

Kozhukhivska O. A.<sup>1</sup>, Fefelov A. O.<sup>2</sup>, Bidyuk P. I.<sup>3</sup>, Kozhukhivskiy A. D.<sup>4</sup><sup>1</sup>Doctor of Philosophy, Senior Lecturer, Cherkassy State Technological University, Ukraine<sup>2</sup>Doctor of Philosophy, Associate Professor, Kherson National Technical University, Ukraine<sup>3</sup>Doctor of Technical Sciences, Professor, National Technical University Of Ukraine «Kyiv Polytechnic Institute», Ukraine<sup>4</sup>Doctor of Technical Sciences, Professor, Cherkassy state technological university, Ukraine, E-mail: andrejkd@mail.ru

## DECISION SUPPORT SYSTEM ARCHITECTURE FOR FORECASTING OF NONSTATIONARY FINANCIAL PROCESSES AND CORRESPONDING RISKS

A concept of a decision support system (DSS) for modeling and forecasting of economic and financial processes is proposed as well as its software implementation. The main functions of the DSS are in modeling and short-term forecasting of nonstationary nonlinear time series in various areas of human activities. On the basis of the proposed concept the software system was developed that possesses a set of the following useful features such as constructing of high quality forecasting mathematical models using time series statistical data; testing automatically quality of models and the forecasts based on them using appropriate sets of statistical criteria; combining various computational procedures for improving forecasts estimates; easy modification and expanding of existing system functionality. The software has been created with the use of cross platform instrumentation Qt and programming languages C++ and QML. Thanks to the open system architecture the functional possibilities of the system created can be expanded substantially at the expense of new model types, extra computational techniques and analytical instruments, possibilities for graphical representation of results as well as the means for interaction with user.

**Keywords:** model, economic and financial processes, statistical criteria, forecasts estimates risks.

### 1 INTRODUCTION

Development of decision support systems (DSS) for immediate practical applications is a popular research and engineering direction in many areas of human activities. Their application is widely spread thanks to the relatively simple system architecture and wide functional possibilities directed towards search of the best solutions for a particular problem. The most popular applications of DSS are modeling and forecasting problems, logistic problems solving, decision support in social, economic and financial areas, development of diagnostic tools and many others [1–3]. It was shown that DSS are especially useful in cases when the volume of computations necessary for forming alternative decisions is substantial [4, 5]. Very often decisions are based on short- and medium term forecasts what requires preliminary data processing techniques, forecasting model structure and parameter estimation, computing of forecasts using special forecasting functions, quality analysis for computations at every step of data processing. Besides, very often it is necessary to apply optimization procedures to find optimal solutions for many specific decision making problems [6]. Thus, the volume of computations may reach very high levels, and it becomes impossible to make quality decisions without appropriately designed DSS.

The system proposed here has been developed as most of modern applications and consists of two basic components: executable and instrumental. The executable component is characterized by a set of functions that allow the DSS to operate as an integrated unit. The basic system functions are as follows: (1) – knowledge and database forming for storing system data and knowledge (DKB), user's and tuning information; (2) – input of data into the system from possible local and net means; (3) – preliminary data processing aiming to data transforming to the form acceptable for parameter estimation and other computations; (4) – synthesis of forecasting models on the basis of time series (TS) data; (5) – testing of forecasting models in real time and in the simulation mode; (6) – reports forming on the results of time series models testing; (7) – current system state visualization as well as intermediate computational results together with the input data used; (8) – forms for the intermediate and final results visualization; (9) – tuning of system parameters and its separate modules. The instrumental component is represented by a set of means that allow a user to manipulate with computational techniques so that to create various combinations or hybrids of forecasting models potentially characterized by enhanced quality of forecasting. The instruments developed differ with their functional assignment, and their set can be expanded with the newly developed modules and integrated into the system.

## 2 PROBLEM FORMULATION

The purpose of the study is as follows: 1) analysis and development of requirements to the forecasting decision support system; 2) development of the system architecture; 3) development of functional layout of the system; 4) selection of mathematical modeling and forecasting techniques for financial and economic processes; 5) illustration of the system application to solving the problem of financial and economic processes forecasting with statistical data.

## 3 GENERAL REQUIREMENTS TO THE FORECASTING DECISION SUPPORT SYSTEM

Modern DSS should satisfy a set of requirements that make them especially suitable and useful for solving particular problems. These requirements include the following basic elements: – user adaptive and friendly interface allowing for prolonged exploitation without substantial strain of a user; – expanding data and knowledge base containing all necessary data, computational procedures, preliminary data processing tools, quality control criteria to be used at each step of data processing, constants, rules etc; – the central data processing subsystem should be highly flexible so that to select automatically and use appropriate high quality computational procedures at each step of decision making (alternatives generation); – means for communication with similar systems and databases in the frames of local and external computer nets; – to provide user with appropriate multiple representation forms for intermediate and final results of computing; – to provide a possibility for easy modifying and expanding the system functionality.

DSS interface should be highly flexible in the sense of adaptation to specific user (say at different levels of company management) and designed according to the modern requirements of human factors and ergonomics. Interface is the most important element of DSS from the point of view of its presentation for a potential user. A user may not be familiar with the modern optimization techniques used for alternatives generation but he understands well how convenient is the system interface and is it tiresome or not to work with him for many hours running. When selecting or developing new computational procedures it is very important to provide necessary quality of computing results. Usually acceptable errors of computing are formulated and registered on the stage of general problem statement regarding system design.

DSS development should be based on the system analysis principles that suppose the following: – hierarchical system design that corresponds to the hierarchy of decision making process by a user; – identification of possible sources of uncertainties that should be taken into consideration while model building, forecasts computing, and generation of decision alternatives; – hiring of optimization procedures whenever it is appropriate and possible; – development and implementation of adaptive computing procedures providing for appropriate model

correction with new data coming; – construction of integrated (combined) models and procedures aiming to quality improvement of the final result; – automatic control of all computational steps of data processing, model building and forecasts generation with appropriate statistical quality criteria; – functional completeness of a system providing availability of all necessary system functions for interaction, communication, computing, results representation etc.

## 4 MATHEMATICAL MODELS AND PARAMETER ESTIMATION TECHNIQUES FOR DSS

Very often appropriate fit to financial and economic processes provide the following types of models: autoregression  $AR(p)$ , autoregression with moving average  $ARMA(p,q)$ , autoregression with integrated moving average  $ARIMA(p,d,q)$ , multiple regression, nonlinear logit and probit models, structured equation models (that take into account in explicit way actual effects available in time series), polynomial models created by the group method of data handling (GMDH), vector autoregression, regression on support vectors, and some other types [7, 8]. An important problem is selection of model parameters estimation techniques. To provide a necessary choice for estimation of the variety models mentioned above the following parameter estimation techniques are used in the DSS: ordinary least squares (OLS), generalized least squares (GLS) and recursive least squares (RLS), maximum likelihood procedures (for linear and some nonlinear models), nonlinear least squares (NLS), two-stage LS, and Markov chain for Monte Carlo (MCMC). The last one represents a universal technique based upon simulation procedures that require appropriately generated sequences of pseudorandom numbers. The most known procedures of this class are: Metropolis algorithm, Metropolis-Hastings and others. An advantage of this approach is in its universality regarding the model types being estimated, though sometimes they are very time consuming and may not be suitable for application in real time. The optimization procedures hired are based upon genetic algorithms and artificial immune system ideas though some of them are classic of Newton type. Parameter estimates computed are tested for statistical significance using  $t$  – statistic when it is appropriate.

## 5 CONCEPTUAL SYSTEM MODEL

The system concept is based on the modern module architecture with the possibility of effective replacement of functional blocks without changing of an overall system structure. Following the modern approaches to software system development the separate functional blocks are integrated into subsystems using the functional similarity patterns (Fig. 1).

As of today we can define DSS as integrating the following five subsystems: (1) – data and knowledge base (DKB); (2) – computing subsystem; (3) – communication subsystem that provides communication with various devices; (4) – interaction subsystem; (5) – application instrumentation subsystem. The first four subsystems are

forming an executable component for application. The fifth subsystem integrates a set of algorithms that implement specific computational procedures related to the problem being solved. A short description of the subsystems and its basic modules is given below. The DKB subsystem contains all necessary for effective functioning of DSS knowledge and data resources. We can define the following basic elements forming the subsystem: reading and writing data formats for input of information into the system and output of intermediate and final results; quality estimation criteria for mathematical models and forecasts estimates; time series data; tuning data for the system and its separate elements; intermediate and final results of computing etc.

The computational subsystem implements a set of basic techniques for preliminary input data processing (preliminary data processing block). Each of the technique requires additional parameters for tuning. There are exists a possibility for a sequential application of several data processing methods for the same data what provides implementation of an effective multistep preliminary processing procedures.

The model synthesis block uses specific model parameter learning procedures to perform a model synthesis using the learning data from DKB. The subsystem is organized in a way that its functioning does not depend on a type of the problem

being currently solved. Such approach provides a possibility for compatibility of the system with arbitrary computational procedure taken from the instrumental subsystem. The purpose of the model testing subsystem is to estimate TS model quality using test sample (in off-line mode) or actual data in real time. A set of standard test statistics is used for the purpose, more specifically: determination coefficient ( $R^2$ ), root mean squared error (RMSE), Durbin-Watson statistic (DW), Information Akaike criterion (AIC), mean error, mean absolute percentage error (MAPE), Theil coefficient etc. Besides, there is a possibility for loading and testing previously constructed models (retrospective analysis) that are stored in local DKB or in external memory. As a matter of fact the model testing block performs the basic goal function for the whole software system, i.e. it implements the forecasting function. This is the main element of the basic application (system) cycle (Fig. 2).

It performs continuous monitoring of forecasts deviation from observed TS values, and, dependently on the deviation amplitude, a decision is made on activation of the synthesis block and correcting of currently used model. The report generation function can be activated during the testing procedure or during model synthesis (learning). During the testing procedure the report reflects, as a rule, the learning

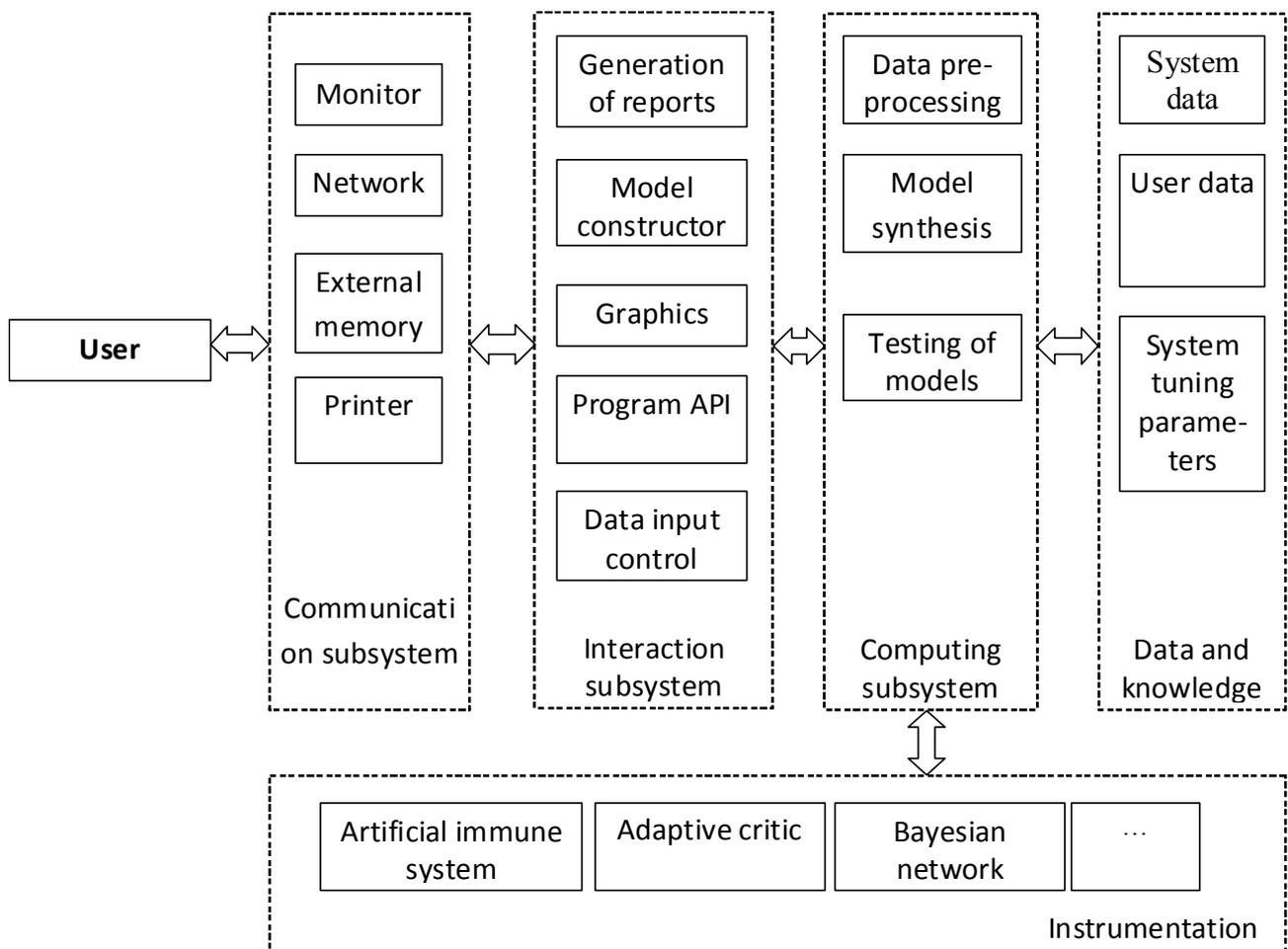


Fig. 1. General conceptual layout of the software system (system architecture)

statistic (dynamics of minimum and mean errors, current generation and mean affinity of a population in a case of the use of artificial immune system). The report can be given in a text or graphical form.

The graphical possibilities of the system are implemented with a set of classes of special graphical primitives representing the elements of mathematical graphs, diagrams, histograms etc. Using declarative features of the QML language these primitives allow to construct complex

compositions and to expand this way the possibilities for graphical data representation. Fig. 3 shows a hierarchy of classes for primitives, and Fig. 4 illustrates the process of a graph integration in QML.

Most of the classes for the hierarchy under consideration were implemented in C++ programming language and physically incorporated into graphical kernel of the system. In principle the functional possibilities of these classes could be reproduced using JavaScript language, which is a «seamless» supplement

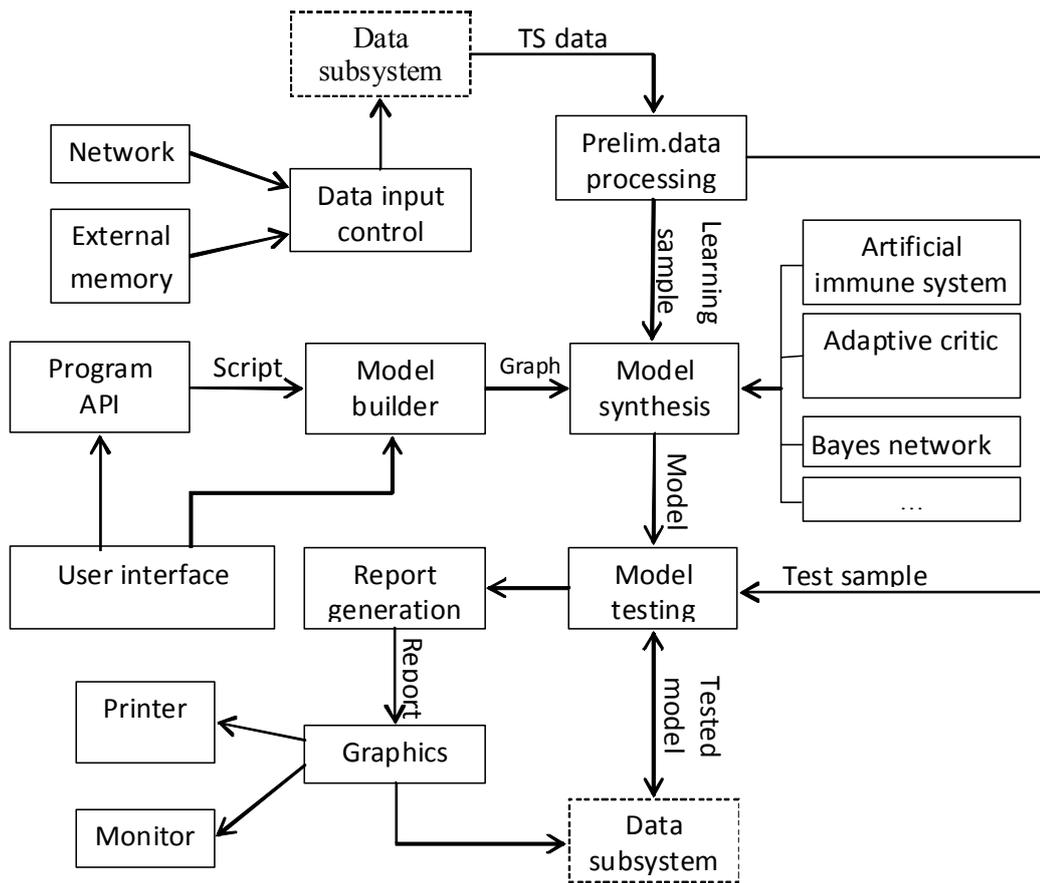


Fig. 2. Functional layout of the software system

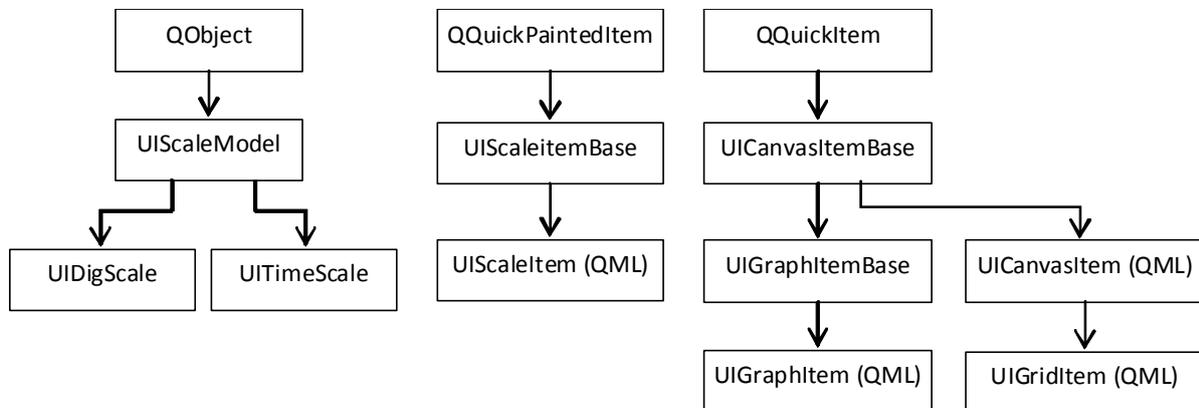


Fig. 3. Hierarchy of classes for graphical primitives of the system developed

```

Item{
  UIGridItem
  {
    // Interactive graphical grid of scale
  }
  UIScaleItem
  {
    // Graphical component of a vertical scale (of
    // values)
    id: verItem
    scaleModel: floatScale//Reference to the scale
    model
  }

  UIScaleModel
  {
    // Model of scale for the floating point values
    id: floatScale
  }

  UITimeScale
  {
    // Model of scale for the time values
    id: timeScale// Reference to the scale model
  }
  }
  UIScaleItem
  {
    // Graphical component of horizontal scale (time)
    id: horItem
    scaleModel: timeScale
  }
}
    
```

Fig. 4. QML-document, implementing construction of a complex graphical component

for QML, but JavaScript is less effective from the point of view of computer graphics implementation because of lower productivity in comparison to C++.

The constructing of a computational instrument for the system is performed in analogous way. The system kernel contains for this purpose the sets of C++ classes that have their reflection in QML. For example, an artificial immune system could be «constructed» with «bricks» with pointing out to the source of data and its format, and the method for computing the goal function. On the other side such a hybrid could be constructed using some other computational procedure. Such a task could be performed with the use of a programmed interface of an application, as well as with the use of graphical interface. An example of hierarchy for classes for artificial immune system is given in Fig. 5.

Ais is the class of artificial immune system instrument (it incorporates the basic algorithm of functioning for this instrument);

Domain is a basic class for the methods of estimation of population individuals for artificial immune system (specific estimation techniques are for the moment the classes of wavelet-neural network, network of feed-forward propagation, and genetic system programming);

CfgParser is a class of configuration tuning for an instrument;

TimeDataStream is a basic class of a data source in the form of time series (the data sources are identified as local and the net sources).

Fig. 6 demonstrates the process of constructing of hybrid artificial immune system and wavelet-neural network in QML notation. An example of work with the instrument of artificial immune system is demonstrated in Fig. 7. For providing a normal functioning of the system developed it is necessary

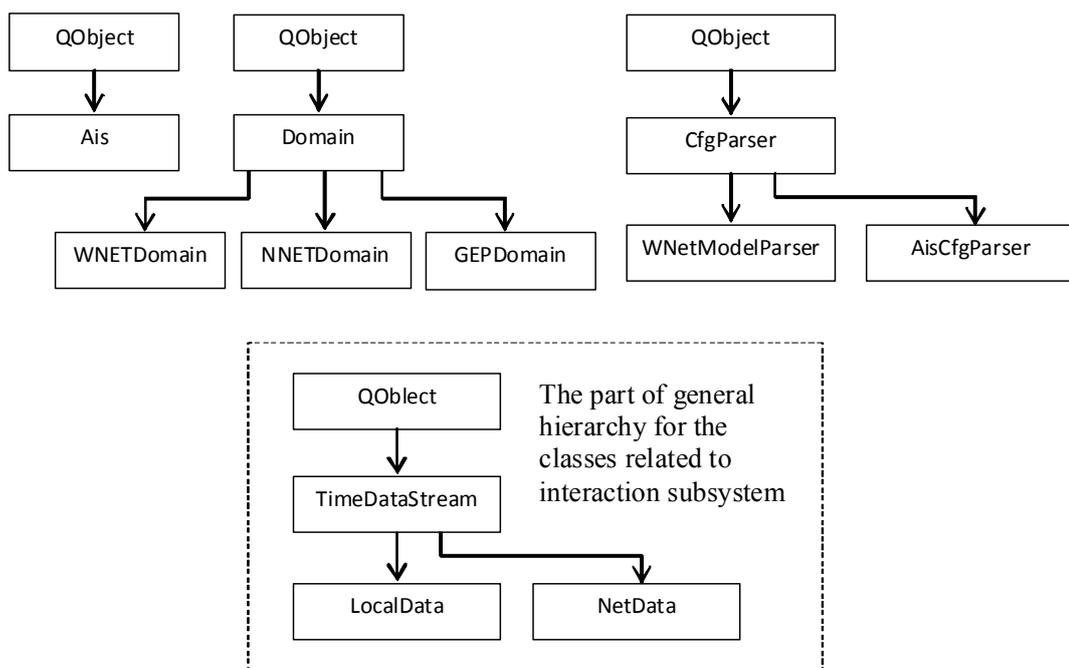


Fig. 5. Classes hierarchy for of the instrument of artificial immune system

to have personal computer and some other technical means with the following hardware characteristics: CPU AMD Phenom IIX2 with the clock frequency of 3 MHz or higher or Intel Pentium with analogous productivity; 2 Gb or more of RAM; video card with the OpenGL support; 100 Mb of hard

disk space for the system files, and 500 Gb for database (the last value is optional, it depends on the specific task performed); operational system Microsoft Windows XP/ Vista/7/8 or Linux; monitor with a definition of 1024x768 pixels or higher (it is recommended to use wide format monitor: 16:9 or 16:10). Besides, appropriate net adapter may be needed in a case of using external data sources or other devices that provide compatibility with the system.

The DSS developed was implemented using cross platform instrumentation Qt, and the basic programming language C++. The software developed with this instrumentation is compatible with the most modern operation systems after program compilation under selected OS without changing the source code.

One of the best features of Qt is the possibility of using the declarative tuning of programming language QML that is oriented (directly) to the user interface development. The modules written in C++ with the use of simple declarations could reflect their features and methods into QML documents. This feature provides a possibility for an effective organization of interaction between the program kernel and the graphic user interface. The declarative approach to constructing the user interface is, no doubt, more convenient in comparison to the classic imperative approach that is used in the most modern programming languages of general purpose including C++.

```

LocalData{
// Setting local data source
id: localData
}

Ais{
id: ais
cfgFilename: "ais.cfg"

// Setting the method of an individual
estimation for immune system
// In fact this is hybrid of immune
system and wavelet-neural network
domain: WNETDomain{
id: wnetDomain
dataStream: localData// Switching in the
class of a local data source
    
```

Fig. 6. QML-document implementing the construction of a hybrid artificial immune system and wavelet-neural network

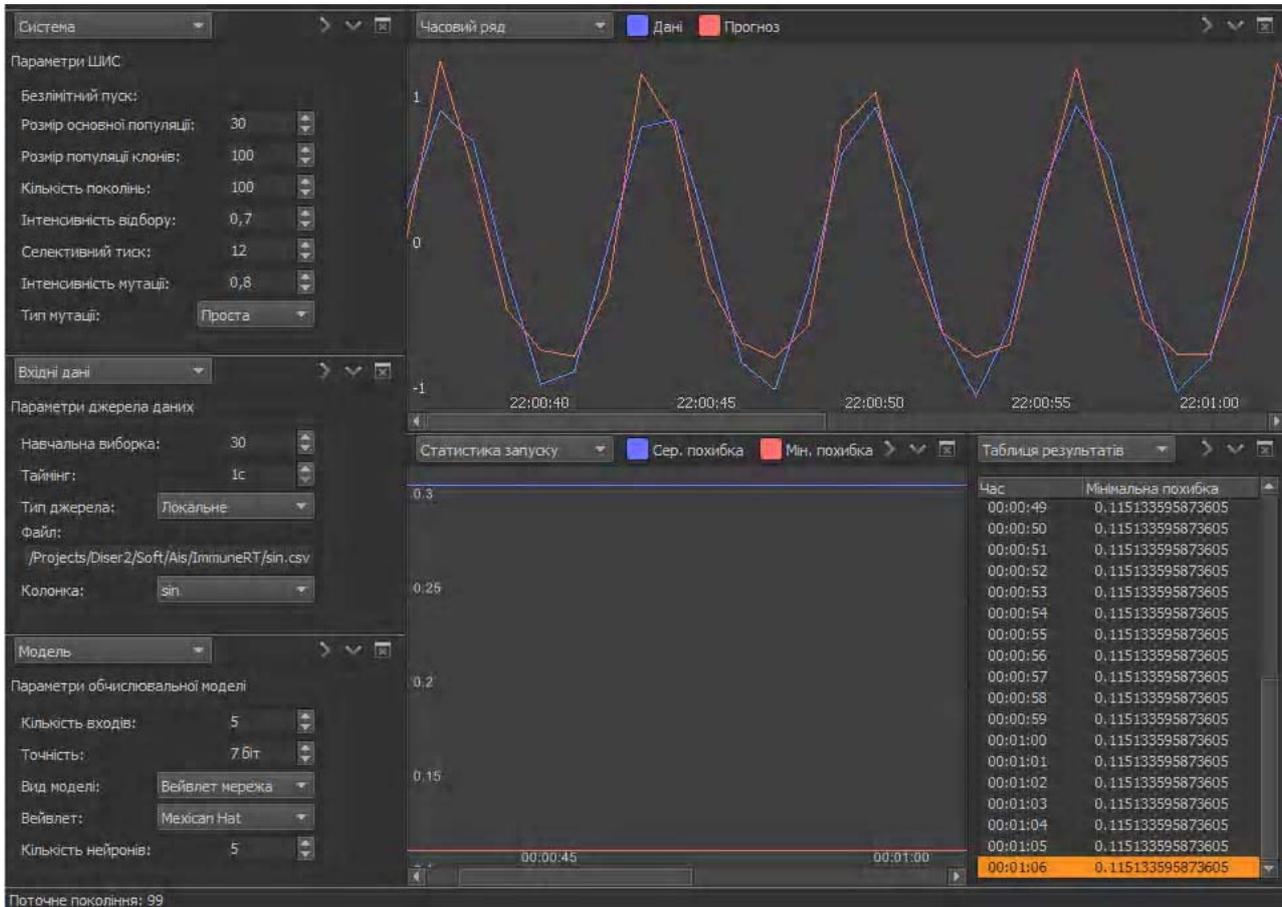


Fig. 7. The tuning parameters and result of functioning of the artificial immune system instrument

## 6 CONCLUSIONS

General requirements to the decision support system to be used for modeling and forecasting financial and economic processes have been worked out, and mathematical models for description of the processes selected. A concept of the decision support system was proposed as well as its software implementation. The main purpose of the DSS is modeling and short-term forecasting of nonstationary nonlinear time series, and the use of forecasts for estimation of some financial risks. On the basis of the concept proposed software system was developed that possesses a set of the following useful features: – it allows to construct high quality forecasting mathematical models using time series statistical data; – to test automatically quality of models and the forecasts based on them using appropriate sets of statistical criteria; – to add and combine various computational procedures for improving forecasts estimates; – to work with local and distributed databases; – to easily modify and expand existing system functionality.

The software has been created with the use of the cross platform instrumentation Qt and programming languages C++ and QML. Thanks to the open system architecture the functional possibilities of the system created can be expanded substantially (to desirable level) at the expense of extra computational techniques and analytical instruments, possibilities for graphical representation of intermediate and final results as well as the means for interaction with a user.

In the future studies it is supposed to incorporate into the DSS new models of static and dynamic Bayesian

networks types. Also a substantial attention should be paid to hiring the methods for appropriate taking into consideration of possible data and process uncertainties. New integrated data, models, and forecasts estimates criteria should be developed and integrated into the system to simplify the process of selection of the best modeling and forecasting results.

## REFERENCES

1. *Hollsapple, C. W.* Decision support systems / C. W. Hollsapple, A. B. Winston. – Saint Paul : West Publishing Company, 1996. – 860 p.
2. *Burstein, F.* Handbook of Decision Support Systems / F. Burstein, C. W. Holsapple. – Berlin : Springer-Verlag, 2008. – 908 p.
3. *Sytnyk, V. F.* Decision support systems / V. F. Sytnyk. – Kyiv : KNEU, 2004. – 614 p.
4. *Polovcev, O. V.* System approach to modeling, forecasting, and control of financial and economic processes / O. V. Polovcev, P. I. Bidyuk, L. O. Korshevnyuk. – Donetsk : Oriental Publishing House, 2009. – 286 p.
5. *Bidyuk, P. I.* Computer based decision support systems / P. I. Bidyuk, O. P. Gozhij, L. O. Korshevnyuk. – Mykolaiv : Chornomorsky State University, 2012. – 380 p.
6. *Gupal, A. M.* Optimal recognition procedures / A. M. Gupal, I. V. Sergiyenko. – Kyiv: Naukova Dumka, 2008. – 232 p.
7. *Tsay, R. S.* Analysis of financial time series / R. S. Tsay. – Hoboken : Wiley & Sons, Inc., 2010. – 715 p.
8. *Bidyuk, P. I.* Methods of Forecasting / P. I. Bidyuk, O. S. Menyailenko, O. V. Polovcev. – Lugansk : Alma Mater, 2008. – 608 p.

Стаття надійшла до редакції 05.04.2014.

Кожуховская О. А.<sup>1</sup>, Фефелов А. А.<sup>2</sup>, Бидюк П. И.<sup>3</sup>, Кожуховский А. Д.<sup>4</sup>

<sup>1</sup>Канд. техн. наук, ст. преп., Черкасский государственный технологический университет, Украина

<sup>2</sup>Канд. техн. наук, доцент, Херсонский национальный технический университет, Украина

<sup>3</sup>Д-р техн. наук, профессор, Национальный технический университет Украины «КПИ», Украина

<sup>4</sup>Д-р техн. наук, профессор, Черкасский государственный технологический университет, Украина

## АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СППР ПРОГНОЗИРОВАНИЯ НЕСТАЦИОНАРНЫХ ФИНАНСОВЫХ ПРОЦЕССОВ И СОПУТСТВУЮЩИХ РИСКОВ

В работе рассматривается концепция построения системы поддержки принятия решений для моделирования и прогнозирования экономических и финансовых процессов, а также ее программная реализация. На основе предложенной концепции разработана программная система, которая имеет следующие положительные характеристики: построение высококачественных математических моделей на основе статистических данных в форме временных рядов; автоматическое тестирование качества моделей и оценок прогнозов, полученных на их основе, с помощью соответствующих множеств статистических критериев качества; комбинирование оценок прогнозов, полученных различными методами, с целью повышения качества комбинированного прогноза; простота модификации и расширения функций созданной системы.

**Ключевые слова:** модель, экономические и финансовые процессы, статистические критерии, прогнозирование оценок рисков.

Кожухівська О. А.<sup>1</sup>, Фефелов А. О.<sup>2</sup>, Бідюк П. І.<sup>3</sup>, Кожухівський А. Д.<sup>4</sup>

<sup>1</sup>Канд. техн. наук, ст. викл., Черкаський державний технологічний університет, Україна

<sup>2</sup>Канд. техн. наук, доцент, Херсонський національний технічний університет, Україна

<sup>3</sup>Д-р техн. наук, професор, Національний технічний університет України «КПІ», Україна

<sup>4</sup>Д-р техн. наук, професор, Черкаський державний технологічний університет, Україна, Е-mail: andrejdc@mail.ru

## АРХИТЕКТУРА ІНФОРМАЦІЙНОЇ СППР ПРОГНОЗУВАННЯ НЕСТАЦІОНАРНИХ ФІНАНСОВИХ ПРОЦЕСІВ І СУПУТНИХ РИЗИКІВ

В роботі розглядається концепція побудови системи підтримки прийняття рішень для моделювання і прогнозування економічних і фінансових процесів, а також її програмна реалізація. На основі запропонованої концепції розроблена програмна система, яка має такі позитивні характеристики: побудова високоякісних прогнозуючих моделей на основі статистичних даних у формі часових рядів; автоматичне тестування якості моделей і оцінок прогнозів, отриманих на їх основі, за допомогою відповідних

множин статистичних критеріїв якості; комбінування оцінок прогнозів, обчислених різними методами, з метою підвищення якості комбінованого прогнозу; простота модифікації та розширення функцій створеної системи.

**Ключові слова:** модель, економічні і фінансові процеси, статистичні критерії, прогнозування оцінок ризиків.

#### REFERENCES

1. Holsapple C. W., Winston A. B. Decision support systems. Saint Paul, West Publishing Company, 1996, 860 p.
2. Burstein F., Holsapple C. W. Handbook of Decision Support Systems. Berlin, Springer-Verlag, 2008, 908 p.
3. Sytnyk V. F. Decision support systems. Kyiv, KNEU, 2004, 614 p.
4. Polovcev O. V., Bidyuk P. I., Korshevnyuk L. O. System approach to modeling, forecasting, and control of financial and economic processes. Donetsk, Oriental Publishing House, 2009, 286 p.
5. Bidyuk P. I., Gozhij O. P., Korshevnyuk L. O. Computer based decision support systems. Chornomorsky State University, 2012, 380 p.
6. Gupal A. M., Sergiyenko I. V. Optimal recognition procedures. Kyiv, Naukova Dumka, 2008, 232 p.
7. Tsay R. S. Analysis of financial time series. Hoboken, Wiley & Sons, Inc., 2010, 715 p.
8. Bidyuk P. I., Menyailenko O. S., Polovcev O. V. Methods of Forecasting. Lugansk, Alma Mater, 2008, 608 p.