A Meta-Model Object Oriented as Ontology for Description Domains

Maurício Gonçalves Vieira Ferreira
National Institute for Space Research, Brazil

SpaceOps 2010 Conference
 <i>Delivering on the Dream</i>
 <i>Hosted by NASA Mars
25 - 30 April 2010, Huntsville, Alabama

AIAA 2010-2172

An Object-Oriented Meta-Model as Ontology for Describing Domains and Problems for Planning Space Applications Planning

Rodrigo Rocha Silva[1]
Laboratório Associado de Computação e Matemática Aplicada – Instituto Nacional de Pesquisas Espaciais (INPE) – São José dos Campos, SP, 12227-010, Brazil

Mauricio Gonçalves Vieira Ferreira[2]
Centro de Controle de Satélites – Instituto Nacional de Pesquisas Espaciais (INPE) – São José dos Campos – SP – Brazil

Nandamudi Lankalapalli Vijaykumar[3]
Laboratório Associado de Computação e Matemática Aplicada – Instituto Nacional de Pesquisas Espaciais (INPE) – São José dos Campos, SP, 12227-010, Brazil

One of the major difficulties in developing planners, by using Planning and Scheduling techniques from Artificial Intelligence (AIPS), is the correct description of domains and problems. AIPS techniques have been in use to plan satellite activities. The work proposed in this paper defines an Object-Oriented Meta-Model as an Ontology to describe domains and problems related to Planning activities. The paper uses ontologies and knowledge management to provide support to model domain in the area of satellite control. A tool has been developed to describe such domains and problems with respect to space applications. The tool provides graphical resources to define concepts and relations besides enabling constructing frameworks of objects from ontologies as well as their definitions, properties and constraints.

I. Introduction

With the widespread use of AIPS the knowledge engineering now has a great influence on the conceptual problems in space applications. Based on this fact, for better understanding and classification of the domains and planning problems, the specification, modeling and domain analysis become essential.

Three major obstacles to planning and scaling within artificial intelligence (AIPS) in real problems are: (i) requirements specification, (ii) modeling (especially in modeling actions) and (iii) domains analysis, since these processes are considered a great "bottleneck" while developing real applications. [5]

Some techniques such as: lexicon, thesaurus, dictionaries, taxonomies and ontologies have the purpose of facilitating the representation and communication of the knowledge among specialists in a field. Lexicon represents the terms (or descriptors) that compose a language, where a specialized team of researchers in the related area develops a specific dictionary that lists the meanings of the terms, organizing them in an alphabetical order. A thesaurus is composed by a list of terms within a domain of knowledge and the relation among them, organized according to the meaning of the terms. Unlike other dictionaries, thesaurus does not include detailed definitions, since its objective is to help the user on selecting exact terms on presenting minimal differences between them.

1 Master´s Student, LAC(Laboratory of Computing and Applied Mathematics), Av. dos Astronautas, 1758, prédio LAC, São José dos Campos, 12227-010, Brazil, rrochas@gmail.com
2 Doctor Engineer/Researcher, CRC (Satellite Tracking and Control Center), Av. dos Astronautas, 1758, prédio CCS, São José dos Campos, 12227-010, Brazil, mauricio@ccs.inpe.br
3 Doctor / Technologist, LAC (Laboratory of Computing and Applied Mathematics), Av. dos Astronautas, 1758, prédio LAC, São José dos Campos, 12227-010, Brazil, vijay@lac.inpe.br

Taxonomies are based on a structural list of subject categories and the hierarchical relations among the topics, without including their own definition. Ontology is an explicit specification of one "conceptualization" [1], consists of a specification of objects, concepts and other entities that are assumed to be existent, beyond relations between concepts and restrictions expresses through axioms [2]. The use of Ontology to represent the concepts of the domain allows these to be reused in various applications.

Ontologies have strong relations among the terms, thereby differing from the thesaurus and also being in a language that can be interpreted by machines. So, ontology is formed by at least: concepts, relations, attributes, instances and axioms.

According to Alemida and Bax [3], an ontology can be defined in the same way as rules for balancing the fusion between the concepts (organized in one taxonomy) and the relations (i.e, the type of interaction among the concepts) in one domain of the knowledge, allowing the users to accomplish consultations to instances (specific elements, i.e, the data itself) using defined concepts by the ontology. Some representations like the ones above are frequently used with several objectives, such as, the information exchange and knowledge among professionals in a clear way, without facing ambiguities.

Due to the mentioned facts and due to the need to tools that can provide support during the lifecycle within a project of planning of domains, this paper investigates the problems involved in the interaction of information for knowledge management to plan and schedule activities, more specifically in the areas of space applications. The contribution of this paper is to offer support to model domains and problems of planning by means of an object-oriented meta-model. It describes AIPS domains and problems and also offers a vocabulary basis of space applications to help mapping the necessary knowledge.

This proposal is necessary, as there is a lack of standardization to represent knowledge about planning activities related to satellite control, complicating the understanding of all actors being involved in the process.

II. Conceptualization about Meta-Model

In "The investigators", Boorstin quotes the following "Einstein used to see the small and the big, the atomic and the cosmic, like an enigma, so as to discover the whole, explained by the laws and reason, inspired by what he used to call yours to be cosmic religious feeling. The individual notices the futility of human desires and the wonderful and perfect order which reveal themselves both in nature as in the world of thought, (…)"[4].

Meta-model seeks this idea, on trying to discover the whole, and then it can also be seen like a representation of the types of the entities that can exist in one model, its relations and the restrictions of applications.

Thus, based on Guarino's definitions [6] about the ontology in the computer science, it is established that: *"an ontology is a logical theory that relates the intended meaning according to a formal vocabulary, i.e, its commitment with a particular conceptualization of the world"*, and, according to Fonseca's [7] definition, components in which *"ontology is a specification theory of relative vocabulary to one determined domain, defining entities, classes, functions and relations among the components";* an object-oriented meta-model is proposed, like domain model of AIPS, because this technique permits to achieve what was said by Gómez-Pérez [8], that for the construction of an ontology, five types of components have to be considered: concepts (terms or classes, and its value domains), relationships, functions (special relations where the nth element of the relation is unique to the n-1 foregoing elements), axioms (model sentences that are always true) and instantaneous.

The combined use of ontologies and domain models have already been addressed in [9], [10] and [11]. However, there hasn't been any proposal, so far, to model AIPS domain as an object-oriented ontology for space applications. Moreover such proposal must be generic enough to describe such domains and it must also serve as an alternative to languages used in representing planning problems. This proposal must also take into consideration features such as flexibility, lose coupling, high cohesion and reusable components.

Through the cited characteristics, this study also supplies the first level of divergence in ontology as described by Klein [12], that concerns the description of the language or meta-model. The included divergences in this level are the syntactic differences, differences related to the primitive meaning of the different languages and differences on the expressiveness of the language. The proposed model is based on inherent objects to the scope of the AIPS not being coupled to none of the primitive of the pre-established.

The meta-model was named of KPlanOO (*Knowledge of Planning Oriented Objects*), which consists of classes that abstract the base of the knowledge to describe the domains and problems of automatic planning, thus, a meta-model to the domain of the AIPS. According to the figure number one, we can observe the abstractions identified and supported by the model.

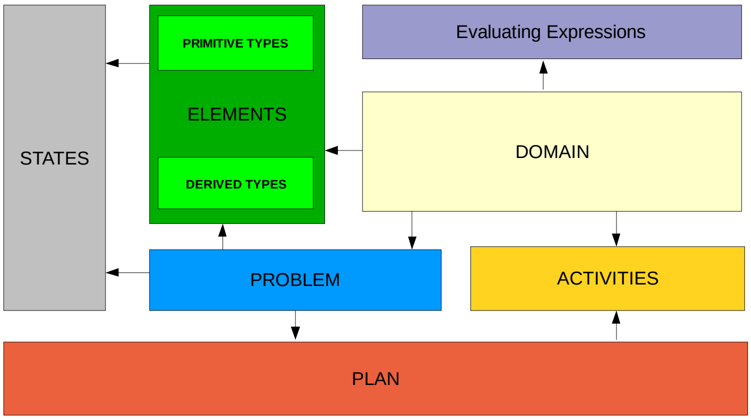

Figure 1: Abstractions identified to the Description of the Domains and Planning Problems supported by KPlanOO.

- **States:** All of the elements of the domain will be constructed by states indicating scenarios, situations and conditions of the modeled objects;

- **Elements:** The elements will define the objects, types one of a domain that will be the abstractions of the forms an object or action parameters and functions, for example, would assume in the domain;

- **Domain:** through its properties and characteristics, it will represent one domain of planning. Its definition will be given by the composition of the instances set from the entities that represent the states of the states concerning its objects and its activities.

- **Evaluating expressions:** expressions that will evaluate the conditions and established situations, that would receive and return values;

- **Activities:** Actions and functions that will be part of the domain, in which these will have properties, values and variables inherent to the planning context;

- **Problem:** problems to the resolution through the planning to a given domain; the initial states will be represented by the problem.

- **Plan:** result of the execution of the planner to a problem based on the domain related to this. A plan is composed by an activity collection (actions) that will be performed according to the given order.

Fouro[15] compares Modeling of the Domain and Ontology and considers the following: Models of the Domain and Ontologies are formed by structuring and organization the knowledge, aiming to create possibilities to its reuse. They are the result of the search by a practice of reuse of the systematical, formal and effective software.

Therefore, this study has a strong relation with the areas of the software reuse and artificial intelligence, since the domain models are used, mainly, by the community of the software reuse [14], and ontologies are applied, mainly, by the community of the artificial intelligence according to the perspective of the modeling of the knowledge.

The Ontology area now contributes to other areas of the knowledge, and the knowledge management is one of such areas [9]. Based on this fact, this study uses ontologies and domain models as management tools for knowledge to be fed into AIPS.

III. Using Ontologies and Domain Models as management tools for knowledge to the AIPS

The techniques of Domain Model and Ontologies are designed to allow the reuse of information about a domain storing information about this. They may also contribute towards reuse when developing a new application related to that domain. The main difference is that ontology does not assume the pre-existence of any system, concerning the domain to be modeled. Thus, the level of the abstraction of the ontologies is higher than those presented in the domain models.

Since its origin, Planning is based on knowledge, i.e, the planning process involves the manipulation of the complex knowledge, such as actions [16], in the process of planning, the knowledge (Objectives, Initial State and Actions) is the starting of the whole process.

Initially, KplanOO was developed based on motivation to contribute to space applications, more specifically to the operation activities of satellites. The development was based on types of ontologies presented in [17]. In spite of this, the developed ended up reaching a generic meta-model enabling modeling of entire knowledge to the planning process, not only to space applications but also to several domains and problems.

The KplanOO bases itself according to the following types of ontologies:

- **Meta-ontologies**, also known as Generic Ontologies or Fundamental Ontologies, that are reusable (or applicable) in different domains.

- **Ontologies of domains** are reusable in one specific domain providing vocabularies about the concepts within one domain and their relationships about the activities that involve this domain and about theories and elementary principles that govern that domain.

- **Applicable ontologies** have the needed knowledge to model specific situations concerning one task in one particular domain.

An ontology project has its approaches based on the object-oriented design [18] [19]. However, the development of the ontologies differs from class and relationships modeling in an object-oriented modeling (represented in UML). Noy & McGuiness [20] clarify that oriented object modeling focuses on some classification methods – an analyst takes some decision based on the operational properties of a class, but an ontology engineer takes decision based on structural properties of a class, making the explicit relationships and formalizing the concepts. As a result, a class and its structure relations in ontology can differ from a structure in an object-oriented modeling for a similar domain.

Following the above information, KPlanOO, despite its OO methodology, it was developed based on the generalization of the structural properties of the domain of the AIPS, thus reaching a structured model of the *componentized* classes that relate in order to establish the concepts and strong metrics to the modeling of the domain in question.

IV. The KPlanOO like Ontology to the Description of the Domains and Planning Problems

The current version of the meta-model is an evolution from the one presented in [13], and underwent some modifications, with 18 classes and 2 interfaces and their respective relations. The static structure of the KPlanOO is in the class diagrams presented by Figure 2.

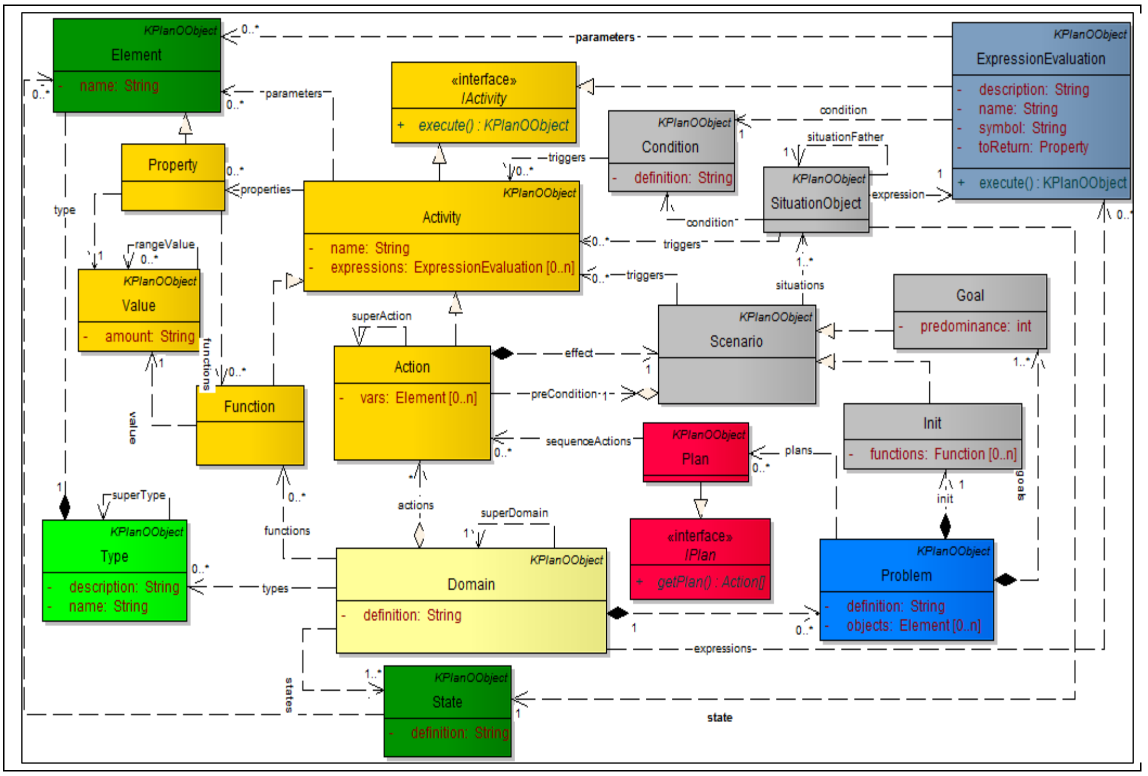

Figure 2: Static Structure of the KPlanOO

The description of each class is presented as follows:

- The class ***Type*** is the abstraction of the types that can be created in a domain, for example: Whole, Satellite, Region, etc…
- ***Element*** encapsulates the objects that will be part of the problem, and also encapsulates the parameters of the states and functions and needed variables on the description of the actions. Every element must have one associated type, and this type must be associated to the domain in which the element is being declared.
- ***State*** entity that represents defined states on the domain that will be applicable to the objects of the problems. An analogy can be inferred from this concept with the concept of the predicates from other Languages of the Domain Modeling like PDDL.
- ***Expression Evaluation*** Abstraction of the expressions of the evaluation, these will be applicable in elements that are transferred as parameters. This class has four attributes. One to represent the name of the expression, another one to the description, one to the symbol to this description and the last one assigns the represented results that will be generated by the expression. After the application of the evaluating expression, a condition will be generated by the expression. Examples of the expressions that would be created: > (bigger), < (smaller), + (sum), & (and) / (or), etc…
- ***Situation Object*** Entity that represents situations of the objects that will be manipulated during the planning. The situation is the set formed by the state of the objects by the circumstances tied to this state. It is possible to have one super situation and associated events to each situation, that would be executed based on the given situation.
- ***Condition*** Abstract of the conditions that one object in one determined state is, each condition can have associated events that are defined by a list called *"triggers"*.
- ***IActivity*** Interface to the activities of a domain.
- ***Activity*** Abstraction of activities of a domain that can be used in the planning process, having a list

of elements that represents its parameters and it can also have a list of properties.

- ***Value*** It represents the value that will be set as in the domain of the function as to the property. The class Value has a self-relationship that becomes possible the configuration of an interval of values.
- ***Property*** Entity that represents properties of an activity, these activities can have their values defined through the execution of a function.
- ***Function*** It encapsulates functions that can make part of the domain, has a list of inheritance that enables the configuration of the initial values or a range of values, and also has a list of the inherited elements that represents the parameters of these functions. Having this abstraction it is possible insert rules, using one set of the State Object.
- ***Action*** Abstraction of the domain actions, this includes the specification of the "Super Actions", through a self-relationship, beyond the list of the inherited elements that represents its parameters, defines another list to variables of each specific action. Each action has a domain scenario as effect and one scenario as the pre-condition that are defined through relation with the *State* entity.
- ***Domain*** It represents the description of the planning domain, composed by a list of actions; it has a list of the likely states to the objects of the problems with defined types. It is also possible to define a set of functions, and the domain can have several instances of the problem.

  The associations, multiplicities, attributes, restrictions and imposed classes by KPlanOO must be followed and obeyed during the definition of the domains and planning problems in order to build valid states in relation to the defined static structure.

  After shaping the field of planning domain it is possible to reach the problems modeling. As seen before, a planning problem refers to one planning domain model and it is usually characterized by a situation where two states are known: the initial state (Init) and the objective state (Goal).

The definition of the classes to define one problem:

- ***Init*** The representation of the initial state of a problem has a list of Functions that includes functions to be performed from the beginning of the planning process. Init is a specialization of the class Scenario.
- ***Goal*** Abstraction responsible for representing the meta to be achieved by a planner, through the states to the declared objects in the problem. It has an attribute *"predominance"* that stabilizes priority on the meta in relation to the other defined in the same problem.
- ***Problem*** Entity having the purpose of representing the description of the problems to a given domain.
- ***IPlan*** Interface to the generated plans.
- ***Plan*** Entity that represents the generated plans.

In order to enable the variety of the class to be instantiated and consequently the use of the design patterns – based on the mechanism of the inheritance like: *Bridge, Chain of Responsibility, Composite, Decorator, Observer or Strategy I* [21] – in the construction of the OO planners using KPlanOO as a tool to the planning domain models, all the conceptual classes of the model are direct or indirect sub-classes from one abstract class named like *KPlanOObject* this class has only one attribute "id". Figure 3 shows the inherited structure of the existing model.

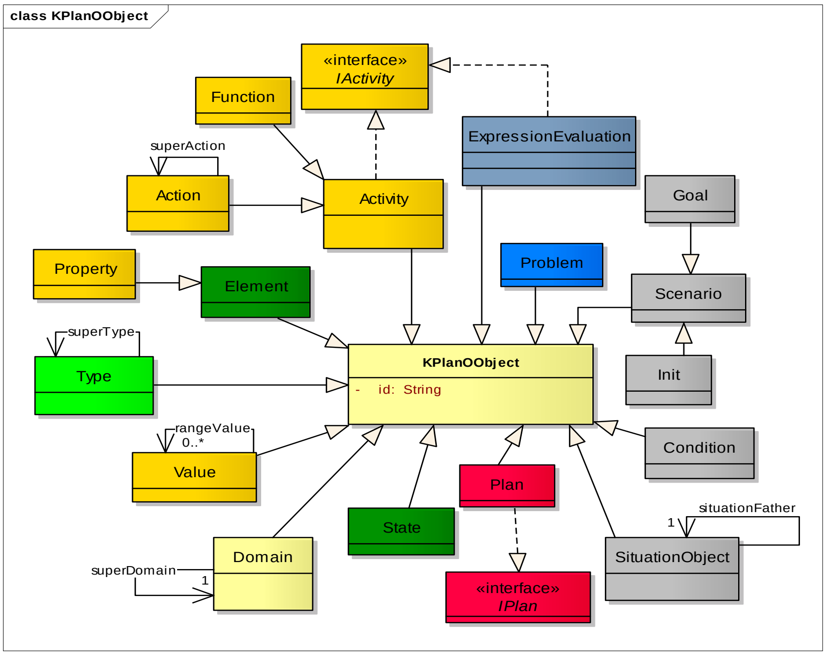

Figure 3: Inherited Structure presents on the KPlanOO

Characteristics such as the one presented by the interface *IActivity*, that follows the specification of the design pattern *Command*, make the KPlanOO highly extensible and enables its reuse. *Command* encapsulates a request as an object, so that customers may parameterize different requests, supporting reversible operations. Therefore, each implementation of the interface *IActivity* can be manipulated in several ways that are convenient to each planning process.

1. Validating the structure of the KPlanOO as Ontology through the tool Protégé

Ontologies Protégé [22] editor was used to model KPlanOO that it is possible to validate its structure with respect to the metrics in creating of an ontology. In Figure 4, one can observe KPlanOO as a tree structure enabled by "Jambalaya" plugin from Protégé.

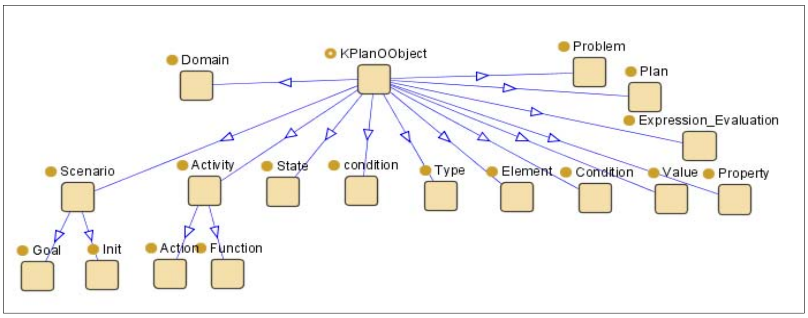

Figure 4: Estructure of the KPanOO modeled according to Protégé tool

KPlanOO was modeled using the Protégé-Frames fresource rom the platform Protégé. Thus, the development started with the user interface and knowledge server to support the construction and development of the domain of the AIPS, adapting the the data input and instantiation input. These data are the description of a satellite control domain of INPE described in PDDL by Cardoso [23]. A part of this description is presented in Figure 5, and Figure 6 presents the interface of the definition of the instance of the domain.

```
(define (domain satellite)
 (:requirements :strips :equality :typing :fluents :durative-actions)
 (:types satellite direction instrument mode)
 (:predicates
             (on_board ?i - instrument ?s - satellite)
             (supports ?i - instrument ?m - mode)
             (pointing ?s - satellite ?d - direction)
             (power_avail ?s - satellite)
             (power_on ?i - instrument)
             (have_image ?d - direction ?m - mode) )
 (:functions (slew_time ?a ?b - direction))

 (:durative-action turn_to
   :parameters (?s - satellite ?d_new - direction ?d_prev - direction)
   :duration (= ?duration (slew_time ?d_prev ?d_new))
   :condition (and (at start (pointing ?s ?d_prev)) )
   :effect (and  (at end (pointing ?s ?d_new))
                 (at start (not (pointing ?s ?d_prev)))))
```

Figure 5: Part related to a Description of a Domain to the Satellite Control of INPE in PDDL
SOURCE: adapted from [23] L. S. Cardoso **(2006), page. 65**

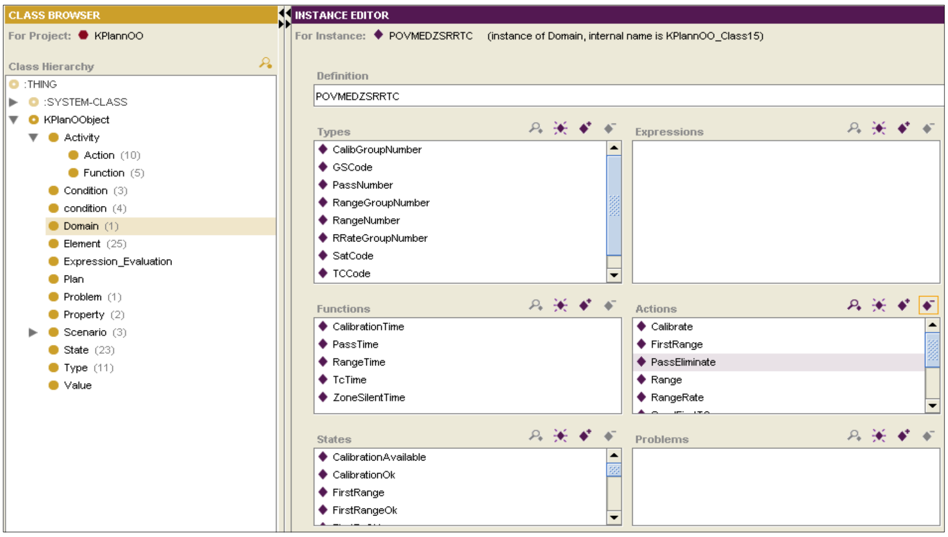

Figure 6: Definition of the Domain to the Satellite Control of the INPE based on the KplanOO using the Potégé tool

This interface was elaborated based on the modeling of KPlanOO based on Protégé tool, where it was possible to create the instances of the entities related to the meta-model to define the domain of the Satellites Control of INPE. In Figure 7, it is possible to observe part of the diagram of the objects representing the generated instances to the definition of this domain.

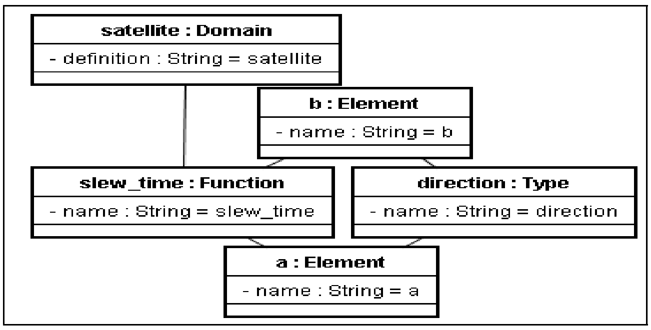

Figure 7: Part of the diagram of the objects representing the Description of one Satellite Domain

V. Conclusion

The approach described here, with the other studies, belongs to an initiative of the Satellite Control Center of INPE, to automate the operations of its satellites.

The proposed model presented in this study brings the planning concepts and the scaling area of IA to the systems of satellite operations, along with the concepts of Ontology and Object Orientation. Besides helping the generation of plans to the space operations, it will also contribute on the representation of the knowledge of the panning domain, facilitating, thus, the correct mapping of the needed data input to generate the definition of the domain.

This approach opens other possibilities concerning the development of the application of planners, also facilitating the reuse and providing higher coupling to the AIPS and the Object Orientation as techniques to be applied in space missions. It is expected to contribute in the generation of knowledge to the use of the AIPS, new possibilities of development of planners with flexibility, extensibility, standardization of projects and reuse.

References

[1] Gruber, T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies (IJHCS),43(5/6),907 928,1995.

[2] Zlot, F., Oliveira, K.M., Rocha., A.R. Modeling Task Knowledge to Support Software Development.SEKE’02, Itália, 2002.

[3] ALMEIDA, M. B.; BAX, M. P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Ciência da Informação*, Brasília, DF, v. 32, n. 3, p. 7-20, 2003.

[4] BOORSTIN,D.J. (2003) *Os investigadores – A história da permanente busca do homem para compreender o seu mundo*, Rio de Janeiro, ed: Civilização Brasileira.

[5] GÓMEZ-SANZ, J.J.. (2001) "*Meta-modelos para el desarrollo de sistemas multi-agente*," Caepia disponível no site [http://grasia.fdi.ucm.es/SP/publicaciones/index.html] acessado em 05/2004.

[6] Guarino, N. *Formal Ontology and Information Systems.* Proceedings of FOIS’98, Trento, Italy, 6-8 June (1998). Amsterdam, IOS Press, pp. 3—15. (http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/FOIS98.pdf)

[7] Fonseca, F., Egenhofer, M. et al. *Ontologias e Interoperabilidade Semântica entre SIGs*. In: II Workshop Brasileiro em Geoinformática - GeoInfo2000, Proceedings. São Paulo.

[8] Gómez-Pérez, A. e Benjamins, V.R. *Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods*. In: International Joint Conference on Artificial Intelligence (IJCAI-99), Workshop on Ontologies and Problem-Solving Methods (KRR5). Stockholm, Sweden,(1999).

[9] MANGAN, M. A. S. ; MURTA, L. G. P. ; SOUZA, J. M. ; WERNER, C. M. L. . Modelos de Domínio e Ontologias: uma comparação através de um estudo de caso prático em hidrologia. In: International Symposium on Knowledge Management/Document Management (ISKM/DM'2001), 2001, Curitiba. IV International Symposium on Knowledge Management/Document Management (ISKM/DM'2001), 2001. p. 149-172.

[10] Faria, C, G., “Uma técnica para a aquisição e construção de modelos de domínio e usuários baseados em ontologias para a engenharia de domínio multiagente”, Dissertação (Mestrado) – Centro Universitário do Maranhão, 2000.

[11] Guizzard G., "Uma Abordagem Metodológica de Desenvolvimento para e com Reuso, Baseada em Ontologias Formais de Domínio". Dissertação de Mestrado. Universidade Federal do Espírito Santo, 2000.

[12] KLEIN, M. Combining and Relating Ontologies: An Analysis of Problems and Solutions. In: *Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence*, 2001, Seattle, USA, p. 53-62.

[13] Silva, R. R.; Ferreira, M. G. V.; Vijaykumar, N. L. *Descrevendo domínios e problemas de planejamento através de um modelo orientado a objetos*. In: IV Fórum de Inteligência Artificial da Região Sul 2008, Canoas, RS, Brasil.

[14] ARANGO, G., 1994, "Software Reusability - Domain analysis methods", *Ellis HorWood*, pp. 17-49.

[15] FOURO, A. M. M., WERNER, C. M. L., 2001, "Modelos de domínios ou Ontologias?", *RTInfo – Revista Tecnologia da Informação*.

[16] T. S. Vaquero. itSIMPLE: *Ambiente integrado de Modelagem e Análise de Domínios de Planejamento Automático*. Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2007.

[17] SOUZA, J. F. ; SIQUEIRA, S. W. M. ; AZEVEDO, L. G. ; BAIAO, F. ; LOPES, M. ; SANTORO, F. M. ; CAPPELLI, C. ; NUNES, V. ; MAGDALENO, A. Gestão de Ontologias. Rio de Janeiro: Departamento de Informática Aplicada da UNIRIO, 2008 (Relatório Técnico).

[18] RUMBAUGH, J.R., BLAHA, M.R., LORENSEN, W., *et al.* Object-oriented modeling and design. Upper Saddle River: Prentice-Hall, 1991.

[19] BOOCH, G., RUMBAUGH, J., JACOBSON, I. The unified modeling language user guide. Addison-Wesley, 1999.

[20] Noy, N.F., MCGuinness, D.L. Ontology development 101: A guide to creating your first ontology, Relatório Técnico KSL-01-05, 2001, Stanford Knowledge Systems Laboratory, Stanford.

[21]Gamma E., Helm R., Johnson R., Vlissides J.; Design Patterns: Elements of Reusable Object-Oriented Software. 1.ed. Estados Unidos da América: Addison-Wesley, 1995. ISBN 0-201-63361-2

[22] Protégé [Online]. 2008 Aug; Available from: URL: http://protege.stanford.edu.

[23] L. S. Cardoso. Aplicação da Tecnologia de Agentes de Planejamento em Operações de Satélites. Dissertação de Mestrado em Ciência da Computação. INPE, 2006.