Computer Science Journal of Moldova, vol.5, no.3(15), 1997

SimW — software for automatic creation of problem-oriented simulation systems

L. Burtseva

Abstract

SimW, a software for automatic creation of problem-oriented simulation systems, is described in detail. Its main features, appearance, principles of operation are presented.

1 Introduction

Problem-oriented simulation systems are at present the most appreciable contribution of simulations as directions in intellectualization of computer technologies. Ensuring dialog with the user on his own language, such systems use technologies, delivered by other branches of Computer Science. However the construction of such a system requires participation of specialist in the field of Computer Science. The user usually has only executable module which implements his simulation model in some way. Software that automatically builds problem-oriented simulation systems on the base of such executable modules helps avoid addressing to programmers. Group of researchers "Simulation systems" has build three simulation systems, oriented to different domains[1]. The architecture of problem-oriented simulation system was than extracted and software tools for their automatic creation was designed by the author on the base of these practical results.

2 SimW purposes

SimW, a Simulation system Wizard, is a software tool for automatic creation of problem-oriented simulation systems. **SimW** allows to build on the base of executable module of simulation model an executable module working under Win32 and implementing the following possibilities:

- Input of experiment parameter values through the dialog in terms of specified domain.
- Running of simulation experiments with these values.
- Representation of results in terms of application domain.
- Analysis of results by means operation of which is explained in terms of application domain.

3 SimW appearance

SimW presents itself as an IDE (interactive dialog environment) under Win32. After starting a main window appears with only one item in its menu. This item deals with the projects. Choice (or initialization) of the project is the obvious step of every **SimW** session. Non-standard buttons in the decoration line are also connected with projects: to open one or to initialize one.

The following obligatory step is a choice of simulation model which will be the base of the future problem-oriented simulation system. Decoration line is changed, standard buttons are added at this step.

After the project and the model are chosen all items needed for the subsequent operation are added to the main system menu.

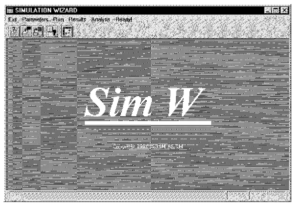


Fig. 1 Main menu

Though all three components of problem-oriented simulation system architecture[st1] have to be “filled” for getting the correct result, it is possible to arrange them in a free order. At this stage the multiwindow mode is supported and arrangement of each component is made in a child window.

Menu item “Ready!” compounds an executable module of the problem-oriented simulation system from the project. This item is active at any stage of the project completeness.

4 SimW Structure

Let us consider now a step-by-step (according main menu items) session of **SimW** utilization.

4.1 “Project” item — load (open new) project

SimW creates a project file keeping of the information about the future problem-oriented simulation system. This information is required for building of an executable module. It can be saved in the process of work. By opening project file at the beginning of the next session all arrangements done earlier are restored.

4.2 “File” item — choose a model to be the basis of the project.

Only the simulation model executable module is necessary for **SimW** work. In this version of **SimW** source models are supposed to be written in simulation language SOL[2], but since **SimW** addresses a model at level of experiment parameters values input and simulation results output, the language of model realization is not a principle feature.

4.3 Construct the target problem-oriented simulation system.

A problem-oriented simulation system built by **SimW** presents itself as an MDI (Multiple Document Interface) with the following main menu:

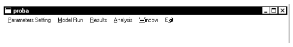
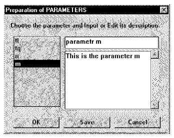


Fig.2 Main menu of a target problem-oriented simulation system

The main menu is uniform for all problem-oriented simulation systems created by **SimW** either as appearances of dialogs, results windows or analysis windows. Consider now how translation of simulation language constructions in terms of application domain is implemented and how it is mirrored in the target problem-oriented simulation system.

4.3.1 “Parameters” item — input parameters



Preparation of PARAMETERS

Choose the parameter and Input or Edit its description

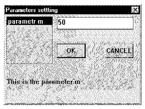
tl
tq
n
m

parametr m

This is the parameter m

Fig.3 The window of input parameters dialog arrangement.

In this window an user carries out arrangement of dialog of input of the experiment parameters values in the target problem-oriented simulation system. It is required that he chooses a name of an parameter in the listbox where the names of all parameters of chosen simulation model are listed. After the choice a possibility is given to assign a name to this parameter in terms of application domain and to give its a more detailed explanation in the proposed subwindow. In the case the repeated choice of this menu item takes place information brought earlier will be restored but can be changed. It is represented below how would the information entered by user in the parameter values input dialog of target problem-oriented simulation system be mirrored.



Parameters settting

parametr m

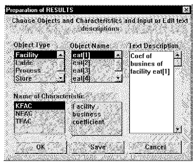
50

This is the parameter m

Fig.4. Example of parameter choice window in the target system

The text that was input in the editing area at design step is flashed at the help line when user of the target problem-oriented simulation system chooses a parameter. In the parameter choice listbox of the target problem-oriented simulation system their names appear in terms of application domain which were given at the design step.

4.3.2 "Results" item — results view arrangement



Choose Objects and Characteristics and Input or Edit text descriptions

Object Type: Facility
Table
Process
Store

Object Name: eat[1]
eat[2]
eat[3]
eat[4]

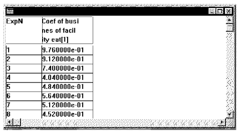
Text Description: Coef of busines of facility eat[1]

Name of Characteristic: KFAC
NFAC
TFAC

Facility business coefficient

Fig.5 Results view arrangement window.

In this window a user gets the possibility to prepare translation of statistical information which will be obtained during simulation: name of objects of simulation and their characteristics can be replaced by corresponding terms of application domain. In the case of the repeated choice of this menu item takes place information brought earlier will be restored but can be changed. In this version of **SimW** representation of the results is made in the most simple way — in the table form. Information entered in the description window of each result will be the name of the corresponding table column.



ExpN	Coef of busines of facility eat[1]
1	9.760000e-01
2	9.120000e-01
3	7.400000e-01
4	4.040000e-01
5	4.840000e-01
6	5.640000e-01
7	5.120000e-01
8	4.520000e-01

Fig.6. A table of results

We intend to make the representation of results more advanced in future versions of **SimW**. Addition of corresponding constructor will allow to arrange results window so that information either is represented by the given GUI (Graphic User Interface) items during experiment or sets in these items after simulation finishes.

4.3.3 “Analysis” item — analysis menu arrangement

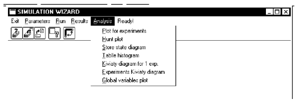
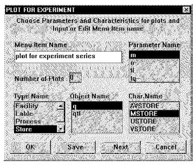


Fig.7 The menu of arrangement of results analysis.

Using submenu items of this menu item the user forms menu of analysis of statistical results of simulation experiments for the target problem-oriented simulation system. Items of this menu could be series of plots, histograms or tables for particular, chosen at the design step, model objects and their statistical characteristics. User defines names of menu items of the menu of analysis means for the target problem-oriented simulation system at this stage of designed in the provided editline. For each menu subitem corresponding dialogs are supported. Consider now these dialogs in details.

4.3.3.1. "Plot for experiments" subitem — plots for experiment dialog arrangement

In this window the user gives a name to the corresponding menu item of the target problem-oriented simulation system and defines dependencies of what particular characteristics will be shown on the plot built by problem-oriented simulation system by the choice of this item of analysis menu.



PLOT FOR EXPERIMENT

Choose Parameters and Characteristics for plots and Input or Edit Menu Item name

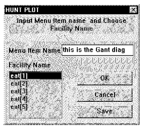
Menu Item Name: plot for experiment series

Number of Plots: 0

Fig.8. A dialog of arrangement of plot of changes of the characteristics values from experiment to experiment.

4.3.3.2. "Hunt plot", "Store state diagram", "Table histogram", "Global variables plot" subitems — arrangement of similar dialogs: Hunt Plot, Store State diagram, Table Histogram, Global Variable Plot

In these windows the user defines particular Gant plots, store diagrams and table histograms, which will be included into the menu for analysis of the target problem-oriented simulation system, and gives names to the corresponding menu items.



HUNT PLOT

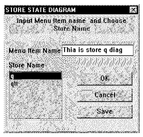
Input Menu Item name and Choose Facility Name

Menu Item Name: this is the Gant diag

Facility Name: eat[1]

eat[1]
eat[2]
eat[3]
eat[4]
eat[5]

Fig.9. A dialog of arrangement of the Gant plots menu items.



STORE STATE DIAGRAM

Input Menu Item name and Choose Store Name

Menu Item Name: Thia is store q diag

Store Name: q

q
qtt

Fig.10. A dialog of arrangement of store diagrams menu items.

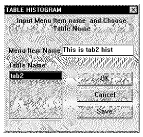


TABLE HISTOGRAM

Input Menu Item name and Choose Table Name

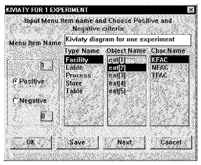
Menu Item Name: This is tab2 hist

Table Name: tab2

Fig.11. A dialog of arrangement of table histograms menu items.

4.3.3.3. "Kiviaty diagram for 1 exp." subitem — dialog of arrangement of menu item for building of Kiviaty diagram for one experiment

In this window the user defines concrete characteristics for the building of Kiviaty diagrams for one experiment, which he wants to include in the analysis menu of the target problem-oriented simulation system, and gives names to the corresponding menu items.



KIVIATY FOR 1 EXPERIMENT

Input Menu Item name and Choose Positive and Negative criteria

Menu Item Name: Kiviaty diagram for one experiment

● Positive: 1

○ Negative: 0

Type Name	Object Name	Char Name
Facility	eat[1]	KFAC
Lable	eat[2]	NFAC
Process	eat[3]	TFAC
Store	eat[4]	
Table	eat[5]	

Fig.12. A dialog of arrangement for menu item of building of Kiviaty diagram for one experiment

4.3.3.4. "Experiments Kiviaty diagram" subitem — dialog of arrangement of menu item of building of Kiviaty diagram for series of experiments.

This dialog has the same type as one on Fig.12. The difference consists in the fact that for each item of created menu only one positive and one negative criterion should be chosen.

Let us consider now how choice made by the user in the dialogs of **SimW** described above is mirrored in the target problem-oriented simulation system. Item names of the analysis menu represent by themselves the descriptions in terms of application domain which user inputs in edit window of each dialog. At the choice of some menu item in child window a particular plot or diagram will appear.

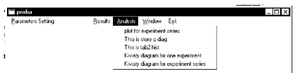

proba
Parameters Setting
Results
Analysis
Window
Exit
plot for experiment series
Thia is store q diag
This is tab2 hist
Kiviaty diagram for one experiment
Kiviaty diagram for experiment series

Fig.13 Analysis Menu of the target system

4.4 Ready!

At the choice of "Ready!" item of main menu **SimW** creates program modules of the target problem-oriented simulation system and compiles them by standard C++ compiler. The obtained executable module is the target problem-oriented simulation system. It may be executed by any way provided by Win32.

5 Specific implementation features

In this section specific features of modules generated by **SimW** and invisible for the user are described. These are program files from which the target problem-oriented simulation system is created: standard.cpp, standard.h, standard.rh, standard.rc. Module "standard.cpp" is not changeable. It keeps classes implementing architecture of problem-oriented simulation system, i.e. their description, constructors, and the functions uniform for any problem-oriented simulation system. Unique exception is the description of main window class which includes functions called by menu items of the problem-oriented simulation system. the number of such items is defined by the user. The description of main window class, functions of menu items, functions of parameter choice dialog, functions of source setting for graphic analysis and functions of representation of results are included by **SimW** in module standard.h at the design step. All variety entered by a particular user are represented in the resources file "standard.rc". This variety can include names of menu items in terms of application domain, arrangement of dialogs and, in future versions, style of window of results representation. This module is in the same way created by **SimW** at the choice of menu item "Ready!"

6 Conclusions.

We present this software tool as "Simulation System Wizard". The notion of "Wizard of Oz technique"[3] appear originally in Artificial Intelligence. The basic idea of a "Wizard of Oz technique" system is the modeling of a system or system behavior, which is not yet or only partly available, by a human (the hidden "wizard") and to hide this fact from the user. By analyzing the performed operations, the user needs could be identified in advance which may lead to a better design of the target system. Being applied to another branches of Computer Science "Wizard of Oz technique" transforms now to "Wizards" attached to all modern software. But among the community of simulation software this word is not put yet. **SimW** is the software which in further development can replace for an user-practician a number of different kinds of software.

References

[1] Simulation systems integration and problem orientation. Science-technical report of "Simulation languages group", Institute of Mathematics, Academy of Sciences, Moldova Republic, 1995.

[2] Magariu G., Madan V. and Burtseva L., What does the simulation system SOL/PC do? *Comput. Sci. J. of Moldova,* vol.1, No.2 (1993), pp.3–13.

[3] D. Salber and J. Coutaz. Applying the Wizard of Oz Technique to the Study of Multimodal Systems, 1993.

L.Burtseva,
Institute of Mathematics,
Academy of Science of Moldova,
2028, Kishinev, Moldova.
e–mail: 24*luda*@*math.moldova.su*

Received October 7, 1997