Journal of Physics: Conference Series

Virtual space virtual satellite

To cite this article: V A Baleskin *et al* 2021 *J. Phys.: Conf. Ser.* **2134** 012014

Virtual space virtual satellite

V A Baleskin[1], M A Bubnova[1], A N Kosinov[1], D A Abrameshin[1], L A Romanov[1], B S Nikitin[1]

[1] National Research University Higher School of Economics, Myasnitskaya ulitsa, 20, Moscow, Russia, 101000

email: baleskin1@gmail.com, mbubnova@hse.ru, akosinov@hse.ru, dabrameshin@hse.ru, laromanov@edu.hse.ru, bsnikitin@edu.hse.ru

Abstract. “Virtual space virtual satellite” (VSVS) software solution has been developed for computing the geomagnetic field and studying the results of modelling small spacecraft flight at low orbit with different parameters. The application will allow to replace a number of experiments. In addition, SGP4 and IGRF models are studied due to their role in development of the solution, as are the possible applications in developing solutions for educating high school and university students as well as introducing them to space-themed projects.

1. Introduction

In the modern world, people's interest in space exploration and the development of space technologies, including satellite technologies, is actively growing. The demand for professionals and educational resources in this area is growing accordingly. This leads to a shortage of specialists due to the growing interest and demand. On March 22, 2021, the satellite CubeSX-HSE of the Higher School of Economics was launched. The satellite’s objective is remote sensing of the Earth. In order to provide school and university students with a chance to participate in the interaction between the satellite and the Earth’s mission control center, the decision to develop VSVS was made.

In order to achieve set goals, an integrated approach, which includes development of new educational products, is required.

Considerable amount of practical experiments, embodiment of which can be problematic due to high money cost or uniqueness and danger of modelled situation. At present, only a small few have resources to buy costly equipment or to make equations with the help of demanding engineering software.

Existing software implementations in other areas have already shown the convenience and efficiency of the client-server model. On the one hand, it allows most of the computing to be done on powerful server machines, significantly reducing the system requirements for the user. On the other hand, it allows you to work remotely and lowers the entry threshold. This architecture makes it possible to expand a potentially interested audience, and also allows existing users to work more comfortably.

The developing application is aimed at visual simulation of the behavior of the spacecraft in near-earth orbit and takes into account the surrounding factors, as well as the functionality of the internal modules of the satellite. The aim of the project is to present a user unfamiliar with the field of satellite construction with a full-fledged understanding of the principle of a spacecraft’s operation, the effects of outer space on it.

1.1 Modelling the geomagnetic field; IGRF

Considering modelling the geomagnetic field, IGRF[1] is one of the most widespread models. The thirteenth generation of the International Geomagnetic Reference Field (IGRF-13) is the most recent version of the standard mathematical description of the magnetic field of the Earth. IGRF has been developed by the International Association of Geomagnetism and Aeronomy (IAGA).

IGRF is a series of mathematical models of the geomagnetic field and its yearly rate of change. In areas with no sources at or above Earth surface, the main geomagnetic field can be represented as a negative gradient of the scalar potential V, which can be computed by (1).

$$V(r,\theta,\phi,t) = a\sum_{n=1}^{N}\sum_{m=0}^{n}\left(\frac{a}{r}\right)^{n+1}[g_n^m(t)\cos(m\phi)+h_n^m(t)\sin(m\phi)]P_n^m(\cos\theta) \qquad (1)$$

In the equation, a = 6371.2 km, and the degree of truncation is N=13. The maximum harmonic degree for the yearly change rate is 8.

1.2. Two-line elements

Two-line element sets (TLE) [2] are a format for representing satellite telemetry describing a state of a space satellite. Despite representing the state with sufficient accuracy, predicting later moments in time (relative to the epoch of the given TLE) will produce an increasing error. TLE data are considered valid for a week since the corresponding epoch.

TLE is generally represented as a set of two lines 69 symbols in length, describing the satellite metadata and telemetry parameters.

1.3. SGP model

Simplified Perturbations Models [3] represent a set of five mathematical models: SGP, SGP4, SDP4, SGP8 and SDP8. They are used for computing orbital state vectors of satellites in earth-centric inertial coordinate system. SGP4 is the most widespread model for determining satellite positions using TLE (two-line elements) as a source of orbital elements. The model and its derivatives have been in operation since the 1970s.

1.4. Client-server architecture

Client-server architecture is a type of computation model where the load is distributed between the service providers (servers) and requesters (clients). Clients and servers engage in an exchange of data over the computing network. The client performs a request, then awaits the response. At the same time, the server processes the request and passes the result to the client as a response.

1.5. gRPC

gRPC [4] is an open-source system for defining and performing remote procedure calls (RPC), that uses HTTP/2 for transport and provides features such as authentication, bidirectional streaming and flow control, blocking and non-blocking bindings, cancellation and timeouts.

Similar to most RPC systems, gRPC is based on an idea of defining the interface of the service as a set of methods, which can be called remotely later using specified parameters and response data type. By default, gRPC uses Google Protocol Buffers for data type and interface definition.

2. Overview of existing solutions

Studying and modelling the spacecraft motion before the launch is a complex problem of three-dimensional nature. In order to receive the satellite telemetry precise time is required when the satellite will fly in the zone where receiving the signal is possible. Due to that, modelling requires accounting for long periods of spacecraft motion.

Software is developed for solving the problems in astrodynamics, providing models for computing satellite passes and cost estimation.

2.1. General Mission Analysis Tool

General Mission Analysis Tool (GMAT) [5] is a freeware developed by NASA. GMAT models the motion of a particle in a gravitational field of multiple attraction centers. Trajectory integration is implemented via Runge-Kutta method (2) and allows for periodic correction of the spacecraft velocity vector in order to lessen the unstable component of the motion of the modelled particle.

$$\ddot{R} = G\sum_{i=1}^{n} m_i \frac{(R_i - R)}{|R_i - R|^3} \tag{2}$$

where n is the amount of attraction centers; G is the gravitational constant; R is the radius vector of the spacecraft; m_i is the mass of body i; R_i is the radius vector of body i.

GMAT has been developed to help in improving the skills required to manage orbits and to enhance the comprehension of astrodynamics.

2.2. SIMULATION AND MODELLING SOFTWARE

SIMULATION AND MODELLING SOFTWARE [6] is an orbit simulation package used for orbital analysis throughout the service life of a satellite. Connectivity analysis tool can be used to determine the launch window. Space Environment and Effects Tools group assesses the effect of space on a spacecraft.

2.3. "Orbicraft" designer kit and "Terra" complex

"Orbicraft" designer kit [7] and "Terra" complex [8] together constitute a functional model of a spacecraft allowing to reconstruct the motion of a satellite in Earth orbit, including lighting, geomagnetic field, position of ground stations and targets. Spacecraft constructed from the parts in the kit uses a radio channel to communicate with the "ground control".

"Orbicraft" designer kit and "Terra" complex can be used as a complex for semirealistic simulation dedicated to teaching school and university students the basics of development, design, assembly, testing and operation of a spacecraft.

2.4. Semirealistic simulation complex SXL-SCOE-ADCS-01

Semirealistic simulation complex SXL-SCOE-ADCS-01[9], developed and assembled at SPUTNIX ltd for testing and development of attitude determination, control and navigation systems (ADCNS) of small spacecraft. Devices that constitute the complex imitate the effects of space factors on the studied object, specifically, factors required for sensors and actuators of ADCNS to function properly. The devices allow for free rotations around the center of mass, variable magnetic field, solar radiation and navigation system signals.

The complex is dedicated for on-Earth experimental development and research of ADCNS algorithms.

2.5. ExoOPS™ - Mission Design

ExoOPS™ - Mission Design [10] is a cloud-based product for development and planning of space satellite missions, including modelling of constellations, launch strategy optimization, choosing a propulsion system and assessing its effect on the satellite.

The software allows to receive telemetry from a spacecraft and to receive manoeuvrings commands from the operator. The features also include issuance of telecommands for ExoMG™ thrusters.

2.6. Trajectory Browser

Trajectory Browser [11][12] is a web application for designing trajectories to small bodies and planets. Provided information includes date of launch, flight duration and change in velocity. The features also include visualising the heliocentric system and the mission and parameter-based orbit search.

2.7. Debris Assessment Software (DAS)

Debris Assessment Software (DAS) [13] is a product dedicated to assessing the compliance of a spacecraft to NASA requirements for reduction of orbital debris. However, DAS is implemented with regards to debris formation and does not assess structural reliability of a spacecraft.

3. The model of "Virtual space virtual satellite"

A client-server architecture was used in development of the "Virtual space virtual satellite" software solution (VSVS). In this case of modelling the functioning of a spacecraft in orbit, three main categories of entities are considered: clients, a server, and the simulated system itself. Data exchange between the client and the server takes place in parallel to the modeling process, and is described by a sequence of remote procedure calls (RPCs) following a specified interface (API). This model is described in detail in [14].

In order to avoid some of the typical errors, explicit time sampling of the process is required to implement data exchange. Each interval of time of a certain length (step size parameter of the modelling process) is assigned a discrete time value, which, in turn, corresponds to the state of the system determined for this value. By specifying the state transition function $F : S_i \to S_{i+1}$ and the initial state S_o , it becomes possible to simulate further states.

Each discrete state of the system contains information on the state of the modelled spacecraft (including its subsystems), as well as the environment state. Of them, the spacecraft state formed at a certain iteration is unstable until the end of the said iteration. That happens due to the existing dependence between the resulting state and the order of subsystem modelling. During the update of a state of a subsystem, it is unknown whether another subsystem will later modify the state. On the contrary, the environment state can be inferred from the simulation parameters and the previous states, making it stable. The state transition function thus takes the form $T : T(S_i, e_{i+1}) \to S_{i+1}$, where S_i, S_{i+1} are the current and new system states, and e_{i+1} is the new environment state.

One of the ways of optimizing modelling of spacecraft functionality can be done by parallel and distributed computation of certain parameters. This way, the simulation of the position of the satellite and the state of the environment can be calculated independently of the calculation of the satellite's new state. Thus, these calculations can be implemented as separate components, independent of the main system. Such components can be called "services". At the same time, since such calculations are necessary for each iteration of the simulation, but they do not require information about the state of the apparatus, it becomes possible to compute these parameters in batches.

In addition to the listed advantages, such an architecture allows for scaling of the allocated computing power to the needs of each user. When simulating spacecraft with different configurations of sensors, and therefore with different functionalities, different calculations are performed, both in essence and in the complexity of the calculations. In particular, not all real spacecraft are equipped with star sensors, which may require significant computing power to visualize their operation in educational software. With that in consideration, in the absence of certain systems in the design of the simulated spacecraft, the modelling of the operation of these systems on the server segment of the software solution can be simply "turned off", thereby decreasing the amount of resources used.

In most system simulations, the internal configurations are considered only to a certain extent. At the highest level of abstraction, the system is viewed as a "black box" with a state and a mechanism for transition from the past state to the next. Further, with increasing "depth" of consideration, the system is subdivided into a set of interacting "black boxes" - subsystems described in a similar way. As the "depth" grows, the number of simulated elements grows, and along with that the complexity of modelling rises. Alongside that, the system has to service the specified amount of users at sufficient quality with limited resources. With the necessity in mind of providing the users with freedom to work with different modelling parameters, including different options for the satellite's payload and service load, the spacecraft is considered to be a set of functional subsystems - modules, such as flywheels, magnetic coils and sensors. Since the solution focuses on CubeSat spacecraft of size 1 to 3 units (1U-3U), a certain size of a module can be considered a unit - “size of a slot”, thus allowing to represent both the module sizes and the capacity of a certain spacecraft size in “slots”. Additionally, the “slots” of a certain spacecraft can be indexed and mapped to subsystems and their states, simplifying the modelling process.

Since the system is modular by nature, each subsystem can be represented in the same way as the spacecraft. That makes it possible to form the next state of the spacecraft by setting its initial value (such as the value of the preceding state) and updating the parameters of the spacecraft itself before sequentially transitioning each subsystem to the next state.

In order to model the user-defined algorithm, the components for checking and executing the provided code are required, as are the data exchange system and the API. The system for checking the code is generally implemented by limiting the use of the standard library of chosen language and prohibiting the use of third-party libraries, and code execution means, in effect, compiling the code with the API and the data exchange system and executing the resulting binary. Implementation of the API on the side of the user program is an interface in form of a set of types and functions using the data exchange system to send and receive messages, specifically - commands. For the main system, the received commands are pushed into a queue before being extracted and executed in the process of modelling by the command processor.

In the data exchange between the server and the client, such parameters as the amount of states computed per second of modelled time, the share of time allocated for computation of one state for one user, the delay between the client sending the request and receiving the response. Furthermore, the amount of time since the beginning of the modelling process measured at client and at server can differ. While introducing the aggregation of the modelled states increases the delay between the action of a user and the reaction of the system being displayed, it also reduces the average time of receiving one state. Given that the delay between the command being sent and the reaction being received is significant in the case of real spacecraft and control centers as well, the delay added by aggregation is insignificant. The effect of that delay can be additionally decreased by limiting the user-system interaction between the user and the modelled system.

The amount of system states, or “frames”, requested at once, can be computed from the time since the last request, the amount of frames used and the expected period of time till the next request. In addition, the minimum buffer size, the critical size when new frames have to be requested can be inferred from the amount of time required to request new frames and the rate of frame usage: the critical size can be no less than the result of multiplication of the two parameters. The criterion for sending the request can also be defined as the amount of remaining frames being equal to the critical size multiplied by some factor. Throughout the modelling process, that factor can be corrected corresponding to the request statistic. For instance, in the case where the amount of frames remaining after the request is performed is less than the critical size (“failure”), the factor can be increased by some value k. In the alternative scenario (“success”), it can be reduced by the same amount. Additionally, k can be modified by being doubled or halved in case of successive “failures” or “successes” respectively, while keeping the factor in a certain interval. For instance, the factor can be limited to being between 0.5 and 17.5 with the initial value being 2. For k, the initial value can be set to 0.5. When the limits are reached, k is no more doubled or halved, until the factor is not at the limit.

4. Overview of “Virtual Space Virtual Satellite”

“Virtual Space Virtual Satellite” is a web browser application, which allows to model a flight process of small spacecraft made in formfactor of cubesat standard varying in size from 1 to 3 units (1-3u). In order to decrease minimal system requirements to the user's personal computer the client-server architecture was used in the development. This architecture allows to redirect a part of calculation to server.

VSVS provides a wide range of features, including performing computations and practical works such as using the flywheel to change spacecraft’s orientation or obtaining magnetometer readings. It can be used to gain professional skills in controlling and programming satellites, flight monitoring and more. Among other things, the application has a high level of integration, which allows the user to create their own labs and practical exercises using additional development tools.

To implement client-server interaction, the gRPC-web technology was chosen. The communication interface is written in Protobuf (Google Protocol Buffers) and for data transport HTTP/2 is used, gRPC is based on the idea of defining a service interface as a set of methods for remote call. By default, Google Protocol Buffers is used along with gRPC as the interface definition language for describing services and message structure. Within the software, services are defined as separate computational modules capable of receiving and executing remote procedure calls, which allows communication between the client and the server, as well as between some server services, such as, for example, the IGRF module described above, which uses the results of SGP4 computations as input.

The server side of the software consists of several components called services. Each of them is responsible for resolving a specific task.

1) SGP module - calculates the position of the vehicle for the given time and TLE, which contains the orbit parameters. The SGP4 mathematical model is used for calculations. The position of the satellite is necessary to simulate flight cycloramas, it allows one to get acquainted with the capabilities of the spacecraft, and also sets the calculation parameters for other services.

2) IGRF module - calculates magnetic vector and magnetic scalar potential at the location of spacecraft's sensor. To optimize the number of calculations, it was decided to separate this process into a separate service, although the result of calculations of the SGP module is used as input data. Calculations from this module are used in the operation of the apparatus modules, such as: a magnetometer, electromagnetic coils, a payload that studies the intensity of the magnetic field.

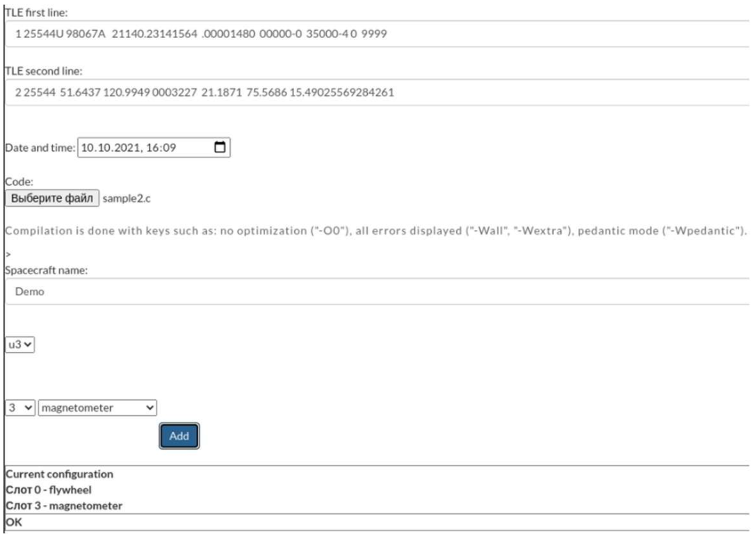
3) Code processing module - responsible for compiling loaded into the system user code which specifies modelling parameters. The user created program code allows to fully simulate the functioning process of the satellite and controls the operation of its subsystems.

4) Assembly module - verifies correctness of satellite's internal systems service load and payload variant sent by user, returns already created configuration if requested with the unique keyword, allows to create brand new system configurations. Determines the functionality of the spacecraft, allowing it to simulate the operation of the systems of the device.

5) Aggregation module - responsible for simulating the functioning of the spacecraft, by interacting with all the services mentioned above. Upon request from the client, transmits the required number of frames of the current spacecraft's state and the environment.

The client side of the software consists of web pages. At the top of the page is the application navigation menu. In total, the main functionality of the project is separated between two pages, which are: "Calculations" and "VSVS". To navigate between different sections of the "Calculations" page a drop-down menu may be used. The user is given the opportunity to calculate the magnetic field at a required point, by TLE at the current time, or by TLE at a user-specified time. By pressing the "VSVS" button user will be redirected to small spacecraft flight simulation page. In the beginning, users are required to input the TLE data for orbit determination before declaring the internal configuration of a cubesat of previously chosen size, ranging from 1 to 3 units(1-3u). After that, the user can upload custom code that implements the functionality of the previously added internal modules and their interaction with the environment (figure 1).

After adding a custom code, the user can click on the "Proceed to VSVS" button, after which the simulation will begin and a scene will render below, demonstrating the current position of the vehicle in orbit. The user can rotate the scene, change the angle and point of view, zoom in and out of the scene. Below the demonstration, the data is displayed in two columns: the first one is the system readings, the second is the readings from the sensors selected by the user in the internal configurator (figure 2, 3).



TLE first line:
1 25544U 98067A 21140.23141564 .00001480 00000-0 35000-4 0 9999

TLE second line:
2 25544 51.6437 120.9949 0003227 21.1871 75.5686 15.49025569284261

Date and time: 10.10.2021, 16:09

Code:
Выберите файл sample2.c

Compilation is done with keys such as: no optimization ("-O0"), all errors displayed ("-Wall", "-Wextra"), pedantic mode ("-Wpedantic").

>
Spacecraft name:
Demo

u3

3 magnetometer

Current configuration
Слот 0 - flywheel
Слот 3 - magnetometer
OK

Figure 1. Input fields for entering the configuration and setting parameters of the orbit

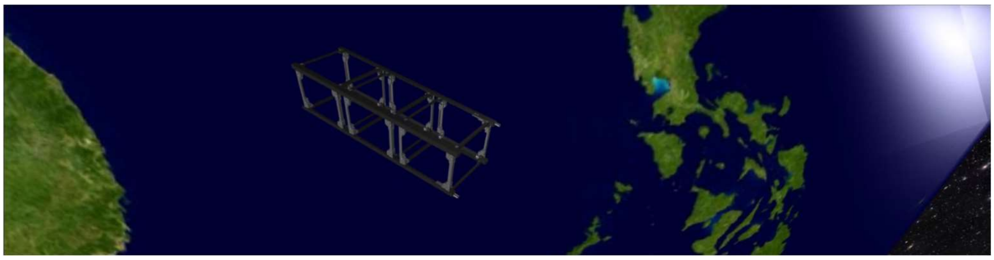
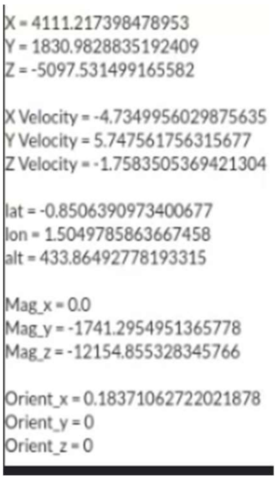


Figure 2. Small spacecraft in low-earth orbit, modelling process



```
X = 4111.217398478953
Y = 1830.9828835192409
Z = -5097.531499165582

X Velocity = -4.7349956029875635
Y Velocity = 5.747561756315677
Z Velocity = -1.7583505369421304

lat = -0.8506390973400677
lon = 1.5049785863667458
alt = 433.86492778193315

Mag_x = 0.0
Mag_y = -1741.2954951365778
Mag_z = -12154.855328345766

Orient_x = 0.18371062722021878
Orient_y = 0
Orient_z = 0
```

Figure 3. Real-time data output

5. Introduction to the design and training activities

“Virtual space virtual satellite” web application was tested at partner schools of the National Research University “Higher school of economics” (NRU HSE). Initially, the test was conducted on a group of teachers that supervise school students participating in aerospace competitions, since increased attention is paid to teachers planning to supervise school-based space-themed projects. The teachers conducted the preoperative testing of the application and rated the quality of the implementation. Based on the feedback the application was improved and a decision was made to progress to testing the application on the students of the partner schools. At the Laboratory of Space Vehicles and Systems' Functional

Safety the students attempted to use the solution in a series of practical works to create their own CubeSats and launch them into low-earth orbit in order to receive telemetry from the satellite later on. According to the received feedback, the application has a user-friendly interface with many tooltips allowing all of the students to pass the series of practical works. According to the devised plan, the products will be used in both the education of specialists and the programs of additional education.

6. Conclusion

The developed software has features for computing the geomagnetic field as well as to studying the effect of different parameters on the results of modelling the small spacecraft flight at low-earth orbit, thus allowing for replacement of a number of experiments. The product allows to monitor the modelling process and for real-time output of the results. The application is getting ready for server deployment for client load reduction.

The product was tested at the partner schools of the NRU HSE and can be used in the programs of additional education as well as space-themed projects and events.

7. References

[1] IAGA V-MOD Geomagnetic field modeling: International Geomagnetic Reference Field. URL: https://www.ngdc.noaa.gov/IAGA/vmod/igrf.html
[2] NORAD Two-Line Element Sets Current Dat URL: https://www.celestrak.com/NORAD/elements/
[3] Wei D and Zhao C Y 2010 An Accuracy Analysis of the SGP4/SDP4 Model *Chinese Astronomy and Astrophysics* **34(1)** pp 69-76
[4] gRPC URL: https://grpc.io/
[5] Berthoud L and Walsh J 2017 Using mission analysis software GMAT to develop skills in astrodynamics *Proc. of the 45th SEFI Annual Conf. 2017 - Education Excellence for Sustainability* (Azores: European Society for Engineering Education (SEFI)) pp 1116-23
[6] SIMULATION AND MODELLING SOFTWARE URL: https://steantycip.com/space-operations-simulation-and-modelling/
[7] SPUTNIX constructor "OrbiCraft" URL: http://orbicraft.ru/orbisat#
[8] SPUTNIX Complex of space environment simulators "Terra" URL: http://orbicraft.sputnix.ru/doku.php?id=terra
[9] Semi-natural modelling stand SXL-SCOE-ADCS-01 URL: https://sputnix.ru/ru/oborudovanie/ispytatelnye-stendy-new/stend-polunaturnogo-modelirovaniya-sos
[10] ExoOPS™ MISSION DESIGN, OPTIMISATION & OPERATION SOFTWARE URL: https://exotrail.com/exoops/
[11] NASA Ames Research Center Trajectory Browser URL: https://trajbrowser.arc.nasa.gov/user_guide.php#overview
[12] NASA Trajectory Browser URL: https://www.nasa.gov/centers/ames/engineering/divisions/spaceflight/flight-dynamics/trajectory-browser/
[13] Debris Assessment Software (DAS) Reentry Risk Analysis: Mission Planning for Compliance with NASA Standards URL: https://orbitaldebris.jsc.nasa.gov/library/das3_0/das3.0_usersguide.pdf
[14] Baleskin V, Kosinov A, Romanov L and Motajlenko I 2021 Client-Server Implementation in Solving the Problem of Modeling the Small Spacecraft Flight at Low-Earth Orbit *2021 Int. Seminar on Electron Devices Design and Production (SED)* (Prague: IEEE)

Acknowledgments

Authors note that this article was prepared within the framework of the Basic Research Program at the National Research University “Higher School of Economics”.