Examples of Integrating Wolfram Mathematica Tools into OSTIS Applications

Valery B. Taranchuk
Department of Computer Applications and Systems
Belarusian State University
Minsk, Republic of Belarus
taranchuk@bsu.by

Vladislav A. Savionok
Department of Software for Information Technologies
Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
v.savenok@bsuir.by

***Abstract*—Within the concept of convergence and unification of intelligent computer systems of the new generation, technical solutions are discussed, examples of development and modernization, integration of Ecosystem OSTIS tools with Wolfram Mathematica (WM) computer algebra system (CAS) are provided.**

On the example of integration with specialized complex of intellectual educational resource for the discipline "Computer Systems and Networks" the possibilities of using WM tools in ostis-system are discussed when solving problems related, in particular, to topology of info-communication networks. The application of WM tools for visualization of network topology, as well as emulation of the search for the optimal route for data transmission is shown.

***Keywords*—OSTIS technology, ostis-system, computer algebra systems, Wolfram Mathematica, Computer Systems and Networks, graph visualization**

I. INTRODUCTION

Following the assessment of the current state of work in the field of Artificial Intelligence (AI), it is possible to affirm active local development of various directions (non-classical logics, formal ontologies, artificial neural networks, machine learning, soft computing, multi-agent systems, etc.), however, a comprehensive increase in the intelligence of modern intelligent computer systems does not occur [1].

The key reasons of methodological problems of current state of Artificial Intelligence, as well as a number of actions required to solve them are outlined in [1]. What actions are needed to improve the current state? First of all, it is necessary to converge and integrate all directions of Artificial Intelligence and corresponding construction of a general formal theory of intelligent computer systems (ICS), the transformation of modern variety of frameworks for development of different ICS components into a single technology of complex design and support of the full life cycle of these systems, which guarantees the compatibility of all developed components, as well as the compatibility of the ICS as independent subjects, interacting between each other. Convergence and unification of new generation of intellectual computer systems and their components is required.

Convergence and unification of new generation intelligent computer systems and their components is necessary. At the same time, convergent solutions basically mean optimized complexes that include everything necessary to solve AI tasks, organized and configured for efficient use of information resources, simplification of implementation processes, meeting the requirements of maximum performance, availability of intelligent interface, simple and understandable for all categories of users.

Supporting the outlined concepts, we note that such problems can be effectively solved by developing, improving, regularly updating the content of intelligent systems by incorporating CAS means. Below are a few methodological and technical solutions for integration of different types of knowledge, implemented by inclusion of Wolfram Mathematica functions in the ostis-system of support and maintenance of the teaching process of one of the basic disciplines in high school, illustrated by examples.

II. CONCRETIZATION AND VARIANTS FOR INTEGRATING THE ECOSYSTEM OSTIS WITH CAS

Integration of the Ecosystem OSTIS with any service means the ability to use the functionality of the service to change the internal state of the system's knowledge base. Within Ecosystem OSTIS full and partial integration levels are acceptable.

According to the Technology, full integration of the Ecosystem OSTIS with any service implies the possibility of executing service's function at platform-independent level using SCP language. That is, the task of integration of such a service is reduced to allocation of a graph structure processing algorithm and its implementation within a system's knowledge base. As a result of such integration there is no need to use a third-party service, in fact, an Ecosystem component is used.

Partial integration means changing the state of the system's knowledge base at the stages of service function execution. The depth of integration can vary. In some cases, a service can refer to the knowledge base to get additional information or to record intermediate results. In the simplest case, a knowledge base can change only once, after a result of the service's function is received. In case of partial integration it is supposed that particular ostis-systems are to play the role of system integrators of included resources and services of other computer systems, as the level of intelligence of ostis-systems allows them to specify the computer systems being integrated to a sufficient degree of detail and, consequently, to "understand" adequately what each of them knows and/or can do.

Separately, let us note that following the Technology, ostis-systems are capable (and it should be used) to coordinate the activities of a third-party resource and service sufficiently well, to provide a "relevant" search for the required component. The systems themselves can also perform the role of intelligent help-systems – assistants and consultants for efficient operations with functional capabilities, when the user interface is implemented with non-trivial semantics in the unique tasks of complex subject areas. Such help systems can be made intelligent intermediaries between the relevant computer systems and their users.

The systems themselves can also act as intelligent help systems. Their relevance is dictated by the high complexity of subject areas and the non-triviality of some unique tasks. Such conditions require the design of appropriate unique and nontrivial user interfaces with additional information support for their use. Such help systems can be made intelligent intermediaries between the relevant computer systems and their users, and the homogeneity of the technologies used ensures seamless integration with the existing system.

Solving the issues of data format coordination. A tedious problem of functional service integration when forming a digital ecosystem of multiple interacting services is the difference in data formats that participants of this Ecosystem work with. Two services, which imply data processing from one subject area, are likely to have different data formats. The problem of coordinating the data format of different services significantly complicates the development of the services themselves and leads to an increase in time costs. Such issues can be effectively solved using CAS import and export functions, such as in Wolfram Mathematica, which supports more than 100 data formats, including graphics, video, and more.

At the current stage of development and usage of the Ecosystem OSTIS, one of the priority areas appears to be the integration of the capabilities of computer algebra systems and intelligent learning systems constructed in ostis-application. The importance of this is due to the relevance, the requirements of the intellectualization of educational resources on the one hand, and on the other – contents of computer algebra systems, which have an undoubted advantage and great opportunities for solving problems relevant to educational systems for virtually all natural-science and technical disciplines, involving the use of complex mathematical apparatus.

It can be stated that, despite the popularity of topics related to automation and intellectualization of educational activities in science disciplines and the development of appropriate computer systems, at the moment the market is practically lacking tested intellectual educational systems capable of self generating and solving various problems, and verifying the correctness of the solution provided by the user. As prototypes, there are some systems that consider non-trivial problems, such as geometry [2], [3] and graph theory [4]. But, to be fair, it should be noted that there is no intelligence in the mentioned systems (in fact, only a specific set of actions is implemented, the tasks are not generated in the applications themselves), there is no means of verification of solutions with even minor deviations of the design rules.

One of the variants for interaction between Ecosystem OSTIS and CAS can be approaches similar to the integration of artificial neural networks in ostis-systems (see [5]).

Developing the aforementioned implementations, the following methodological and technical solutions can be considered:

- Black-box integration, when the knowledge base of the ostis-system contains the specification of the used kernel function of the computer algebra system, as well as the specification of the method of calling this function (for example, specifying through which software interface the interaction with this external system is performed). This integration variant is the easiest to implement and generally has the advantages listed below. At the same time, this variant has a disadvantage that the ostis-system does not contain means of analysis and explanation of how a certain step of solving a problem that is realized by a used CAS function was taken.
- A tighter integration, in which a particular function is still a part of a third-party CAS, when not only the result of its performance is loaded to the knowledge base of ostisystem, but also all possible specification of it, e.g. explanation of the problem solution step, indication of particular algorithms and formulas which can be involved in the solution, description of possible alternative solution variants, evaluation of solution efficiency and so on. In this variant of integration, the ostis-system gets more opportunities of analysis and explanation of the problem solution process. (Note that this doesn't apply specifically to CAS Wolfram Mathematica, because it always has detailed explanations for all solutions, and allows

for step-by-step execution.)

- Full integration, which translates computer algebra functions in use from this system's internal language into the ostis-system. This variant is the most labor-intensive and complicated in terms of updating the capabilities of computer algebra systems in the corresponding ostis-systems taking into account their constant development. At the same time this integration variant, in comparison with the two previous ones, has an important advantage: it ensures platform-independent solution and allows using all the advantages of the approaches proposed within the OSTIS Technology in solving a concrete problem, in particular the possibility of multi-user parallel knowledge processing and the possibility to optimize a problem solution plan or its fragments directly during the solution.

The approach to solving the problems of intellectualization of educational activity, based on the integration of ostis-systems and computer algebra systems, has several advantages:

- When developing ostis-systems, the need to program many functions that have already been implemented and tested in CAS is eliminated. This is fundamental because computer algebra systems are developed by highly qualified specialists in the relevant fields, the implementation of similar functions in ostis-systems may require significant financial and time expenditures.
- A concrete ostis-system using individual functions of CAS, due to the approach to the development of hybrid problem solvers in OSTIS Technology, gets the possibility to *self plan* the course of problem solving provided that some of its steps are implemented by means of the attached functions. From the point of view of the approach proposed within the framework of OSTIS Technology, each function of the computer algebra systems, computer mathematics systems (CMS) becomes a *method* for solving problems of some class. This class of problems is described in the knowledge base of the ostis-system and allows it, when solving a concrete problem, to independently draw a conclusion about the expediency of applying one or another CAS function. Such integration with ostis-systems will make it possible to eliminate a possible disadvantage of individual computer algebra systems noted earlier (determined by which CAS are used – explained below in the overview of computer mathematics systems, conditions of their application and access to individual components).

We emphasize that these integration variants are not mutually exclusive and can be combined. In addition, integration can be deepened step by step taking into account the above advantages and disadvantages as well as the relevance of using certain functions of computer algebra systems in solving specific tasks within the Ecosystem OSTIS and corresponding ostis-systems.

In general case, step-by-step integration of CAS with the Ecosystem OSTIS implies, as a minimum, description of the specification of the basic functions of the selected computer algebra system by means of the OSTIS Technology, in other words – development of the ontology of external functions. In case of Wolfram family systems, the process of developing such ontology can be automated due to the presence of the Wolfram Language formal language and good documentation of system functions.

Summarizing the above, we state: the integration of educational systems developed on the basis of OSTIS Technology and computer algebra systems will allow us to create systems with intelligent properties in a shorter time, and with the use of carefully developed (mathematically, algorithmically) and repeatedly tested tools.

III. FUNDAMENTALS, TERMINOLOGY. COMPUTER MATHEMATICS SYSTEMS, COMPUTER ALGEBRA SYSTEMS

In the mid-twentieth century, at the junction of mathematics and computer science, a fundamental scientific trend, computer algebra, the science of efficient algorithms for calculating mathematical objects, emerged and intensively developed. Synonyms for the term "computer algebra" are: "symbolic calculations", "analytical calculations", "analytical transformations", and sometimes "formal calculations". The field of computer algebra is represented by theory, technology, software tools. Applied results include developed algorithms and software for solving problems using a computer in which the original data and results take the form of mathematical expressions, formulas. The basic product of computer algebra became software computer algebra systems (CAS). The range of mathematical problems solvable with the help of CAS is constantly expanding. Considerable effort is devoted to developing algorithms for computing topological invariants of varieties, nodes, algebraic curves, cohomology of different mathematical objects, and arithmetic invariants of rings of integers in the fields of algebraic numbers. Another direction of modern research is quantum algorithms, which sometimes have polynomial complexity, while existing classical algorithms have exponential complexity.

Research and development of theoretical foundations and technologies for implementing methods and software implementations of computer algebra tools continues. Terms, definitions, names in descriptions of functions and tools of these systems also undergo changes, some formulations earlier given in separate manuals, reviews of tool capabilities are being not only refined, but also changed. This is normal for new scientific directions and technologies. The reader should not be surprised if in other sources they meet different formulations, terms.

A detailed description of the functionality of CAS and CMS can be found in [6], the current state, an overview of the systems on the market can be clarified in [7], a brief outline (basic features) for beginners can be found in [8].

IV. COMPUTER ALGEBRA TOOLS

Regarding the classification of CAS. A fairly complete list with the functionality of symbolic computation systems and the platforms they operate on can be found in [7]. Classification attributes of CAS are: functional purpose, type of architecture, means of implementation, fields of application, integral quality assessments.

A. *Basic functionality of CAS*

CAS allow computer-assisted implementation of analytical and numerical methods for solving problems, presenting the results in mathematical notation, providing graphic visualization, design of the results, and preparation for publication. Using CAS and a computer, it is possible to perform in analytical form the following computations:

- simplifying expressions or reducing to the standard form;
- substituting symbolic and numeric values into expressions;
- extraction of common factors and divisors;
- exponentiation of products and powers, factorization;
- decomposition into simple fractions;
- finding limits of functions and sequences;
- operations with series;
- differentiation in full and partial derivatives;
- finding undetermined and definite integrals;
- continuity analysis of functions;
- finding extremes of functions and their asymptotes;
- operations with vectors;
- matrix operations;
- finding solutions to linear and nonlinear equations;
- symbolic solution of optimization problems;
- algebraic solution of differential equations;
- integral transformations;
- direct and inverse fast Fourier transform;
- interpolation, extrapolation and approximation;
- statistical computations;
- machine theorem proving.

If a problem has a accurate analytical solution, the user of CAS can get this solution in explicit form (of course, we are talking about problems for which the algorithm of solution construction is known).

Also, most of the CAS provide:

- numerical operations of arbitrary precision;
- integer arithmetic for large numbers;
- calculation of fundamental constants with arbitrary precision;
- support for number theory functions;
- editing mathematical expressions in two-dimensional form;
- graphing of analytically defined functions;
- function graphing using table values;
- plotting function graphs in two or three dimensions;
- animation of the plots of various types;
- use of special-purpose extension packages;
- programming in the built-in language;
- automatic formal verification;
- program synthesis.

CAS in the modern implementation are not only applicable for the study of various mathematical and scientific and technical problems using built-in and additional functions, but also contain all the components of programming languages – de facto are problem-oriented high-level programming languages.

Mathematica and Maple are the leaders of CAS – these are powerful systems with their own kernels, equipped with an advanced user interface and with a variety of graphical and editorial capabilities. CAS are also widely used today: Derive, Maxima, Axiom, Reduce, MuPAD. Computer mathematics systems MATLAB, MathCad occupy a special place.

B. *Noncommercial general-purpose CAS*

A distinctive feature of the current state of IT is that commercial software products in many cases can be fully or partially replaced by non-commercial software, analogues with open source – free software. This includes software products which, with or without modification, have no restrictions on use, copying or transferring to other users, whether for a fee or for free. The following is a reference to software released under the GPL.

Maxima. Maxima is a free, full-featured computer algebra system, a descendant of Macsyma, which was developed as part of the Artificial Intelligence Project at the Massachusetts Institute of Technology from 1968 to 1982 (development stages and leaders of the development teams of the main sections listed in [9]). Experts note that Maxima, unlike Mathematica and Maple, is mainly oriented toward applied mathematical calculations. In this connection, the system lacks or reduces sections related to theoretical methods, such as number theory, group theory, algebraic fields, and mathematical logic. At the same time, numbers in mathematical expressions in the system are assumed to be real by default. This allows to get analytical solutions for many computations encountered in applied problems (such as algebraic transformations and simplifications, integration, solution of differential equations), for which solutions do not exist in the complex domain. Maxima itself is a console program; it “draws” all mathematical formulas with regular text characters. This has some advantages. For example, you can use Maxima itself as a kernel to build various graphical special interfaces on top of it. There are several examples to date. We can recommend the following textbook as a basic introduction to Maxima CAS, available in electronic form [10].

Axiom. Axiom is a free computer algebra system [11]. It consists of an interpreter environment, a compiler, and a library describing a strict, mathematically correct type hierarchy. Its development was begun in 1971 by a group of IBM researchers, led by Richard Dimick Jenks. The original name of the system was Scratchpad. Originally the project was seen as a research platform for developing new ideas in computational mathematics. It was sold to the Numerical Algorithms Group (NAG) in the 1990s, named Axiom, and became a commercial product, but was not a commercial success and was withdrawn from the market in October 2001. NAG made Axiom free software and opened the source code under a modified BSD license. Development of the system continues, with releases of new versions [11]. In 2007, Axiom had two open-source forks: OpenAxiom and FriCAS.

OpenAxiom [12] released version 1.4.2 in April 2013. The main changes implemented in this version relate to the work of the compiler. The aforementioned system for preparing and editing documents with GNU TeXmacs mathematical notation can be used as an OpenAxiom interface.

Another branch of Axiom that is being actively developed is FriCAS [13], version 1.3.8 (version 22/06/2022) is currently in use. FriCAS favourably differs from other general-purpose CAS by the developed type hierarchy corresponding to real mathematical structures. Axiom and the named branches are inferior to Maxima at this stage in the pace of development. It is better for beginners to focus on Maxima.

The above information about Maxima, Axiom CAS is specifically given because their codes are open, can be used in the Ecosystem OSTIS.

C. Proprietary CMS, CAS

MATLAB programming system (short for Matrix Laboratory) was developed by The MathWorks, Inc. It is one of the oldest, thoroughly developed and time-tested systems for automating mathematical calculations, built on an extended representation and application of matrix operations. Nowadays the system has gone far beyond specialized matrix and has become one of the most powerful universal integrated CMSs. MATLAB includes tools for developing complex programs with an advanced graphical interface, is an effective environment for conducting research, creating models, solving natural science and engineering problems [14]. The system has become de-facto one of the world standards in the field of modern mathematical and scientific and technical software. First of all, CMS is focused on numerical calculations, with matrix algebra being particularly prominent. The effectiveness of the system is primarily due to its orientation to work with multidimensional arrays, large and sparse matrices with software emulation of parallel calculations and simplified tools for setting cycles. Recent versions of the system support 64-bit microprocessors and multicore microprocessors such as Intel Core 2 Duo and Quad. System functionality is provided by a rich command library and its own programming language. MATLAB is the largest of all PC-oriented systems due to large number of extension packages it comes with. Its file size exceeds 3 GB. MATLAB works on most modern operating systems, including Linux, macOS, Solaris (support for Solaris is discontinued since R2010b) and Windows. There are many publications describing the system and its components – in Russian [15] can be noted.

Versions history of MATLAB can be traced in [16]. Focusing only on notable items in terms of AI, machine learning, and data mining for versions after 2012 (code R2012* means the 2012 version):

- MATLAB 8.2 R2013b – added table data type, Java runtime updated to version 7;
- MATLAB 8.4 R2014b – added improved user toolbar, new functions and packages, such as py (to use Python), web page counter, histograms, TCP client, and others;
- MATLAB 8.6 R2015b – new runtime mechanism (LXE) and new classes, such as graphs and orgraphs, have been added to handle graphs;
- MATLAB 9.1 R2016b – official MATLAB engine for JAVA, new encoding and decoding functions for JSON, new "string" data type added; algorithms for handling non-memory data, including algorithms for dimensionality reduction, descriptive statistics, k-means clustering, linear regression, logistic regression and discriminant analysis; Bayesian optimization to automatically adjust machine learning algorithm parameters, component neighborhood analysis (NCA) to select machine learning model functions;
- MATLAB 9.5 R2018b – implemented graph axis interaction, which provides efficient data analysis with panning, zooming; added functions: removing outliers in an array, table or schedule; setting a local environment about each element in the input data;
- MATLAB 9.6 R2019a – Added Functions to specify the location of a missing value, detect outliers using percentiles; Implemented improvements for artificial intelligence and analytics;
- MATLAB 9.7 R2019b – includes updates on artificial intelligence (new features allow users to train advanced network architectures using custom learning cycles, automatic differentiation, common weights, and custom loss functions; users can create generative adversarial GANs, Siamese networks,

variational autocoders, and attention networks; Deep Learning Toolbox can also now export to ONNX format networks that combine CNN and LSTM layers, and networks that include 3D CNN layers);

- MATLAB 9.11 R2021b – added: a set of tools for statistics and machine learning (signal and image analysis, preprocessing and parameter extraction using wavelet methods and interactive applications for artificial intelligence models), k-means clustering in real problems;
- MATLAB 9.13 R2022b includes updates on artificial intelligence, a set of system identification tools – create nonlinear state-space models based on deep learning using neural ordinary differential equations (ODEs); machine learning and deep learning techniques can also represent nonlinear dynamics in nonlinear ARX and Hammerstein-Wiener models.

MATLAB is a commercial system; there are non-commercial versions of its type that are compatible in basic language constructs, but not compatible in library functions. For example, Scilab, Maxima, Euler Math Toolbox and Octave.

MATLAB includes a command interpreter, graphical shell, editor-debugger, profiler, compiler, symbolic kernel Maple for analytical calculations, mathematical libraries and Toolboxes libraries, designed to work with special classes of tasks.

MATLAB language. MATLAB system is both an operating environment and a programming language. MATLAB language is a high-level interpreted programming language. Programs written in MATLAB are of two types: functions and scripts. MATLAB programs, both console and GUI, can be compiled using MATLAB Compiler component into independent executable applications or dynamic libraries. MATLAB Builder programs extend the capabilities of MATLAB Compiler and allow you to create independent Java, .NET or Excel components.

Basic extension packages. A feature of MATLAB CMS is the ability to create special toolboxes. MathWorks supplies more than 80 toolboxes that are used in many areas. In recent releases, the company classifies them into three families – MATLAB, SIMULINK and Polyspace [14], as well as partner products.

Maple. Mathematica and Maple are the leaders of CAS, and they are often compared. This seems counterproductive. Each of the systems named has its own characteristics, and they have their own strengths and weaknesses; constantly competing with each other, they are evolving and improving. Most CAS users have experienced several other systems before choosing their primary system. The exchange of opinions, the analysis of publications, and presentations at conferences and seminars allow us to state that each system has its own adherents, and it is useless to convince specialists who have been using CAS for quite a long time that a system other than the one they prefer is somehow better than the others. In most cases, the main factor in using a particular CAS is the user's habit. However, many note that having mastered any of the systems, it is easy to work with others.

In terms of completeness and interface solutions, Mathematica and Maple tools for symbolic and numerical computation are perfect, not the lack of any functions or tools, but the skills of the users. It is impossible to give a complete overview of the capabilities of Maple, as well as Mathematica. It is unlikely that any of the authors of even specialized publications with a book focus on a particular class of problems can present all of the tools of the named CAS from the spectrum they cover. This material can only be regarded as an introduction to the capabilities of the system, mentioning classes of problems of interest to undergraduates, graduate students, postgraduates, researchers, and programmers. Again, the functionality of Mathematica and Maple in almost everything related to mathematics, applied mathematics, and computer science is not only sufficient, but also redundant. Since the main list of features of CAS has already been given above, and in Maple they are implemented, here we will note what in a number of sources is either omitted or called by other terms.

There are many books devoted to the Maple system around the world, a list of which can be found on the developer's section [17]. Publications in Russian can be tracked by [18], [19]. Despite its fundamental nature and focus on the most serious mathematical calculations, Maple-class systems are needed by a fairly wide category of users: university students and teachers, engineers, graduate students, researchers, and even students of mathematics classes of general education and special schools. Maple is a typical integrated software system [19]–[21].

The main Maple document is the Worksheet, which is similar to editing in a text editor. Text can be formatted at the paragraph level, with different styles, or as symbols. The content of the document can be structured into sections, sub-sections, etc., all the way down to cells.

Like most CAS, the Maple interface combines text and command processor functions. Since version 8, Maplets have been added to the system to support visually-oriented dialog.

Maple (like Mathematica) integrates three languages: communication language, implementation, and programming.

The Maple kernel and all of its components improve from version to version. Many of the functions built into the system, like the kernel functions, can be used without any declaration, others need to be declared. There are several problem-oriented packages (packages) that cover many branches of classical and modern mathematics. The total number of functions in Maple, including those built into the kernel and placed in packages, exceeds 5500. The kernel (not in full composition) uses MATLAB and MathCad (starting from version 14, symbolic kernel MuPAD is used).

The main stages of development, additions in the Maple versions can be traced in [20], but it must be stated that machine learning, artificial intelligence, and data mining are not yet priorities for the system.

Wolfram Mathematica. Mathematica – is a computer algebra system developed by Wolfram Research company. It is one of the most powerful and widely used integrated multimedia-technology software package [6], [22]–[25]. Mathematica is recognized as a fundamental advancement in computer-aided design. It is one of the world's largest software packages by volume of modules and contains many new algorithms, as well as many unique developments. Mathematica lets users use virtually every analytic and numeric option, and it supports databases, graphics, and sound. Mathematica lets you work, analyze, manipulate, and graph almost any function of pure and applied mathematics. The system provides calculations with any specified accuracy; construction of two- and three-dimensional graphs, their animation, drawing geometric figures; importing, processing, exporting images and sound.

Mathematica has evolved from a program used primarily for mathematical and technical calculations to a tool widely used in various other areas [22], [23]. It is recognized among specialists as a development platform that fully integrates computation into the workflow from start to end, seamlessly guiding the user from initial ideas to deployed custom and industrial solutions.

Mathematica has a built-in Wolfram Language, including tools for creating programs and user interfaces, connecting external dlls, and parallel computing. The system's programming language is a typical interpreter; it's not designed to create executable files, but it incorporates the best of such programming languages as BASIC, Fortran, Pascal, and C. The Mathematica programming language supports all known paradigms: functional, structural, object-oriented, mathematical, logical, recursive, and more. It also includes visual-oriented programming tools based on the use of mathematical symbol templates, such as integral, summation, product, etc.; this language exceeds the usual general-purpose programming languages in its ability to perform mathematical and scientific computations.

Like all computer algebra, Mathematica is a type of software tool designed to manipulate mathematical formulas. Its main purpose is to automate the often tedious and in many cases difficult algebraic transformations. User works in the system with notebooks - NB documents, each document contains at least one section (cell). An explanation of the preference adopted here is a comparison with MS Excel, where the term cell is steadily and universally used. Those who have experience with Excel and Mathematica understand the difference and that in MS Excel it is cells, and in Mathematica notebooks it is more general objects.

NB documents can be opened, viewed, edited, saved, executed in their entirety, or individual cells. Notebook's interface contains many palettes (menus) and graphical tools for creating, editing, viewing documents, sending and receiving data to and from the core. Notebook includes one or several cells that can be grouped together as needed. Each cell contains at least one line of text or formulas, a digital audio or video object. Notebooks can be edited as text in any editor or in the Mathematica interface. The kernel performs the computations and can be run on the same computer as the interface, or on another computer connected through a network. Typically, the kernel is started when the computation begins. Cellss in Mathematica can be roughly divided into input and result (output) cells. In the input cells, the user enters or places commands, comments, multimedia objects, and they can be executable or otherwise; the executable cells are processed – the system returns results and displays them in output cells.

All versions of Mathematica include a powerful reference database, and the built-in Help, Documentation Center is an example of an NB document in itself. Without interrupting work on modules, you can clarify any function, option, directive, or service word; explore the capabilities of "live" examples to get and document results; and embed examples or code snippets from examples in your own code.

From the chronology of Mathematica versions. The first release of Mathematica was in June 1988, the basic concept being to create once and for all one system for different computations in a consistent and unified way. The basis for this was the creation of a new symbolic computer language for controlling, with a minimum number of inputs, the large number of objects involved in technical computation. Since its inception, all Wolfram Research Inc. developments have been regularly ranked first among IT achievements, highlighted by the media.

Release dates, additions, and updates to Mathematica are fully reflected in a number of publications and websites, e.g. [22], [23]. Experts note that the list of updates to Wolfram Mathematica reflects many completely new advances that have found application, development in other systems, and information technology. Wolfram Research Inc. developments are mostly characterized by interface continuity and the ability to use source code from previous versions.

About Mathematica features. A complete list of capabilities would require several times as much space as this presentation allows. For example, manual [24] has over 600 pages of content, but in fact it only outlines the basic functions of the CAS. Originally published in 1988 and updated in Mathematica 5, “The Mathematica Book by Stephen Wolfram”, Fifth Edition, 2003, is 1,488 pages long. A list of books by S. Wolfram’s books can be viewed at at [25].

Help system in Mathematica. The Documentation Center, Function Navigator, and Virtual Book are part of the Help system. These modules provide all necessary information to guide users through the language and functionality of Mathematica. Built-in documentation contains more than 150,000 representative and illustrative examples of Wolfram code. All documents are fully interactive; they are Mathematica notebooks in which the user can try out their own code and modify the examples directly in the Help system.

The theses outlined above are important from the position of developers of computer systems for artificial intelligence to understand the current state in closely related fields, in particular because computer algebra systems, which implement intelligent computations with the help of a computer, are also one of the (and quite successfully developed) areas to adopt artificial intelligence.

V. Examples of integrating WM tools into OSTIS applications

A. *Wolfram Mathematica. Current state*

Building on over thirty years of research, development, and use around the world, Mathematica and Wolfram are geared for the long term and especially successful in computational mathematics The roughly 6,000 functions (symbols) built into Wolfram allow the user to represent and manipulate a huge variety of computational objects – from special functions to graphics and geometric regions.

In addition, the Wolfram knowledge base [26] and its associated entity structure [27] allows to explain, interpret, and formalize hundreds of specific “things” (facts, situations, objects). For example: people, cities, food, structures, planets, etc. appear as objects that can be manipulated and counted.

B. *Wolfram knowledge base. Coverage areas*

The growing Wolfram Data Repository (WDR), based on Wolfram Alpha and the Wolfram Language, is now the world’s largest repository of computable knowledge. Covering thousands of fields, the WDR contains carefully selected expert knowledge obtained directly from primary sources. It includes not only trillions of data elements, but also a huge number of algorithms that encapsulate methods and models from virtually every domain. The Wolfram Knowledge Base is based on Wolfram’s three decades of accumulated computable knowledge. All data in the Wolfram KB can be used immediately for Wolfram computations. Every millisecond of every day, the Wolfram Knowledge Base is updated with the latest data.

Major coverage areas of WDR [26] are shown in Fig. 1.

Major coverage areas

- Units & Measures
- Places & Geography
- People & History
- Music
- Space & Astronomy
- Weather & Meteorology
- Health & Medicine
- Organizations
- Web & Computer Systems
- Dates, Times & Calendars
- Demographics & Social Statistics
- Arts & Media
- Math & Statistics
- Life Sciences
- Engineering
- Food & Nutrition
- Sports
- Technological World
- Words & Linguistics
- Money & Finance
- Art & Design
- Physics & Chemistry
- Earth Sciences
- Materials
- Educational Data
- Transportation

Figure 1. Coverage areas of WDR.

In [28] typical options for working with WDR in Education are outlined, as well as examples of interaction with Wikipedia.

With extensive statistics on hundreds of thousands of educational institutions around the world, Wolfram|Alpha can calculate answers to complex questions about education. For example, you can query what academic degrees students receive at prestigious universities, average enrollment figures by year for selected majors. In the examples [28] illustrations of the response to the query about the number of students in the Republic of Belarus, quantitative indicators for the leading universities of BSU and BSUIR are given. Ways to present knowledge and access to it are described. It is noted that access to the Wolfram knowledge base is deeply integrated in Wolfram Language (WL). Free-form linguistics makes it easy to identify many millions of entities and many thousands of properties, and automatically generates accurate Wolfram Language representations suitable for extensive further computation. WL also supports custom entity stores that allow you to perform the same computations as the built-in knowledge base and can be linked to external relational databases.

People interact with each other through speech and text, and this is called natural language. Computers understand people’s natural language using Natural Language Processing (NLP). NLP is the process of manipulating human speech and text with artificial intelligence so that computers can understand them. In [28] the basic NLP tools implemented in Wolfram Mathematica are noted. In particular: Speech recognition; Voice assistants and chatbots. Auto-substitution and auto-prediction. Email filtering. Sentiment analysis. Divertissements for the target audience. Translation. Social media analytics. Recruitment (staffing). Text summary (abstracting). Several representative examples with explanations of the functions of WL groups Structural Text Manipulation, Text Analysis, Natural Language Processing are mentioned.

Also in [28] examples of knowledge extraction, entities from Wikipedia articles are discussed. Wikipedia data uses the MediaWiki API to extract article and category content and metadata from Wikipedia. An article can be specified as a string or Wolfram Language object. Retrieving articles associated with language entities is provided by the WM TextSentences feature, in particular, it is possible to work with Wikipedia resources. Presented are specific results of the TextSentences function, with parameters WikipediaData, Entity, "Person", "AlexeiLeonov".

These examples of working with knowledge bases using WM tools, since the system kernel functions can be used in programs developed on other platforms, can be interpreted as proposals for the innovative improvement of existing tools, components of any intelligent computer systems, and of course the Ecosystem OSTIS.

VI. EXAMPLE OF INTEGRATING WOLFRAM MATHEMATICA WITH EDUCATIONAL OSTIS-SYSTEM PROTOTYPE FOR DISCIPLINE "COMPUTER SYSTEMS AND NETWORKS"

Here is an illustration of the combined use of WM and OSTIS-prototype for discipline "Computer Systems and Networks" ostis-system for working with computer network topologies. The results below show the possibilities of using the visualization performed in WM in the ostis-system. Moreover, implementations are available using an appropriate programming interface (it is possible to execute WL code hosted in the Wolfram cloud within a user program, such as Python or C++ [29]) or import, export tools. According to Mathematica $\$ImportFormats$ and $\$ExportFormats$ functions, it supports more than 100 formats, the list of formats is as follows:

3DS, ACO, Affymetrix, AgilentMicroarray, AIFF, ApacheLog, ArcGRID, AU, AVI, Base64, BDF, Binary, Bit, BMP, BSON, Byte, BYU, BZIP2, CDED, CDF, Character16, Character8, CIF, Complex128, Complex256, Complex64, CSV, CUR, DAE, DBF, DICOM, DIF, DIMACS, Directory, DOT, DXF, EDF, EML, EPS, ExpressionJSON, ExpressionML, FASTA, FASTQ, FCS, FITS, FLAC, GenBank, GeoJSON, GeoTIFF, GIF, GPX, Graph6, Graphlet, GraphML, GRIB, GTOPO30, GXL, GZIP, HarwellBoeing, HDF, HDF5, HIN, HTML, HTTPRequest, HTTPResponse, ICC, ICNS, ICO, ICS, Ini, Integer128, Integer16, Integer24, Integer32, Integer64, Integer8, JavaProperties, JavaScriptExpression, JCAMP-DX, JPEG, JPEG2000, JSON, JVX, KML, LaTeX, LEDA, List, LWO, M4A, MAT, MathML, MBOX, MCTT, MDB, MESH, MGF, MIDI, MMCIF, MO, MOL, MOL2, MP3, MPS, MTP, MTX, MX, MXNet, NASACDF, NB, NDK, NetCDF, NEXUS, NOFF, OBJ, ODS, OFF, OGG, OpenEXR, Package, Pajek, PBM, PCAP, PCX, PDB, PDF, PGM, PHPIni, PLY, PNG, PNM, PPM, PXR, PythonExpression, QuickTime, Raw, RawBitmap, RawJSON, Real128, Real32, Real64, RIB, RLE, RSS, RTF, SCT, SDF, SDTS, SDTSDEM, SFF, SHP, SMA, SME, SMILES, SND, SP3, Sparse6, STL, String, SurferGrid, SXC, Table, TAR, TerminatedString, TeX, Text, TGA, TGF, TIFF, TIGER, TLE, TSV, UBJSON, UnsignedInteger128, UnsignedInteger16, UnsignedInteger24, UnsignedInteger32, UnsignedInteger64, UnsignedInteger8, USGSDEM, UUE, VCF, VCS, VTK, WARC, WAV, Wave64, WDX, WebP, WLNet, WMLF, WXF, XBM, XHTML, XHTMLMathML, XLS, XLSX, XML, XPORT, XYZ, ZIP.

In the example below, the initial data (a particular graph of the topology of a computer network) is imported from a teaching ostis-system for the "Computer Systems and Networks" discipline, visualized by WM graphics, then solved the typical problem and the preferred final results are exported back to the teaching ostis-system for the discipline. Initial data, the specific graph of the network topology used next is shown in the Fig. 2.

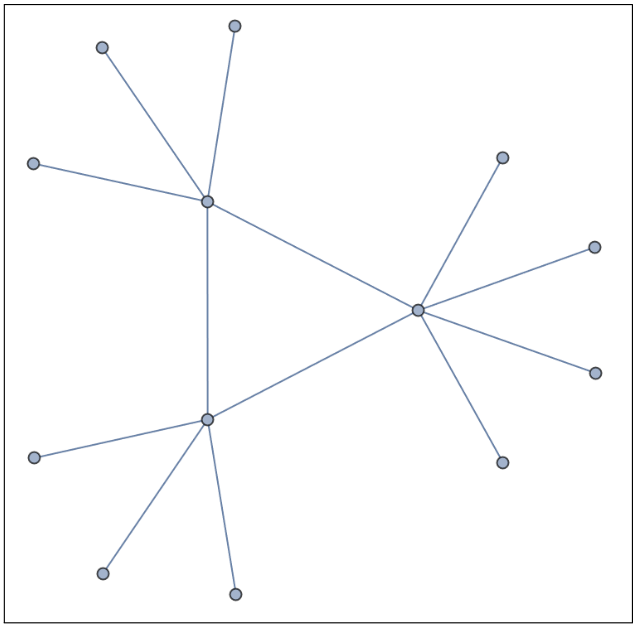
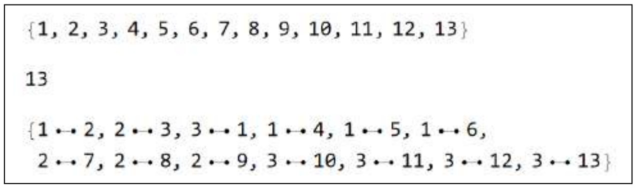


Figure 2. Network topology graph, nodes and connections (in the ostis-system).

The following illustrations are generated in WM.

For the imported graph in WM, you can get general information such as: number of vertices (network nodes), list of edges (connections between nodes), and visualize it. Fig. 3 shows the output of the vertex list (VertexList), the number of edges (EdgeCount), and the edges list (EdgeList).

The three output layouts are shown below for an example visualization. Fig. 4 shows vertices and edges with their weights. This form of representation is preferable for visualization of logical network topology, in which the directions of data flows are indicated. Edges with



```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}

13

{1 ↔ 2, 2 ↔ 3, 3 ↔ 1, 1 ↔ 4, 1 ↔ 5, 1 ↔ 6,
 2 ↔ 7, 2 ↔ 8, 2 ↔ 9, 3 ↔ 10, 3 ↔ 11, 3 ↔ 12, 3 ↔ 13}
```

Figure 3. Network topology graph, general information (output in Wolfram Mathematica).

no arrows are bidirectional, which is analogous to two opposing edges with the same weight.

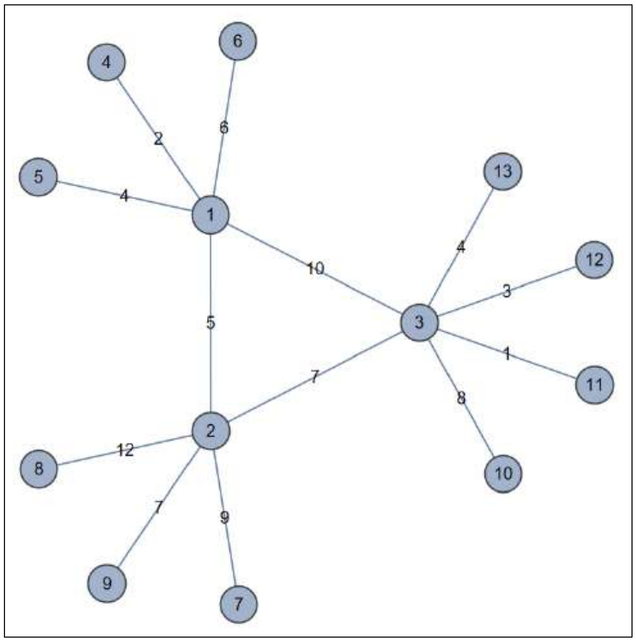


Figure 4. Network topology graph, connections with weights (output in Wolfram Mathematica).

The example output with specifying the LayeredEmbedding graph format, applying the vertex design options is shown in Fig. 5.

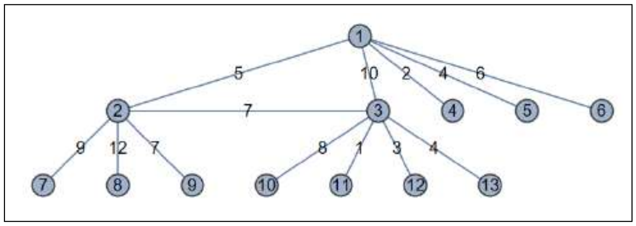


Figure 5. Network topology graph with LayeredEmbedding layout (output in Wolfram Mathematica).

The example output with specifying the LayeredDigraphEmbedding graph format, applying the vertex design options is shown in Fig. 6.

Let us consider one of the dynamic routing protocols, OSPF (Open Shortest Path First), as the example of the described network topology. This protocol is interesting because it uses the Dijkstra algorithm to find the shortest path [30].

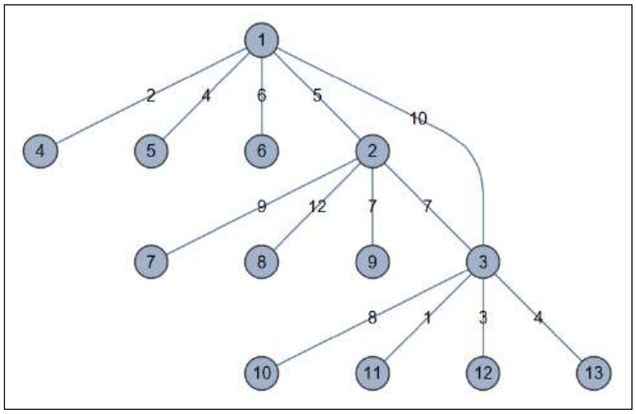


Figure 6. Network topology graph with LayeredDigraphEmbedding layout (output in Wolfram Mathematica).

The principle of the protocol is as follows:

1) Once routers are enabled, the protocol searches for directly connected neighbors and establishes communication with them.
2) Routers exchange information with each other about the networks connected and available to them – build a network map (network topology). This map is the same on all routers.
3) Based on this information, the SPF (Shortest Path First) algorithm is run, which calculates the best route to each network. This process is similar to building a tree, with the root being the router itself, and the branches being paths to available networks.

In the OSPF protocol, convergence occurs rather quickly due to the use of the Dijkstra algorithm [31].

Fig. 7 illustrates the example of solving the problem of finding the optimal route between two nodes in a network. The following Wolfram Mathematica functions were used: GraphDistance, NeighborhoodGraph, Sow, DirectedEdge, Placed, Union, and Flatten.

Consider the following situation: in this initial network topology the communications are upgraded. Between nodes 2 and 3 a fiber optic is laid, which provides more bandwidth and data transfer speed. As a result of the replacement, the “cost” of transmitting data over this link decreases from 8 conventional units to 3. How will the changes affect the solution to the problem at hand: the optimal route between nodes 4 and 12 of the network?

When routers are included in the network, they check the speed capabilities of all available data channels. In our situation, routers 2 and 3 will determine that the communication between them has improved qualitatively. In accordance with the OSPF protocol, these devices on request will transmit this information about the topology change to the neighboring router 1. Now, in the future, when deciding on a path to transmit data from node 4

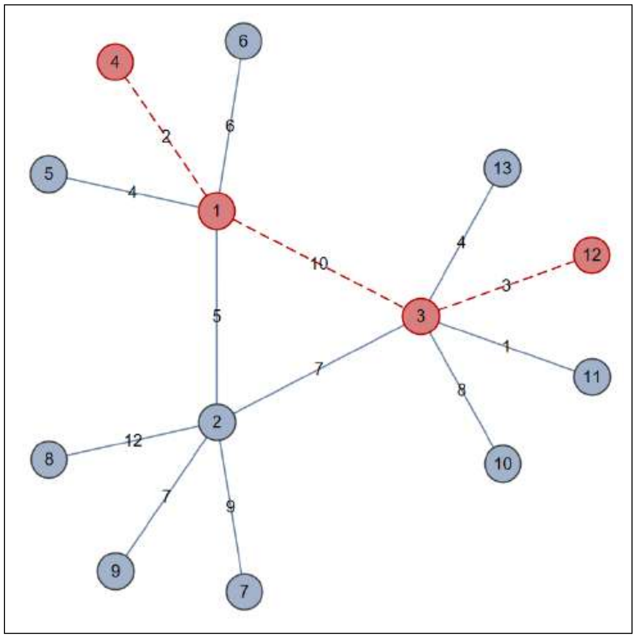


Figure 7. Solving the problem of finding the optimal route between two network nodes (output in Wolfram Mathematica).

to node 12, router 1 will choose a new shortcut. The solution to find the optimal route between two nodes in the changed network configuration is shown in Fig. 8.

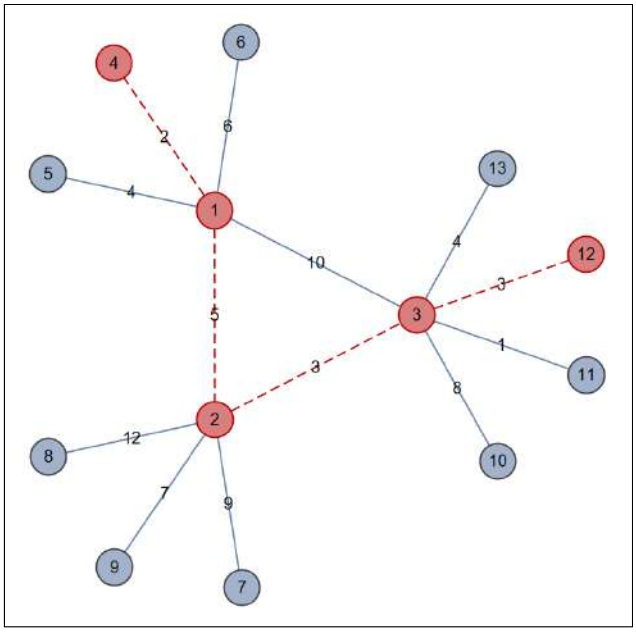


Figure 8. Solving the problem of finding the optimal route between two network nodes in updated topology (output in Wolfram Mathematica).

The results obtained and considered include labor-intensive for implementation in programming languages graphics problems, as well as mathematically and algorithmically complex problems of the subject area. Presented options of visualization, finding a solution require only a careful study of examples of the help system Wolfram Mathematica, certain programming skills, i.e. are available to most software engineers. Transferring results to other software applications isn't difficult either, because WM provides export options in any standard format.

VII. CONCLUSION

Programming language, high level of documentation, features of step-by-step problem solving and graphical visualization of initial data and computational results favorably differentiate Mathematica from other CAS. These features combined with an intelligent predictive interface help subsystem, provide wide range of opportunities for integrating Wolfram Language functions with components of Ecosystem OSTIS.

The presented example of addition of intellectual educational resource built within the ostis-system for discipline "Computer Systems and Networks" illustrates connection of difficult for programming functions of visualization and solution of complex mathematical problem of finding the optimal path in the info-communication network with a complex topology.

REFERENCES

[1] V. Golenkov, N. Guliakina, and D. Shunkevich, Otkrytaja tehnologija ontologicheskogo proektirovanija, proizvodstva i jekspluatacii semanticheski sovmestimyh gibridnyh intellektual'nyh komp'juternyh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems], V. Golenkov, Ed. Minsk: Bestprint [Bestprint], 2021, P. 690 (In Russ.)

[2] GeoS — Geometry Problem Solver. Available at: https://geometry.allenai.org (accessed 2023, Feb).

[3] Handwriting Recognition. Available at: https://mathpix.com/handwriting-recognition (accessed 2023, Feb).

[4] Create Graph online and find shortest path or use other algorithm. Available at: https://graphonline.ru/en (accessed 2023, Feb).

[5] M. Kovalev, A. Kroshchanka, V. Golovko, Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2022, pp. 173-186.

[6] V. D'yakonov, Entsiklopediya komp'yuternoi algebry [Encyclopedia of Computer Algebra], Moscow, DMK Press, 2009, P. 1264 (In Russ.)

[7] List of computer algebra systems. Available at: http://en.wikipedia.org/wiki/List_of_computer_algebra_systems (accessed 2023, Mar).

[8] V. Taranchuk, Osnovnye funktsii sistem komp'yuternoi algebry [Basic Functions of Computer Algebra Systems] : posobie dlya studentov fak. prikladnoi matematiki i informatiki, Minsk, BSU, 2013, P. 59. (In Russ.)

[9] Maxima, Sistema komp'yuternoi algebry [Maxima, Computer Algebra System]. Available at: http://maxima.sourceforge.net/ru (accessed 2023, Feb) (In Russ.)

[10] N. Stakhin, Osnovy raboty s sistemoi analiticheskikh (simvol'nykh) vychislenii Maxima [Fundamentals of working with the system of analytical (symbolic) calculations Maxima]. (PO dlya resheniya zadach analiticheskikh (simvol'nykh) vychislenii), Moscow, Federal'noe agentstvo po obrazovaniyu, 2008. 86p. (In Russ.)

[11] AXIOM. The Scientific Computation System. Available at: http://axiom-developer.org (accessed 2023, Feb).

[12] OpenAxiom: The Open Scientific Computation Platform. Available at: http://open-axiom.sourceforge.net (accessed 2023, Feb).

[13] FriCAS — an advanced computer algebra system. Available at: http://fricas.sourceforge.net (accessed 2023, Feb).

[14] MathWorks. MATLAB. The Language of Technical Computing. Available at: http://www.mathworks.com/products/matlab/ (accessed 2023, Mar).

[15] V. D'yakonov, MATLAB. Polnyi samouchitel' [MATLAB. Complete tutorial], Moscow, DMK Press, 2012, P. 768 (In Russ.)

[16] MATLAB Versiya — Osobennosti i preimushchestva versii MATLAB [MATLAB Version — Features and Benefits of MATLAB Version]. Available at: https://ru.education-wiki.com/2303364-MATLAB-version (accessed 2023, Mar). (In Russ.)

[17] Maplesoft. Books. Available at: http://www.maplesoft.com/books/index\.aspx (accessed 2023, Feb).

[18] V. Aladjev, V. Vaganov Modul'noe programmirovanie: Maple vs Mathematica, and vice versa [Modular programming: Maple vs Mathematica, and vice versa], CA: Palo Alto, Fultus Corp., 2011, P. 417 (In Russ.)

[19] V. D'yakonov, Maple 10/11/12/13/14 v matematicheskikh raschetakh [Maple 10/11/12/13/14 in mathematical calculations], Moscow, DMK Press, 2011. 800 p. (In Russ.)

[20] Maple Product History. Available at: http://www.maplesoft.com/products/maple/history (accessed 2023, Mar).

[21] V. Alad'ev, Sistemy komp'yuternoi algebry: Maple: Iskusstvo programmirovaniya [Computer Algebra Systems: Maple: The Art of Programming], Moscow, Laboratoriya Bazovykh Znanii, 2006, P. 792 (In Russ.)

[22] Mathematica. Istoriya versii. Available at: http://ru.wikibooks.org/wiki/Mathematica/История_версий (accessed 2023, Mar). (In Russ.)

[23] Mathematica Quick Revision History. Available at: http://www.wolfram.com/mathematica/quick-revision-history.html (accessed 2023, Mar).

[24] V. D'yakonov, Mathematica 5/6/7. Polnoe rukovodstvo [Mathematics 5/6/7. Complete Guide], Moscow, DMK Press, 2009, P. 624 (In Russ.)

[25] Stephen Wolfram. Books. Available at: http://www.stephenwolfram.com/publications/books (accessed 2023, Mar).

[26] Wolfram Mathematica: Modern Technical Computing. Available at: https://www.wolfram.com/mathematica (accessed 2023, Feb).

[27] V. Taranchuk, Intellektual'nye vychisleniya, analiz, vizualizatsiya bol'shikh dannykh [Intelligent Computing, Analysis, Visualization Of Big Data]. *BIG DATA i analiz vysokogo urovnya [BIG DATA and Advanced Analytics]*, 2019, no. 1, pp. 337–346. (In Russ.)

[28] V. Taranchuk, Integration of computer algebra tools into OSTIS applications. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2022, pp. 369-374.

[29] APIFunction — Wolfram Language Documentation. Availabe at: https://reference.wolfram.com/language/ref/APIFunction.html (accessed 2023, Mar).

[30] OSPF Version 2. Available at: https://www.ietf.org/rfc/rfc2328.txt (accessed 2023, Apr).

[31] T. Roughgarden, Algorithms Illuminated: Graph algorithms and data structures. Part 2, New York, Soundlikeyourself Publishing LLC, 2018, P. 222

Примеры интеграции инструментов Wolfram Mathematica в приложения OSTIS

Таранчук В. Б., Савёнок В. А.

В рамках концепции конвергенции и унификации интеллектуальных компьютерных систем нового поколения, обсуждаются технические решения, приведены примеры разработки и модернизации, интеграции средств Экосистемы OSTIS с системой компьютерной алгебры (СКА) Wolfram Mathematica (WM).

На примере интеграции в специализированном комплексе интеллектуального образовательного ресурса по дисциплине “Компьютерные системы и сети” рассмотрены возможности использования инструментов WM в ostis-системе при решении задач, связанных, в частности, с топологией инфокоммуникационных сетей.

Показано применение инструментов WM для визуализации топологии сети, а также эмуляции поиска оптимального маршрута передачи данных.

Received 10.03.2023