

Multilevel Functional-logic Models in 3D-simulators of Electrical Equipment

A.L. Safonov¹, D.I. Kopeliovich¹

safonoval@yandex.ru | dkopeliovich@rambler.ru

¹Bryansk State Technical University, Bryansk, Russian Federation

At present, much attention is paid to the training of specialists who operate at responsible and hazardous industries, using various types of benches and simulators. A special place among them is occupied by three-dimensional computer simulators, which are characterized by low cost and sufficient quality of training. The paper presents an approach to developing an integrated control system for computer simulators; its functional chart is proposed. In this work, the problem of modeling complex technical systems taking into account the criteria of complete and realistic modeling and performance is specially noted. There is also an approach to constructing simulators with variable technical object models depending on the target and with a given level of detail. The results of the work were tested in the development of simulators of electrical equipment.

Keywords: computer simulators, computer graphics, specialist training, complex technical systems.

1. Introduction

Gaining practical experience in dealing with complex technical systems is an integral part of training. This is primarily connected with hazardous industries, as well as with those where errors and accidents can lead to significant economic losses. Such activities are characterized by the use of expensive equipment, maintenance and repair of which are also not cheap, for example, the control of oil and gas equipment units, electric power facilities, special types of vehicles.

Modern technologies allow to replace the use of full-scale training in testing grounds by training methods in specialized computer systems. The most interesting application for such tasks is three-dimensional computer simulators (3D simulators), which realistically simulate technical systems, form highly qualified specialists [1].

In open sources, including the market of information systems, a significant number of computer simulators are presented, although the problem of developing realistic simulators has not been fully solved yet. First of all, this is connected with the modeling of complex technical systems, which should be characterized by sufficient detail of objects and at the same time have high performance. The paper deals with the methods of modeling such systems.

2. Simulators

Early graphic applications were made practically from the scratch, now they are developed by means of ready-made tools (game engine). This allows developers to reduce project time and improve its quality.

Along with the availability of a wide range of tools for developing 3D-simulators, there are several tasks of their implementation for complex technical systems (for example, electricity generation facilities), which require the development of a methodological approach that takes into account the peculiarities of the subject area, limitations on hardware and software, the possibility of combining with other components in integrated software solutions [4].

The most important issue in developing simulators is the balance of performance/details of the model (price/quality).

The main aspects affecting the performance of the model concerning simulators are:

- 1.Perfection of 3D-engines.
- 2.Detail level of objects.
- 3.The quality of developing application logic (algorithm optimality).

No matter how advanced is the logic of operating with models meaning its performance in the framework of developing simulators and how "fast" is the engine on the selected platform,

the speed and lack of delays in the operation of the computer application largely depends on the properties of 3D models of technical systems.

Objects in 3D application development systems can be created in different ways: within the environment, by importing a model, they can have different shapes and complexity, but all of them are represented as polygon meshes which is a set of vertices, edges and faces that define the shape of a polyhedral object in 3D computer graphics and solid modeling. Objects are colored in certain colors or specially prepared raster objects (textures) are placed on them with the help of specially written shaders (programs designed to provide vital information about your video card and graphics processor (GPU)). To display a polygon mesh, it must pass through a video processor for processing and rendering.

Models of complex technical systems consist of a large number of bodies with different geometries. Often, most of them contain curved features that require thousands or even more elements to be represented by polygon meshes in order to render the model correctly.

Several approaches can be used to reduce the computational complexity of models [1]:

1. Using simple shaders.
2. Visualization technique with different levels of detail (levels of detail, LOD).
3. Using MIP (a texturing method that uses multiple copies of the same texture with different details. The name comes from latin "mutuum in parvo" which means "much in small").

Professional 3D simulators are widely represented at the market. For example, KIT 3D system of Dipol company (Fig.1 shows a simulator operating with power lines), simulators of United 3D Labs company and other solutions [5].



Fig. 1. KIT 3D power line model of Dipol company

They are made in the form of separate applications, which allows to use all the resources of the computer station, in contrast to the system on the web-platform considered in this paper. For complex models and simulators these systems are demanding of hardware. At the same time, some of the objects of the environment related to the subject matter of the simulator are not presented accurately and fully enough. For example, as Fig. 1 shows, wire fastening to the power line insulators does not match the correct solution with the help of helical ties.

3. Selection of 3D-simulator System Architecture

To define the rational architecture of the system, which determines the main ways of interacting its elements, the key point is selecting the target platform of 3D applications (scenes). This will largely affect the properties (capabilities) of both the visual part of 3D simulator and the choice of technologies and methods of interaction with other parts of the system.

Evaluation criteria of the target platform are the following:

1. Ease of control.
2. Restrictions on computing resources (object granularity, number of polygons).
3. Memory size limits.
4. Graphics restrictions (quantity and quality of textures, using shaders).
5. Networking with the server (possible architecture, messaging frequency, message size, etc.).
6. Methods of distribution (sales): SaaS-platform, full packaged type, distribution through app stores (App Store, Google Play, Amazon).

There is a variant of implementing 3D applications by means of cross-platform supporting, i.e. the possibility to implement one developed project through several system types. In most cases, this is problematic due to the different requirements for different technologies (specifications, file structure, I/O devices and product distribution methods). Therefore, in 3-D simulators project it was decided to focus on one specific option, and in the future to develop an alternative option involving own free labour force (for example, an option with large highly detailed models, or, on the contrary, one that is easy to use in mobile devices).

Below are the options for implementing 3-D simulators project, which are more or less suitable for it.

1. Mobile devices.
2. Personal computer (Windows, Mac, Linux).
3. Consoles
4. Web application in web browsers.
5. VR (virtual reality environment)

Analyzing the features of these platforms and the need to link 3-D simulators project in a single system with other modules (training, statistics, administration) we determine to use Web-platform (players, WebGL) as the main implementation option with the ability to switch to another platform quickly.

Fig. 2 shows the architecture of the software system, which is a classic version of 3-level web-system: client-server-data stores.

Game engine is the central software component of computer video games and other interactive applications with graphics processed in real time. It provides core technologies, simplifies development, and often gives the game an ability to run on several platforms. The game engine includes a whole range of applications, including rendering engine (visualizer) for 2D or 3D graphics, physics engine, sound, animation, artificial intelligence, network code, memory control, graphics scenes. That is, all parts of the code written by programmers for the development of the game are components of the engine [2].

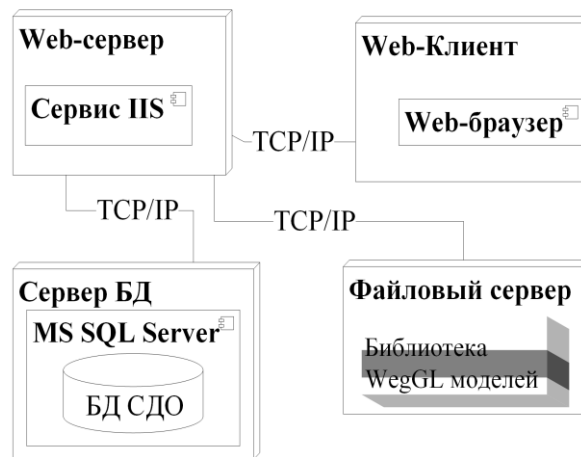


Fig. 2. General system architecture.

At present, there are a huge number of game engines, from GameMaker, Cocos2D, Marmalade to Unity3D and Unreal Engine [3]. Many issues are not disclosed in official sources, you need to seek help in the Internet resources.

Game Maker is more focused on beginners just starting to develop games. It is rather easy to learn it, but the possibilities of Game Maker are limited.

Cocos2D is optimized for the development of two-dimensional games. The engine is not suitable for modeling data simulators.

With the help of Marmalade such well-known games as "Plants vs. Zombies" and "Godus" were created. One of its most significant drawbacks was that the community of Marmalade specialists is not big.

So, among the major alternatives only Unreal and Unity are the most common options for developing 3D applications. Unity3D engine was selected for 3D simulators project.

The key findings of comparison:

Unity is much easier to learn and understand. C# language is used for Unity scripting.

Unity is cross-platform, i.e. the same code written in Unity engine with minimal changes can be transferred to different platforms (PC, Mac, Android, iOS, Web, game consoles)

Unity is widely used by other developers, so there is an opportunity to get high-quality and fast support and take advantage of a large number of already developed resources.

4. Development of 3D simulators

In the system of 3D simulators Unity-application is an interactive part of the system, where the user interacts directly with the models of technical objects (he trains or performs tasks). The application itself is cross-platform, and in the developed system this application is a WebGL-package, executed in the browser environment. The tools of Unity 3D allows to develop a virtual environment, including that one operating within the framework of certain physical laws, and then put into this environment a user-controlled object giving it the functions of interaction with the surrounding space and objects.

For example, Fig. 3 presents the model of 110/10 kV distribution station (Unity-application).

Here are the principal units of the system: steel structures, elements of operational control (cabinets, control panel), the building with the equipment. The controls are interactive: there is a possibility to change the state of objects (turn on/off switches, open doors, etc.).

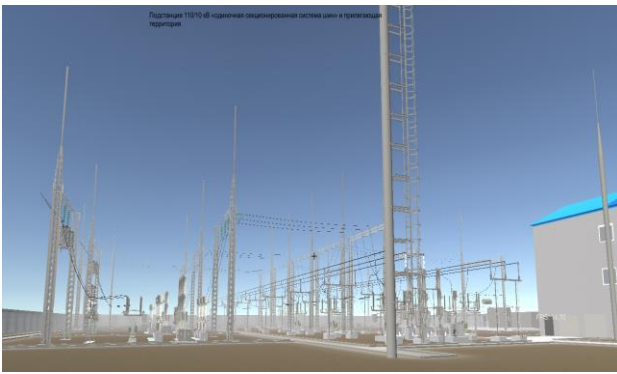


Fig. 3. Unity-application with the model of 110/10 kV distribution station.

Software development allows to provide for the interaction of application objects, including the process logic of simulator tasks.

3D simulators should provide the formation of competencies, the main of which are:

1. Knowledge of the design of technical systems in the subject area.
2. Ability to identify faults in the design and operation modes of technical objects.
3. Skills of safe operation of hazardous technical systems.

According to the competencies above, the simulator can operate in one of three modes:

1. Excursion mode – moving around specific points of the simulator with the demonstration of the main model elements and their detailed description;
2. Inspection mode – free movement of the player (trainee) on Unity scene (technical system models), followed by testing of the objects seen, for example, a list of faults;
3. Practice mode – performing tasks of 3D simulator in real-time.

Let us consider the practice mode as the most interesting variant of the system operation. It involves the performance of tasks by trainees and evaluation according to a pre-developed scenario that corresponds to the data model presented in Fig. 4.

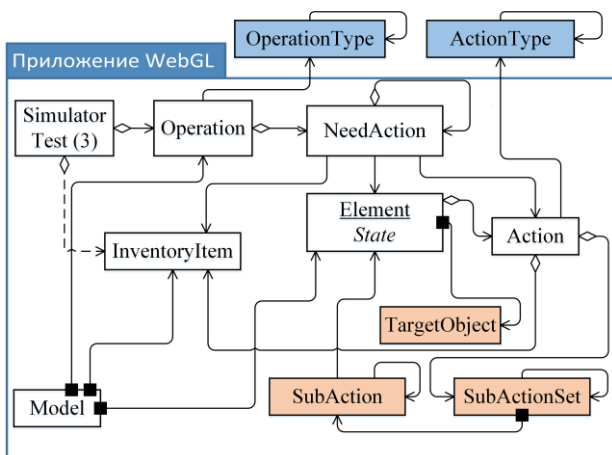


Fig. 4. Data model of the simulator (diagram entity-connection).

Task performance involves actions in real time, and after completion, checking the list of completed actions and their order in accordance with the options given in the script.

5. Ensuring the Performance of Simulators Using Models of Complex Technical Systems

The information model and using the tools of the proposed approach allowed to develop three-dimensional simulators of technical objects: station areas, power lines, etc.

However, often training how to work with complex systems requires simulation models, which should ensure maximum compliance with the original objects. For example, power distribution units of electrical stations, control panels, etc. These systems have a large number of relatively small objects, which cannot be ignored or greatly simplified in models for simulators- this will lead to a mismatch with the original and disrupt the training process. In such models, with a "direct approach" to modeling, the number of polygons can reach tens of millions, which significantly affects the performance of the system, influencing the time of rendering (frame rendering) and RAM used.

A slight improvement is achieved by using LOD technology – a technique in three-dimensional graphics programming, which consists in the creation of several hierarchically nested levels of detail of the object, which switch depending on the removal of the object from the virtual camera.

For example, Fig. 5 and Fig. 6 show portal structures with different levels of detail. Fig. 5 shows a strut type, which uses a plane with a texture lattice structure as vertical struts, and Fig. 6 presents a multi-polygonal strut model. In addition, one can see the change in the number of elements, in particular sets of insulators when approaching the object.



Fig. 5. Portal structure with low level of detail.



Fig. 6. Portal structure with high level of detail.

However, in many models of technical systems, the use of standard LOD mechanisms is insufficient to achieve satisfactory system functionality even if all the requirements for the quality of the geometric model are met.

In this case, the system of simulators uses a variable functional-logic model of a group of objects or their elements corresponding to a given function of the simulator. In the logical model (Fig. 7) the introduction of a link between a specific simulator task (Simulator Test) and the elements of the model (Element) is provided. This allows to change the model when performing a specific task of the simulator.

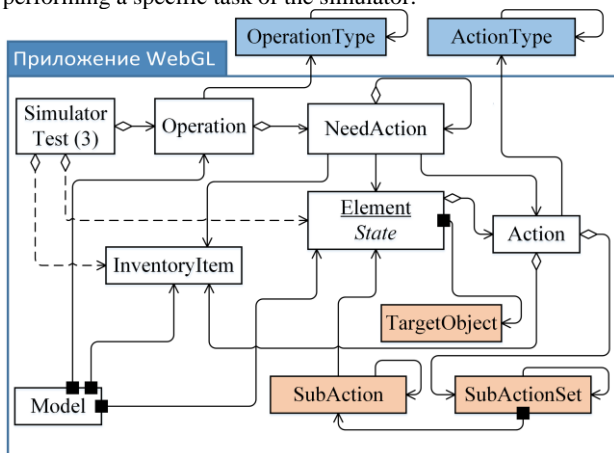


Fig. 7. Simulator data model taking into account the level of detail of the model elements.

According to the model, all possible tasks are listed in the simulator operation mode (task) listing file (xml format) in Simulator Tests section. After selecting a task performed by the simulator, the models of all the elements listed here with the specified parameters are updated, indicating the network path to the updated model.. An example of xml data file is shown in listing 1:

Listing 1. An example of xml data file

```

<SimulatorTests>
  <SimulatorTest Guid="146a67b9-86ed-4af6-
a80c-18cd493e458d">
    <Description>Вывод трансформатора в
ремонт</Description>
    <OperationRefs>
      <OperationRef Guid="b83eaf7e-01c1-4fa2-
8302-305a7911438f" />
      <OperationRef Guid="99008bb5-9460-4de4-
bd18-6b49ef8e5f71" />
    </OperationRefs>
    <InventoryItemRefs />
    <ElementModels>
      <ElementModel GuidElement="dfe3fcda-
c554-48d8-9a69-d3cdf7b9c5b3"
PrefabName="Провода 10 кВ от Т1 к Крун
(детализированная версия)"/>
      <ElementModel GuidElement="7b86609e-
2271-4a83-90a4-fe20b870f71f"
PrefabName="Шкаф управления Крун
(детализированная версия)"/>
    </ElementModels>
  </SimulatorTest>
  ...
</SimulatorTests>
...
<Elements>
  <Element Guid="dfe3fcda-c554-48d8-9a69-
d3cdf7b9c5b3">
    <Title>Провода 10 кВ от Т1 к
Крун</Title>
    <ElementState>Вкл</ElementState>
    <TargetObjects>
      <TargetObject>Rope_Trans1_Razed/Rope_3/
</TargetObject>

```

```

      <TargetObject>/Разъединитель РН-СЭЩ-
110/Деталь6_005</TargetObject>
    </TargetObjects>
  </ActionRefs />
</Element>

```

Thus, to perform a specific task of the simulator, only a small number of element models that are really necessary are loaded in detail, and the remaining elements are represented by default models.

6. Conclusion

The use of the proposed approach allows to develop professional simulators, which model technical systems as realistically as possible. This opens wide opportunities for the formation of professional competencies in the field of technical systems: electricity, oil and gas sector, ad other production branches. The proposed approach to changing 3D models on the basis of not only geometric details, but also the functional-logic model allows to design and use a system of multi-functional 3D simulators of technical systems.

The given approach was tested in the project to develop an integrated system of training to operate electrical installations, performed by JSC "Klinskiy Institute of Safety and Working Conditions" for PJSC "MRSK Severo-zapada", which components are computer simulators. Training systems are allowed to form and certify competences in the field of electrical safety when working with the main types of electrical station equipment and power lines.

7. Acknowledgments

The authors appreciate Klinskiy Institute of Safety and Working Conditions for the methodological and technical support of this work, in particular, Bragin A. A., the head of the Department of Information Technologies.

8. References

- [1] Bunto P.A., Kulikov V.A. Virtual reality tools and simulation models contribute to the efficient and safe operation of industrial facilities. CAD/CAM/CAE Observer. 2018. n.93.pp. 64-69.
- [2] O'Connor K. GPU Performance for Game Artists. <https://www.gamedev.net/articles/programming/graphics/gpu-performance-for-game-artists-r4632>.
- [3] Pasko D.N. Modern game engines. Innovatsionnaya Nauka. Ufa, Aeterna, 2016. n. 2. pp. 128-130.
- [4] Razumovskiy A.I. Development of 3D Simulators. Applied Informatics. 2012. n. 2(38). pp. 78-83.
- [5] Satdarov M.R., Staroverova N.A. Development of the Duomatic 09-32 virtual simulator. Modern high technologies. 2018. n.7. pp. 84-87.