modules provided the automatic generation of the problem file for each pass to be tracked.

During the prototype development it was verified that the structure of the PDDL 2.2 language did not allow the classification of the predicates in the domain file. Without this classification, the automatic generation of the problem files would be a very much harder process because each predicate, instead of each type of predicate, has to be dealt individually. In such case, a generic solution could not be programmed.

The solution was to adopt the definition of predicates categories, and the name of each category was inserted in the domain file as a commentary line. Each predicate had then been inserted in the domain file, below of the commentary line that represented its category. The categories of predicates that were defined are the following ones:

- **Initial States of the Environment** predicates, which are those predicates that inform the state of the environment to begin the preparation of the plan.
- **Internal Events** predicates, which are used to define those states that occur during the planning, but they do not arise in the problem file.

- **Exogenous Events** predicates, which are used to represent unconditional, deterministic and exogenous events that can change to true or false, in defined times, independently of the actions that would be inserted in the timeline by the planner.
- **Goal states** predicates, which are those predicates that may be part of the goal to be reached with the plan.

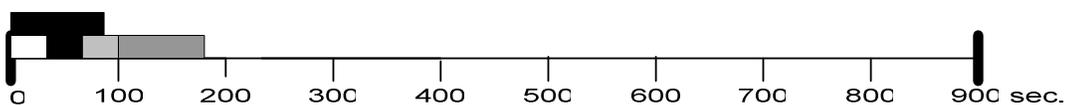
This classification allowed the development of a generic process to generate and assemble any predicate by means of its category.

Another weakness is that the language does not allow defining priorities in the exogenous events. This means that if in the problem file there are two exogenous events that happen at the same time, the planner can not find a solution.

To solve this limitation, a code was implemented to establish a priority for each exogenous event. When two or more exogenous events are coincident, then their start and end times are modified in accordance with these priorities.

Figure 4 illustrates an example of the use of the priority criteria for three exogenous events that are coincident in a satellite pass. The first graph presents a pass which have three coincident exogenous events (silent zone, first telecommand to be sent and range rate request). The second graph presents the final situation of exogenous events in the pass after the use of the priority criteria. The silent zone keeps its position because it has the biggest priority. The other two events are shifted respecting their priorities. The telecommand event is placed before the ranging/range rate event because its priority is higher.

1st) Exogenous Events that coincide in a pass:



2nd) Event Exogenous situation after the use of priority criteria:

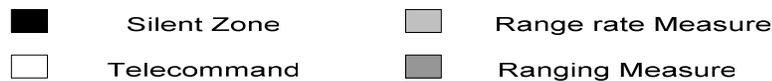
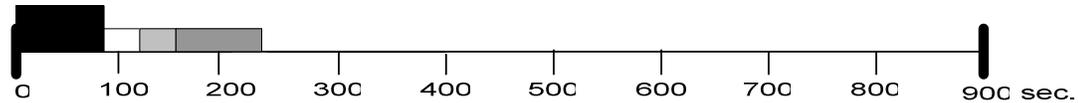


Figure 4 – Priorities between Exogenous Events

Other limitation of the PDDL 2.2 language is also related to priority. It is not allowed the establishment of priorities in the sub-objectives to be reached in the plan. For example, when a problem file has five sub-goals to be reached in the plan, but they exceed the period of time available in a pass, the planner does not find a solution. If it would be possible to define priorities for sub-goals a plan could, then, be generated by considering only part of the complete final goal instead of not finding any solution. The low priorities sub-goals could be discarded without affecting the satellite operation.

This limitation was partially overcome by implementing a special code to discard the sub-goals that exceeded the period of time comprised by the pass. But the prototype code did not take in account all the possible sub-goals conditions. This is the reason of the existing cases for which no solutions were found during the prototype tests (see Table 2).

Figure 5 illustrates the use of the priority criteria for discarding sub-goals. The first graph exemplifies the case where there are two coincident exogenous events (first ranging measurements request and silent zone), and where the remainder of the pass period could support the execution of five more ranging measurements, considering that each execution lasts ninety seconds.

As illustrated in the second graph, after the use of the priority criteria between the exogenous events, the silent zone event kept its position, and the first ranging measure event was shifted. Consequently, the period of time available in the pass for execution of other ranging measurements was reduced. If it was possible to define priorities

for the sub-goals, the problem file would not need to be modified in relation to the amount of ranging measurements executions. The sub-goals would continue to be the insertion of five ranging measurements requests, and the planner would insert only four requests in the plan by applying the sub-goals priority criteria.

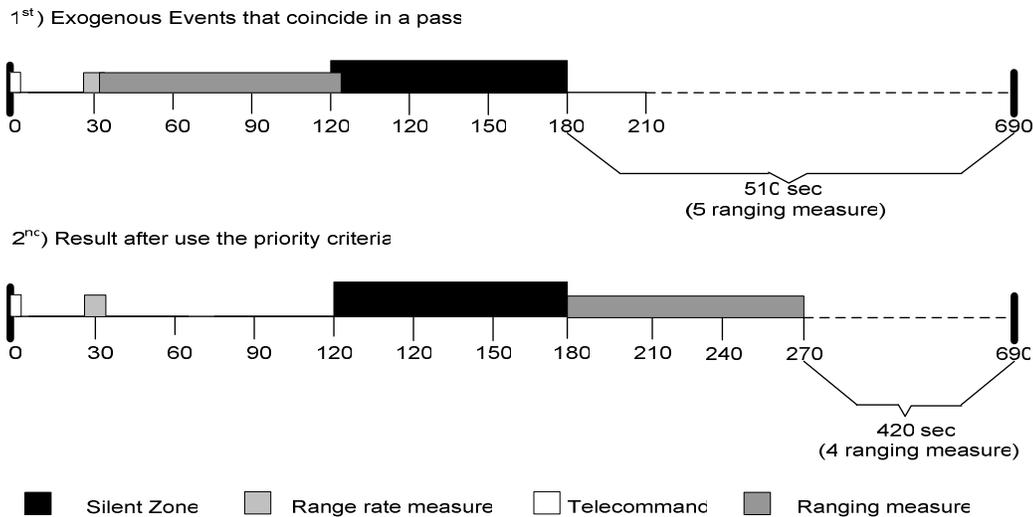


Figure 5 – Priorities of Sub-goals

A great number of code lines could be avoided in order to automate the generation of problems files process, if the language would allow the definition of priorities for the exogenous events and sub-goals.

VI. Conclusion

The development of the PlanIPOV system able to generate FOP, to support the on-ground satellite control activities, during the routine phase of satellite life time, is considered a first step towards the automation of the satellite operations activities at INPE.

The PlanIPOV system was developed with basis on the technology of planning agents. The planning standard language PDDL 2.2, for modeling the knowledge base of satellite operation domain, was applied in the implementation of a prototype.

The PlanIPOV prototype incorporates the architecture of a planning agent and automatically generates the needed problem files, in PDDL 2.2 language, for each visible satellite pass, during the time interval which is to be covered by the corresponding FOP. Some limitations found in the application of the PDDL 2.2 language for automatic FOP generation had been overcome by using C++ language to program certain specific functions. This solution made possible to use the PlanIPOV for generating FOPs to the satellites currently being controlled by the INPE's Tracking and Control Center.

The obtained results show that it is feasible to cope with a change in the satellite operation scenario, without having the need of modifying the domain knowledge base file, or/and the software code. In case of supporting a new satellite with the same operational requirements of the previous ones, it will be enough to insert the related data in the configuration database of the system. In an analogous way, if a new tracking ground station is to be added to the system, it will be enough to insert the data concerning this new station in the same database. In both cases, it is necessary to store in the configuration database the links between the satellites and the involved ground stations. Therefore, in order to include a new satellite or ground station in the FOP generation process it is enough to update of the configuration database avoiding, in this way, the need of made any alteration in the software code.

Along the development of this work, and especially after modeling the knowledge base of the satellite operation domain, and building the prototype, it could be confirmed that the use of the technology of planning agents for the automatization of satellite control operations is a promising way for automatic FOP generation. This is considered as a first step in direction to get a higher automation level for the execution of satellite control activities. This feature could significantly reduce the costs imposed by these activities. Despite the existence of some emphasized limitations, the research about planning technology is, nowadays, in great expansion and new solutions are being found, making easier the use of this technology in a great variety of real world applications.

References

- ¹ Cardoso, L. Ferreira, M. Orlando, V. Bianco, A. Aplicação da Tecnologia de Agentes de Planejamento em Operações de Satélites. VII Simpósio Brasileiro de Automação Inteligente, São Luís, Maranhão, setembro de 2005.
- ² Weld, D. S. Recent Advances in AI Planning, *AI Magazine*, 20:2, October - 1999
- ³ Feber, J. *Multi-Agent Systems an Introduction to Distributed Artificial Intelligence*, Addison-Wesley, London, 1999, pp. 152
- ⁴ Edelkamp, S. and Hoffmann, J. PDDL 2.2: The Language for the Classical Part of the 4th International Planning Competition, Albert Ludwigs Universität Institut für Informatik, Freiburg, Germany, Technical Report No. 195, January, 2004
- ⁵ Gerevini, A., Saetti, A., Serina, I. and Toninelli, P. LPG-TD: a Fully Automated Planner for PDDL2.2 Domains, (short paper), in International Planning Competition, 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04), *booklet of the system demo section*, Whistler, Canada, 2004
- ⁶ Gerevini, A., Saetti, A. and Serina, I. Integrating planning and temporal reasoning for domains with durations and time windows, in Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05), IJCAI-Inc., Edinburgh, Scotland, UK, 2005