S E S 2 0 2 0
Sixteenth International Scientific Conference
SPACE, ECOLOGY, SAFETY
4 – 6 November 2020, Sofia, Bulgaria

DEVELOPMENT OF TOOLS FOR MODELS' DESIGN OF SYSTEMS OF MULTI-SATELLITE SYSTEMS

Atanas Atanassov[1], Liliya Atanasova[2]

[1]Space Research and Technology Institute – Bulgarian Academy of Sciences
[2]Medical University – Sofia
e-mail: At_M_Atanassov@yahoo.com

Keywords: *space mission analysis and design, multi-satellite systems, system of systems, models, computer simulation, program object creation*

Abstract: *An approach to create a multiphysical system for simulation of space missions based on multi-satellite systems is considered. Some basic models devoted on structurally and functionally complex space systems simulation are considered. At this stage, models of multi-satellite systems, space debris, models for calculating parameters of environmental elements, situational tasks can be created.*
Further, models of solar arrays and rechargeable batteries, memory for data storage have also been developed.

РАЗРАБОТКА НА СРЕДСТВА ЗА СЪЗДАВАНЕ НА МОДЕЛИ НА СИСТЕМИ ОТ МНОГО-СПЪТНИКОВИ СИСТЕМИ

Атанас Атанасов[1], Лилия Атанасова[2]

[1]Институт за космически изследвания и технологии – Българска академия на науките
[2]Медицински университет – София
e-mail: At_M_Atanassov@yahoo.com

Резюме: *Обсъжда се подход за създаване на мултифизична система за симулация на космически мисии основани на много-спътникови системи. Разгледани са някои основни модели въз основа на които могат да се симулират структурно и функционално сложни космически системи. На този етап могат да се създават модели на многоспътникови системи, космически отломки, модели за изчисляване на параметри на средата, ситуационни задачи.*
Разработват се също и средства за създаване на модели на слънчеви и акумулаторни батерии, както и на памет за запазване на данни.

Introduction

Advances in space technology have led to the development of concepts and their implementation related to the launch of satellite systems into outer space [1]. The concepts of satellites and space missions are evolving. Expensive satellites, equipped with a lot of scientific equipment, give way to small (mini, nano-, pico-, femto-, chip), cheap, specialized platforms [2, 3]. Concepts based on separate modules and systems not located on one platform, moving freely and cooperating to solve common problems – fractionated or partitioned spacecraft [4], are in the process of research. Satellite systems can connect with both ground stations and communication satellite systems for information exchange and control. Thus, there are systems with different structural elements (homogeneous or heterogeneous), organized in different hierarchical template.

Life cycle simulation of multisatellite systems at the stage of their analysis and design is possible by computer mutiphysics applications. Creation of some classes of models for space mission simulation is exposed in this article.

Previous work

The development of a software system for space mission simulations was initiated [5]. Initially, it included tools for defining models of motion of the multi-satellite systems and an integrator of equations of motion [6]. In addition, tools have also been developed to describe the definition of situational tasks and to determine the time intervals in which various orbital events take place (based on specific parameters and constraints). The developed parallel processor for situational analysis includes computational models for different situational conditions [7]. Models and computing tools have also been developed to visualize 3D scenes involving the Earth and satellite systems [8]. Other tools, useful in developing multiphysical simulation applications, have also been developed [9, 10].

Statement of the problem

Each model represents a set of parameters, some of which play the role of initial conditions for the described system. A description of the behavior or functionality of the system is also required. An example of a satellite system model is the/a set of initial conditions for all satellites along with some form of equation of motion, including different perturbations. This is essentially a mechanical model. The individual satellite subsystems and instruments are represented by mechatronic models. These include geometric descriptions, related to the location and orientation in the satellite's coordinate system (for optical instruments), and functional behavior, which may include descriptions of mechanical movements, execution of specific commands, power consumption and data generation. The purpose of the models is to describe the behavior and changes in the state of the described object over time. The different models describe individual aspects of the object under consideration.

The different models, describing a system, are interconnected and the results of one can be the input parameters for another. Some models may use results from more than one model. For example, the model of satellite motion in a multi-satellite system provides the coordinates of objects at successive points in time. These results are used by model(s) to calculate environmental parameters. The coordinates of the objects and the parameters of the environment can be used in situational analysis, to determine the time intervals in which to perform relevant satellite operations. The results of the situational analysis are used in planning the satellite operations and simulation of the functional behavior of the satellite subsystems. Development of some models and linking their implementations in appropriate computational tools in a multiphysical application is the subject of this paper.

Development of some basic models

Satellite constellations. The multi-satellite system models are based on various parameters. Wertz [11] consider different types satellite constellations. So far, only Walker-Mozhaew multi-satellite systems are considered in our work. The satellites of the considered constellation type are distributed in several orbital planes. The number of the both, satellites and orbital planes are basic parameters of constellation model. The large semi-axes and inclinations of the orbits are also important for solving of the tasks. The eccentricities of the orbits must have values close to zero. A certain phase shift is required for perigee arguments of the satellites located in two adjacent orbital planes. The shapes of satellite constellations are determined by specific values of orbital parameters. At the end, each multi-satellite system model has private name.

Based on these parameters, for each of the satellites, the initial conditions (coordinates and velocities) for integrating orbital motion are calculated. The choice of the motion model also implies the indication of a model of the disturbing forces. To use the parallel integrator of satellite movements [6], the number of calculation threads for the specific calculation task is also indicated. Models of more than one satellite system, each marked with its own name, can be created within a project.

Space debris. The developed model of space debris has as its main parameter the number of particles N. The orbits are evenly distributed in an area of space limited by the minimum and maximum height above the earth's surface. An interval (0 – 180°) is specified for the inclination $\boldsymbol{i}$ of the orbits. A random number generator is used to determine the orbital parameters of each particle.

Within one simulation project, different space debris models can be created, each of them characterized by various additional parameters – masses and dimensions, areas of space, disturbing forces. Each model is characterized by a name.

The velocity vectors and coordinates for the initial moment (initial conditions for the integration of the equations of motion) are determined, based on the orbital parameters of the particles. When creating the model of space debris ensemble, a model of motion is chosen, indicating the forces disturbing the movement. The number of threads for the parallel integrator could also be chosen.

Observer. This model allows dynamic scenes to be rendered [8]. One or two objects of type **Observer** could be created. Parameters of this model determine the direction in which the Observer is looking and the size of window in which the dynamic scene is displayed. This program object uses different calculation instruments based on a single thread. In addition, each object of this type can be attributed to different data - to depict the Earth, different satellite constellations, space debris and others.

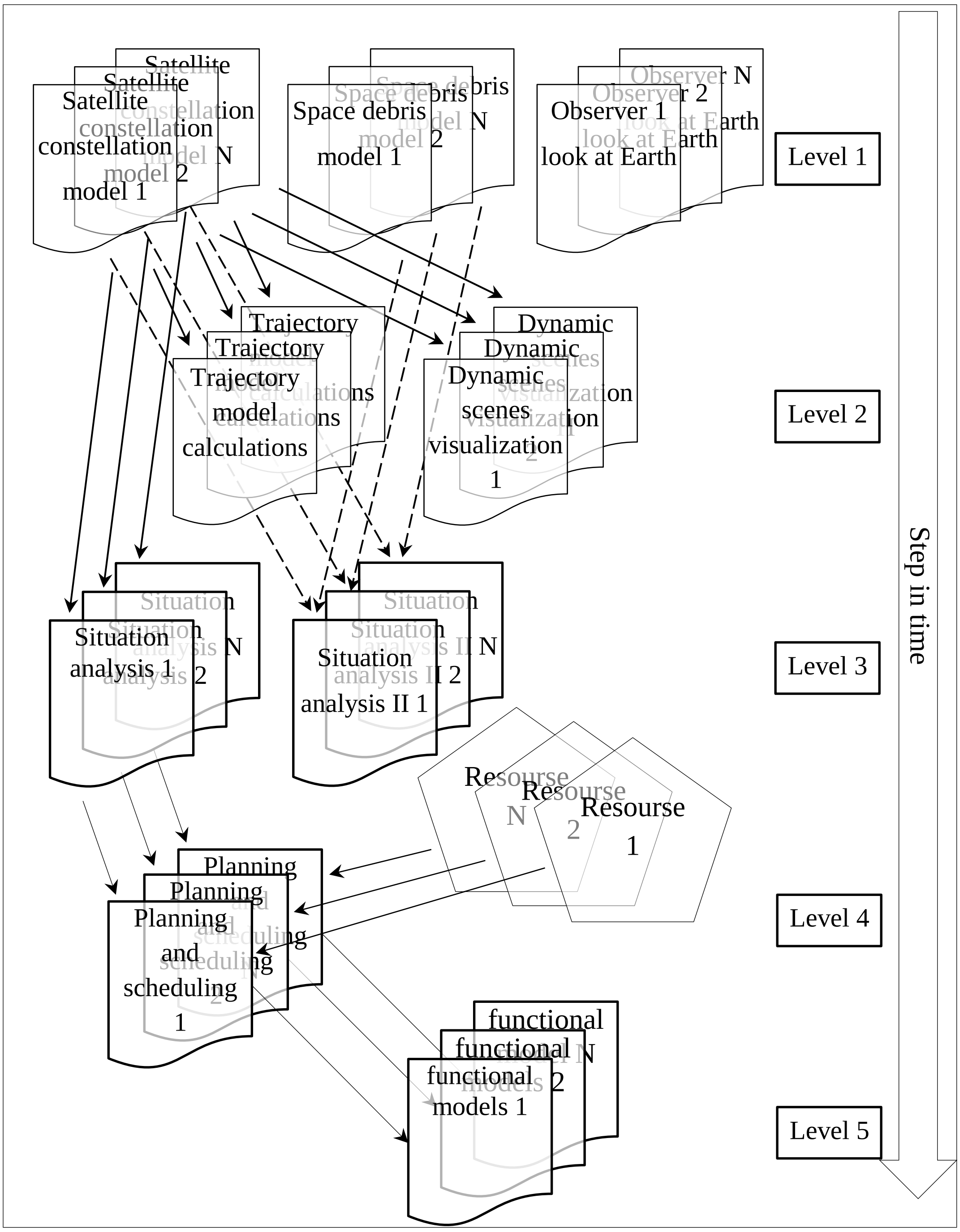

Fig. 1. Developed models, distributed by levels, with connections between them, are shown (some of the models shown are not considered)

Environmental models. Calculation of model values of various parameters of the environment, at points on the trajectory of the satellites, is used in solving of various tasks – situational analysis, simulation of an active experiments or measurements. As an example, we can mention the parameters of the neutral atmosphere, ionosphere, and magnetosphere.

Dialog tools allow specifying – to calculate the same or different environmental parameters for individual satellites of the constellation.

Situational tasks models. In the analysis of multi-satellite missions, the situational tasks for all satellites are the same. Therefore, the means of dialogue provide an opportunity to choose when defining the parameters of a situational task whether to apply to one satellite or to all of the constellation. At this stage, we are working on situational tasks related to a multi-satellite system. There may be a need to solve tasks related to satellites from different multi-satellite systems.

In the analysis of multi-satellite missions, the situational tasks for all satellites are the same. However, the dialogue tools provide an opportunity to choose, when defining the parameters of a situational task, whether to apply them to one satellite or to the entire constellation. At this stage, we are working on situational tasks related to a multi-satellite system. There may be a need to solve tasks related to satellites from different multi-satellite systems.

Solar array model. Solar array model includes some attributes as: number of panels of the solar array, area of the solar panels, efficiency and other. The model of functioning of satellite solar arrays includes a mode of orientation relative to satellite platform and orientation relative to the Sun (variable or not in the time outside Earth's shadow). A possibility to choose a law of ageing as result of the hostile space environment (charged particles, vacuum, electro-discharging) is insured. The dialogue tools provide an opportunity to choose, when defining the parameters of a solar array model, whether to apply them to one satellite or to the entire constellation.

Rechargeable battery model. The model of battery comprise the next set of parameters: number of modules, maximal capacity, lower and upper voltage limits, charging mode, maximal number of cycles and ageing model. Ageing of battery is result of the hostile space environment. The developed dialogue tools provide an opportunity to choose, when defining the parameters of a battery model, whether to apply them to one satellite or to the entire constellation.

Data storage model. The main parameter in the memory model is its volume S, set in Mb. Another important parameter, that can be used in analyzes of changes in the functionality of satellite systems, is the coefficient of storage ageing, result from degradation process ka [ΔS / Δt]. Due to the influence of cosmic factors, memory is damaged and the volume decreases linearly over time.

The operation of the satellite instruments generates data that is stored in memory. When it is possible for a specific satellite to communicate with a ground station or a communication satellite, the data is transmitted and the memory is freed. Memory is a passive element of the satellite system. It is a resource that determines the feasibility of the other satellite operations, related to the other instruments and systems.

Program object descriptors

Space missions are presented by different objects. When we analyze the processes, associated with a space mission, we consider the objects included in it and their motions and functioning. Computer simulation can be performed based on a model, based on program objects (object-oriented programming). Different type's objects are created on the base of respective above discussed classes.

```
AI_pool_par%AI%num_threads   = NumThreads         ! number of integrator threads
AI_pool_par%AI%    ha_race       = AI_handler_addr    ! handler of integrator pool                    (a)
AI_pool_par%AI%counter_adr    = AI_GlbCount_addr ! address of the global counter
AI_pool_par%AI%thread_par_adr= PoolParam_addr    ! address of the array containing pool's parameters
AI_pool_par%AI%granulation      = Granularity           ! task partitioning parameter

CALL  add_object(num_AIs,AIs_descriptor_adr,AIs_descriptor_adr,AI_pool_par)



IVP_par_SatCon%kind                  = 1                    ! type of the task to be solved           (b)
IVP_par_SatCon%name                = s_const_name ! satellite constellation name
IVP_par_SatCon%IVP%integ_index        = num_AIs; ! serial number of the integrator for this initial values problem
IVP_par_SatCon%IVP%num_objects      = Numsat     ! number of satellites in the constellation
IVP_par_SatCon%IVP%t_adr                  = LOC(t);    ! address of time variable t
IVP_par_SatCon%IVP%dt_adr                = LOC(dt);  ! addres of variable containing time step Δt
IVP_par_SatCon%IVP%xvn_adr            = xvn_adr;   ! velosities of satellites at the begining of step Δt
IVP_par_SatCon%IVP%xvk_adr            = xvk_adr;   ! velosities of satellites at the end of step Δt
IVP_par_SatCon%IVP%eps_adr             = eps_adr;   ! tolerans for numerical integration
IVP_par_SatCon%IVP%adr_Grv_model  = adr_perturbations;! model of perturbation forces
IVP_par_SatCon%IVP%len_Grv_model  = len_Grv_model   ! size of array containing perturbation force model [in byts]
IVP_par_SatCon%IVP%transfer_data_adr= transfer_adr !  workspace for the integrator
IVP_par_SatCon%IVP%work_data_adr    =     work_adr !  workspace for the integrator

CALL  add_object(num_IVPs,IVPs_descriptor_adr=IVPs_descriptor_adr, &
                          IVPs_descriptor_adr_new=IVPs_descriptor_adr, &
                                          level__1=IVP_par_SatCon) !
```

Fig. 2. Creation of the descriptors and saving/store them in special arrays. (a) Descriptor of integrator pools AI_pool_par. (b) Descriptor of initial value problem about motion of satellites in the constellation IVP_par_SatCon. The add_object subroutine adds new members to the appropriate levels.

```
type      level_0
 UNION
   MAP                        ! for pool of threads program model
     type (pool_par) AI
   END MAP
   MAP                        ! for single thread program model
      type(OneThread_type) OneTh
   END MAP
 END UNION
end type  level_0

type      level_1            ! level_1 combines IVP and 3D_Earth vizualization
  integer    kind            ! kind of task
  character    name*25
 UNION
  MAP                        ! IVP%t1
    type(IVP_par) IVP
  END MAP
  MAP                        ! 3D_Earth%t2       - for vizualization
    type(type_addr_3D_p) Earth
  END MAP
 END UNION
end type  level_1

type      level_2            ! level_1 combines environment parameters calculation or drawing satellites
  integer    kind            ! kind of task
  character    name*25
 UNION
   MAP                       ! Model parameters
     type(TrajParam) TrPar
   END MAP
   MAP                       ! Painting satellite & space debris
     type(ObsObject) ObsObj
   END MAP
 END UNION
end type  level_2

type      level_3
  integer    kind
  character    name*25
   type(SitProblems) SitProb
end type  level_3
```

Fig. 3. Description of different levels, containing descriptors of different program objects

Different calculations are performed in a certain sequence within one step of the system time in computer simulation, reflecting the processes that are performed with a separate object from a space mission.

For example, the coordinates of the satellites at the end of the interval Δt are determined by numerical integration. For various analyzes model calculations are performed, and parameters of the environment are calculated based on respective models. The appropriate time intervals to perform satellite operations are obtained through situational analysis. The operation of satellite systems and instruments can then be simulated. The solution of the listed computational problems is performed by specific methods. The data corresponding to each task and the method implemented programmatically represent an object in terms of object-oriented programming.

Figure 1 illustrates the placement of different models at different levels and the relationships between them. Descriptors represented by different types can be grouped at one level. This is a polymorphism and involves a different interpretation of the data according to the type of program object.

Figure 2a illustrates the assignment of parameter values to a program object descriptor for calculation instruments, based on thread models. Figure 2b illustrates assigning parameter values to a program object descriptor for solving initial values problems. All object descriptors are saved through subroutine **add_object** in special arrays according to approach developed in [10].

Figure 3 presents description of different levels – **level__0**, **level__1**, **level__2** and **level__3**. These types are compiled through inheritance of other types, described elsewhere.

The order of execution of calculations in a simulation model (within one step of the system time) depends on the created program objects, whose descriptors are located at different levels. The execution of the solvers for each of the program objects is done sequentially by levels, which reflects the connections between them.

Conclusion and future work

Dialogue tools, approaches for creation of connections between models, and a concept for synthesis of simulation models have been developed. The models can describe different types of objects –multi-satellite systems of different types, service systems and scientific tools; sets of space debris with different characteristics. The developed classes of models will be further developed by adding other variants, as well as by refining the individual parameters.

Work is underway to create tools for modeling some optical devices for remote sensing. These tools include dialog forms that determine the basic parameters of the instruments, as well as the orientation of the satellite platform.

References:

1. Selva, D., A. Golkar, O. Korobova, I. L. I. Cruz, P. Collopy, & O. L. de Weck, Distributed earth satellite systems: What is needed to move forward?. *Journal of Aerospace Information Systems*, *14*(8), 2017, pp. 412-438.
2. Barnhart, D. J., T. Vladimirova, & M. N., Sweeting , Very-small-satellite design for distributed space missions. *Journal of Spacecraft and Rockets*, *44*(6), 2007, pp. 1294–1306.
3. Barnhart, D. J., Vladimirova, T., Baker, A. M., & Sweeting, M. N. (2009). A low-cost femtosatellite to enable distributed space missions. *Acta Astronautica*, *64*(11-12), 1123–1143.
4. Brown, O., P. Eremenko, & P. Collopy, Value-centric design methodologies for fractionated spacecraft: Progress summary from Phase I of the DARPA System F6 Program. In *AIAA Space 2009 Conference & Exposition,* 2009, p. 6540.
   Available on: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1263&context=smallsat
5. Atanassov, A. M., Program System for Space Missions Simulation – First Stages of Projecting and Realization. Proceedings of SES 2012, 2013, pp. 209–214.
6. Atanassov, A. M., Parallel, adaptive, multi-object trajectory integrator for space simulation applications, Advances in Space Research, v. 54, 8, 2014, pp. 1581–1589.
7. Atanassov, A. M., Parallel satellite orbital situational problems solver for space missions design and control, Advances in Space Research, v. 58, 9, 2016, pp. 1819–1826.
8. Atanassov, A, Atanasova, L., Development of New Functionalities for a Space Missions Simulation Program System, Proceedings SES 2019, 2019, pp. 70–74.
9. Atanassov, A., Method of Thread Management in a Multi-Pool of Threads Environments, *Proceedings of SES,* 2014, 2015, pp. 241–246.
10. Atanassov, A. M., Development Classes of Objects' Descriptors for Space Missions Simulation. Proceedings SES 2015, 2016, pp. 51–55.
11. Wertz, J. R., Orbit & Constellation Design & Management: Spacecraft Orbit and Attitude Systems. Microcosm Press. 2001.