UDC004.9

VISUAL SIMULATION OF FUNCTIONING COMPLEX SYSTEMS

Dovbysheva Kateryna, Gavrilyuk Nikita
Ph.D.,senior lecturer,Denisenko Andrey
OdesaPolytechnicNationalUniversity,UKRAINE

ANNOTATION. The work is devoted to a well-known modeling complex for studying complex technological processes of a hybrid nature using an integrated visual modeling information technology that allows the rotation of data changes in the process ofthe complex technological system.

Introduction. The informatization of society, the growth of the social role of information technologies in various spheres of human activity, the transition to the foundations of intellectual work in the context of rapid transformations in technology and technology require constant improvement and modernization of models and decision-making methods when creating automated control systems.

A large role in this is given to the information technology of visual modeling (ITVM), which is widely used in industry,aviation, cosmonautics, radar,defense complex, railway transport, oil and gas, hydro and thermal power. engineering and other adits, design and operation CTS (Complex TechnologicalSystems).When visualizing the functioning of a technological process,the complexity of the simulated technological process or commplex is essential.

The purpose of the work. Based on the integrated system developed by ITVM, which has the added function of interaction with the DCNET platform, which makes it possible to take into account changes in models during the operation of the CTS.

The main part of the work. The papers [1,2] consider the issues of development existing information technologies for visualizing the functioning of complex technological complexes(CTS)by integrating various software tools in order to increase their functionality.In this work,when developing the MC(Modeling Complex),the animation tools of Unity-technologies and the computational tools of DCNET for numerical calculations were jointly and productively used. CTS state vector with the additional introduction of DCNET feedback with the Unity platform. This makes it possible to significantly expand the functionality of the developed modeling complex MC in comparison with previous developments. In this case,the procedure for forming the state vector is used $\boldsymbol{F}_{\mathbf{In}}\boldsymbol{F}_{\mathbf{In}}$,which is included in the vector transmission format of continuous $Xc(t)$ and discrete $Xd(t\kappa)$ variables from the simulation platform to the multimedia platform,as well as the formation of the inverse vector $\boldsymbol{F}_{\mathbf{out}}\boldsymbol{F}_{\mathbf{out}}$transfer from Multimedia to the simulation platform.Model MC consists of the following main elements: mathematical description of CTS; DCNET: CTS model development, CTS modeling (state vector calculation) XML: state vector structuring and parameterization; Unity: world shaping, workflow visualization.Developed MC can be used as simulators,in the laboratory study of complex technological physical and chemical processes,in testing or testing equipment elements,where there is a characteristic problem of reflecting circuits and its visualization.The mathematical model of the CTS uses the apparatus givenbelow,whichfocusesonthevector$\mathrm{W}_{\mathrm{i}}^{\mathrm{d}}\mathrm{W}_{\mathrm{i}}^{\mathrm{d}}$ discrete-event states of the model in DCNET. It is related to DCNET feedback generation operations, as well as for external influence involving the operator in terms of any collisions.

Inwork$\mathrm{W}_{\mathrm{i}}^{\mathrm{d}}$ $\mathrm{W}_{\mathrm{i}}^{\mathrm{d}}$ –vector of discrete event states,used for feedback from Unity to DCNET. At the same time, additional structure-controlled transitions and positions are introduced into DCNET, corresponding to the current state of the CTS. The development of the mathematical scheme of the selected CTS was carried out in theDCNET editor.

Thus, in the DCNET software environment, the state vector is formed $\boldsymbol{F}_{\mathbf{In}} = [\bar{\mathbf{x}}_{\mathbf{c}}(\boldsymbol{t}), \bar{\mathbf{x}}_{\mathbf{d}}(\mathbf{t}_{\mathbf{k}})]\boldsymbol{F}_{\mathbf{In}} = [\bar{\mathbf{x}}_{\mathbf{c}}(\boldsymbol{t}), \bar{\mathbf{x}}_{\mathbf{d}}(\mathbf{t}_{\mathbf{k}})]$,which is included in the vector transmission format of continuousXc(t)iXd(tк) discrete variables with DCNET in Unity where

$$\bar{x}_c(t) = [\vartheta_1(t|t_k|), \vartheta_m(t|t_k|), P_G(t|t_k|), l(t|t_k|)]^T$$

$\overline{x}_c(t) = [\vartheta_1(t|t_k|), \vartheta_m(t|t_k|), P_G(t|t_k|), l(t|t_k|)]^T$ –vector of continuous-event states; $\overline{x}_d(t_k)\, \overline{x}_d(t_k)$ $=|\mu(p_1) \dots \mu(p_n)|^T |\mu(p_1) \dots \mu(p_n)|^T$ – vector of discrete-event states; $P_n\, P_n$– marking of the n-th position of the DFC of the DN-network.

In this case,the state vector is formed according to the following equation of states of theDFC model:

$$\overline{x}_d(t_k) = \overline{x}_d(t_{k-1}) + |W| * V + \overline{w}_d(t_k) + u_{dl}(t_k)$$

$$\overline{x}_d(t_k) = \overline{x}_d(t_{k-1}) + |W| * V + \overline{w}_d(t_k) + u_{dl}(t_k), \vartheta \qquad (1)$$

where $\overline{u}_{dl}(t_k) = |f_{c/d}(\vartheta_1)\, f_{c/d}(\vartheta_m)\, f_{c/d}(P_G)\, 0 \ \dots \ 0|^{\mathrm{T}}$ $\overline{u}_{dl}(t_k) = |f_{c/d}(\vartheta_1)\, f_{c/d}(\vartheta_m)\, f_{c/d}(P_G)\, 0 \ \dots \ 0|^{\mathrm{T}}$ –vector of discrete-event states fromDFCtoDFC мodel in DCNET; $\overline{w}_d(t_k) = |w_1^d \ w_2^d \ \dots \ w_n^d|^{\mathrm{T}}$ $\overline{w}_d(t_k) = |w_1^d \ w_2^d \ \dots \ w_n^d|^{\mathrm{T}}$–a vector of discrete-event states from Unity to the DFCmodel inDCNET formula1; wherein

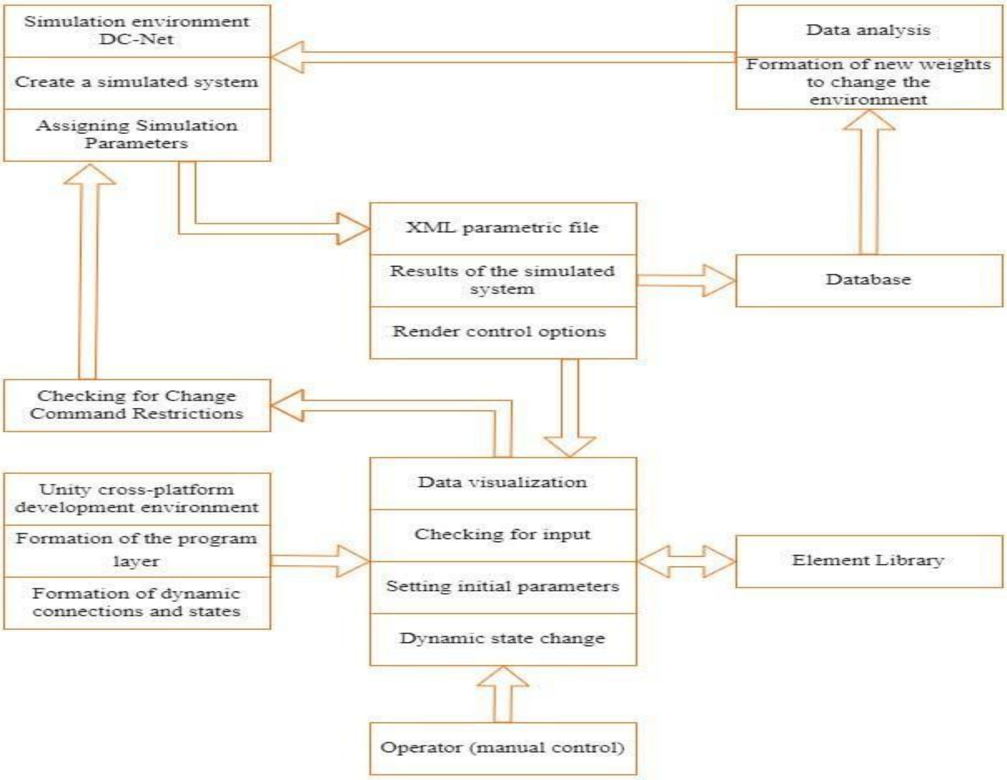
where$y_{F_i}, x_{F_i} y_{F_i}, x_{F_i}$–coordinates ofthei-th objectinUnity visualization;

Accordingly, the movement of objects $m_i m_i$or $m_j m_j$corresponds to coordinates $y_{F_i}, x_{F_i} y_{F_i}, x_{F_i}$according to the expressions formula 2:

$$m_i: \left(\begin{matrix} y_{F_i} = f(N,t) \\ x_{F_i} = f(N,t) \end{matrix} \right. m_i: \left(\begin{matrix} y_{F_i} = f(N,t) \\ x_{F_i} = f(N,t) \end{matrix} \right. \qquad N=1,2\dots k \qquad \text{when} \qquad x^d(t_d) = 1 x^d(t_d) = 1;$$

$$m_j: \left(\begin{matrix} y_{F_i} = f(\mathrm{x}^c(t)) \\ x_{F_i} = f(\mathrm{x}^c(t)) \end{matrix} \right. m_j: \left(\begin{matrix} y_{F_i} = f(\mathrm{x}^c(t)) \\ x_{F_i} = f(\mathrm{x}^c(t)) \end{matrix} \right.; \qquad (2)$$

The closure of the Discrete Continuous Part (DCP) and Continuous Event Part (CEP) parts is implemented using the vector – $\overline{u}_{do}(t_k) = \Lambda * \overline{x}_d(t_k) \overline{u}_{do}(t_k) = \Lambda * \overline{x}_d(t_k)$, the elements of which discretely change the coefficients of differential equations representing the BPC of the model implemented by means of DCNET.On the basis of a parametric XMLfile,the process of managing the animation part is defined.The animation part of the simulated object is implemented from a set of various developed elements. The modeling environment issues a parametric file that contains the visualization control parameters. Data visualization occurs with the help of Unity. The visual block requests the necessary objects in the element library and takes them for display. When we receive the visualization control parameters, a check for the presence of input data begins in order to exclude errors and failures of the visual system. Next, the initial parameters are set, and with each new data, the data changes dynamically with animation on the visual diagram.In an emergency,the operator can turn on the manual control switch of the visual part of the system and command changes to the simulation parameters in DC-Net, having previously passedthecheck forsystem limitations. Also (Picture1)shows a change in the simulated system with the help of feedbackt hrough self-learning.All parameter data is collected for this purpose in a database. And then the data analysis is started and, based on this data, the parameters of the scales in the simulated system will change. To train the system,it runs about100times to get the data and reduce the likelihood of errors to a minimum. For this, server logic and a mathematical block for calculating and analyzing data are implemented.


Simulation environment DC-Net
Create a simulated system
Assigning Simulation Parameters
Data analysis
Formation of new weights to change the environment
XML parametric file
Results of the simulated system
Render control options
Database
Checking for Change Command Restrictions
Unity cross-platform development environment
Formation of the program layer
Formation of dynamic connections and states
Data visualization
Checking for input
Setting initial parameters
Dynamic state change
Element Library
Operator (manual control)

Picture1–Structure of the modeling complex

Conclusions. Thus, an MC has been developed that implements a platform forthe formation of various virtual laboratory stands–simulators that represent the functioning of multi-mode systems with a logical-dynamic nature of functioning. Compared to previous developments, due to introduction of feedback, MC allows to take into account changes in data during the operation of the CTS and automatically select the appropriate mode of operation of the CTS at a given moment in time.This makes it possible to continue further research in the aspect of artificial intelligence.

REFERENCES

1. Denisenko A. V., GurskiyA.A. Information technologyof visualization for technological processes for research modes offunctioning of complex technological systemsISSN 1681-6048 System Research &amp; Information Technologies. 2021. № 2. C 74-84.
2. Kurt Jensen,Lars M.Kristensen.Coloured Petri Nets.Modellingand Validation of Concurrent Systems—Springer, 2009