An approach to hybrid learning resource design for training professionals in Computational Science

Klavdiya Bochenina, Alexey Dukhanov, Maria Karpova, Vadim Shmelev
ITMO University, Saint Petersburg, Russian Federation
k.bochenina@gmail.com, dukhanov@niuitmo.ru, karpovams@niuitmo.ru, vad1611@yandex.ru

Abstract
Recent trends of mass character and individualization of education impose new requirements on technology-enhanced learning environments. One of the main problems in maintaining high-quality education in knowledge-intensive areas such as Computational Science is rapid obsolescence of learning materials due to the accelerating rate of an appearance of novel research and technological achievements. In this paper, we propose an approach to a fast design of hybrid learning resources which combine different forms of representations of novel scientific results (including hypertext and software) with out-of-the-box support of learning objects’ execution in cloud computing platforms. The experimental part of the study considers the preparation of hybrid learning resources for Master students in High-Performance Computing Department of ITMO University, Russia. We show how the implementation of proposed approach on the basis of CLAVIRE platform speeds up the translation of scientific results into reusable learning materials and thus contributes to the improvement of the quality of training professionals in Computational Science.

Keywords: research objects, reusable learning objects, learning resource design, Computational Science education

1 Introduction

The rapid growth of information-communication technologies (ICT) and their entry into practically all branches of science provide new opportunities and raise new challenges for higher education. Two major trends in modern education — individualization and mass character — are both inspired by and supported by the ubiquity of ICT. ICT are used not only to provide easy remote access to reusable learning materials (maintaining the mass character of education) but also to create new learning objects in a reproducible, automated manner (maintaining fast deployment of personalized learning trajectories). Interactive, reusable learning objects which combine different kinds of media with computer simulation tools are gradually replacing more traditional forms of learning materials, especially in dynamically developing domain areas. Approaches to the design of learning materials are changing accordingly to support quick delivery of newest scientific and technological achievements to the classrooms.
Learning resources developed with traditional methods quickly become outdated because of the length and high labor-intensity of a design cycle. This can lead even to the situation when learning material is obsolete already at the time of its publication (for example, if a new version of API standard was launched during a preparation of a learning material for software engineering students). From the other side, time-consuming preparation of learning resources makes fast incorporating of research into practice extremely difficult. This becomes crucial during training of professionals in knowledge-intensive areas such as Computational Science.

Existing concepts and standards of development and integration of different types of learning resources such as Reusable Learning Objects (RLO) and Learning Tools Interoperability (LTI) have a number of drawbacks impeding their usage as a basis for fast translation of scientific results into practice. These drawbacks include mono-disciplinary orientation, reproduction of existing educational processes (rather than the creation of ICT-enhanced learning environments) and the restricted students’ access to the primary research results (such as scientific papers, reports, and software). With the rapid growth of interdisciplinary research, training of professionals in Computational Science should focus on providing hands-on experience of working in scientific and engineering teams during solving real-world problems. This experience includes collaboration with experts from different domains, usage of primary research materials, development of new methods, approaches and technologies individually or in groups etc. In other words, it is necessary to bridge the gap between competencies of a graduate and the requirements of an employer by offering near real-time problems to students during the educational process.

In this paper we address two main questions:

(i) How to maintain a fast preparation of actual learning materials to support a translation of novel scientific and technological achievements into educational process? (Q1)
(ii) How modern ICT (including cloud computing technologies) can be used to enhance training of high skilled professionals in Computational Science? (Q2)

The rest of the paper is organized as follows. Section 2 presents the overview of existing concepts, models, and approaches which are used to create learning materials including those which incorporate different kinds of scientific results within a single object. Section 3 is dedicated to the formal description of hybrid learning resources (HLR) and the underlying principles of translation of scientific results into HLR. Also, Section 3 addresses the technical issues of design, deployment, and usage of HLR on the basis of cloud computing platform. Section 4 presents the results of implementing the proposed approach during Masters’ training in High-Performance Computing Department of ITMO University. Finally, Section 5 presents conclusions.

2 Background and Related Work

As one of the goals of our work is to develop an approach to incorporate scientific results and other forms of representation of codified knowledge within a single framework providing unified interface for accessing them, here we provide an overview (parts A, B) of existing concepts of learning objects (LO) and their application to research-based education. Part C describes concepts and approaches which are currently used to disseminate novel results of scientific activities in higher education.

A. Reusable Learning Objects (RLO).

The concept of the reusable learning object (RLO) became widely known because it clearly describes the contents of a learning unit that can be simply used in instructional design. According to [1], there are four types of RLO in the sense of horizontal reusability (subject specificity): generic, interdisciplinary, subject specific, resource specific. Interdisciplinary RLO can be used in frames of several disciplines for students with a different background.
The information object is often used as a base for a learning object (as proposed by Learnativity Content Model and CISCO Standard for RLO [2]). Allen and Mugisa [3] review the most widely used learning object models and introduce criteria characterizing them to increase reusability of learning resources in the combination with OOGLOM model. The lowest level of learning object granularity (shown in [3]) is presented by content assets (items) which are classified by the type of media (e.g. text, audio, image). These items are used to create information objects. In the case of LOs based on scientific results, existing studies usually do not consider information objects or information resources as a core of an LO and use the concept of Generative Learning Object instead of it ([4], [5]).

B. Object-oriented programming (OOP) and Generative Learning Objects (GLO).

Reusability as an essential property of learning objects has been used by several researchers to connect concepts of LO and object-oriented programming. [6] explains the application of OOP concept by reusing LO, and [7] suggest to do it through metadata tags. [8] states that LO as an object in OOP has methods (typically rendering and assessment) and properties (content and relationships to other resources). Some of the attempts ([6], [8]) to describe a learning object through the concept of OOP were criticized (see, e.g. [9]) as they only bring complexity to the description of the LO concept and do not facilitate practical use of LO.

Some studies provide descriptions of how to use the OOP concept together with the concept of RLO. The first step in this direction was made in [10] where authors explain how to consider learning object as an instance of OOP class (they propose to inherit all learning objects from the superclass LearningObject). In [8, 10] authors use term ‘learning resource’ (instead of ‘information resource’) as a core of the LO. In [10] such properties as Version, Context object and Combination object are introduced. Authors of [11] clarify the way of implementation of the OOP concepts (class structure, inheritance, aggregation, and polymorphism) for learning objects. However, the paper provides only an example of a hierarchy of types of learning activities strictly connected with LO. Thus, the methods of usage of OOP principles during LO creation are weakly represented.

The Generative Learning Object (GLO) concept was proposed in [12]. [13] define GLO as “an articulated and executable learning design that produces a class of learning objects”. To use GLO (for making an instance of RLO), the context in which it will further be used, should be analyzed. [14] describes the following types of contexts: thematic context, pedagogical context, learner context, organizational context and historical/statistical context. Authors of [4, 5] study the application of learning objects in the field of Computer Science education (where students are to be taught how to develop and deploy software to solve interdisciplinary problems). The context of LOs in [4] is described by the TPACK (Technological Pedagogical Content Knowledge), it consists of three main domains: content, pedagogical and technological. Authors of [5] mark the value of context and distinguish context-based parameters (curriculum objective, learning activity, learner’s previous knowledge level, learning pace, learning method) and content-based parameters (such as algorithm’s type).

C. Dissemination of scientific results in education.

The results of research activities can be categorized into several types which have different electronic representations (electronic versions of scientific manuscripts, software and software documentation, visualization tools etc.). ICT allows facilitating different processes related to obtaining and dissemination of scientific results. Along with services which enhance scientific collaboration by providing the access to databases of papers and profiles of researchers (e.g., ResearchGate and Mendeley), there exist platforms which allow incorporating different types of media within the electronic version of the manuscript (e.g. “Article of the Future” project [15] by Elsevier). These platforms can also be used by educands to investigate current state-of-the-art in the area of their specializations.

The next step is a direct interaction with scientific content which can be supported by providing to students an access to a scientific software. Such access is often organized on the basis of cloud computing technologies in frames of models SaaS, PaaS or IaaS [16]. Usage of cloud computing is
convenient as it combines reusability and unification of learning resources with the hiding of technical
details of deployment and computational resources’ consumption from an end user. To meet the
requirements of interdisciplinary research and education it became necessary to provide a way to
combine several software packages in a single logical unit. This led to the introduction of a concept of
’scientific workflow’ (SWF) which is a directed acyclic graph of computational tasks. Construction
and execution of SWFs is supported by a variety of modern scientific workflow management systems
(SWFMS) such as Taverna, Kepler, and Pegasus. The concept of Research Object (RO) was
introduced in [17]. It can be viewed as a container of scientific resources which are obtained during all
stages of scientific knowledge creation. In this paper, we are focused on learning resources which are
more general than RO in a sense that single HLR can be constructed from different parts of several
ROs in many possible ways (thereby implementing the OOP principle of polymorphism).

Summarizing, although RLO and GLO are in the center of attention of many researchers, and there
exists the set of basic ideas on how to create LO on the basis of different conceptual frameworks
(including OOP), there is a lack of methods describing full cycle of representation of scientific results
in a form of RLO or GLO. This paper is focused on providing the conceptual ground, methods, tools
and technological support of fast design and deployment of hybrid learning resources incorporating
different forms of scientific results. Thus, it is aimed to fill the gap between theoretical concepts of
RLO/GLO and their implementations in technology-enhanced learning environments (in particular, we
consider those deployed on the basis of cloud computing platforms). Although results presented in
Section 4 are focused on the preparation of HLR for education in Computational Science, concepts
and notations described in Section 3 can be used in other domains.

3 Formal description and design of hybrid learning resources

Here we present the formal description of components of hybrid learning resources (HLR). We call
considered learning resources ‘hybrid’ as they incorporate different types of objects and resources
(including both learning and computational resources) in frames of different educational contexts. The
basic concepts that we use further are: (i) informational object (IO) — an entity which describes a fact,
a term, a concept, a procedure, a process or a computer program and contains keywords, link to the
content and form of its representation; (ii) informational resource (IR) — a set of IOs combined by a
certain feature; (iii) learning object (LO) — contains links to IOs and information about how they are
applied in education, combines materials for theoretical and practical learning with assessment tools;
(iv) abstract learning resource (ALR) — contains links to IRs and LOs and information about contexts
of their usage. Context of ALR is represented by a set of three parameters: (i) technical context (e.g. a
number of processors necessary to perform a simulation), (ii) domain context (e.g. a number of agents
in multi-agent model), (iii) educational context — a form of training activity (e.g. lecture or master
class). Thus, ALR as a combination of the content and the context deployed on the computing
platform is a hybrid learning resource.

We consider a research object as a basis of the creation of HLRs. Research object (as a result of
activities performed by a team of researchers in frames of mono- or interdisciplinary project) can have
different forms (see [18] for the detailed description). In the area of Computational Science, the usual
form of practical representation of the results is software package (1) which can be considered as a
mapping $F$ from inputs $Ins$ to outputs $Outs$ according to predefined formats of data. Software packages
are supplied with a documentation which contains: (i) annotation, (ii) description of functionality,
(iii) ways of program’s execution, (iv) description of input and output data formats, and (v) technical
requirements.

$$\text{Package} = \{\text{Ins}^{\text{Pack}}, \text{Outs}^{\text{Pack}}, F_{\text{conv}}^{\text{Pack}}\}. \quad (1)$$
Scientific reports usually contain experimental documentation which includes: (i) the description of the goal of an experiment, (ii) the description of the object and the methodic of an experiment, (iii) the protocol of an experiment, and (iv) the description of the results of an experiment. Another form of representation of the results is a scientific paper which also has a strict structure (introduction, problem statement, related work, models and methods, experimental study, results and discussion).

To maintain the access to software package via cloud computing platform it is necessary to introduce a transformation to represent basic package (1) as a cloud package:

\[ PM \left( PMI \left( Ins^{\text{ClPack}}, \text{Package}, PMO \left( Outs^{\text{Pack}} \right) \right) \right) : \]

\[ \text{Package} \rightarrow \text{CloudPackage} = \left\{ Ins^{\text{ClPack}}, P^{\text{ClPack}}, Outs^{\text{ClPack}}, F^{\text{ClPack}} \right\}, \]

(2)

Here \( PMI \) is a mapping from inputs of cloud package to inputs of the basic package, \( PMO \) is a mapping from outputs of basic package to outputs of cloud package, \( P \) is a function which checks the correctness of input data, indices \( ClPack \) and \( Pack \) denote cloud and basic packages, respectively.

![Diagram](http://example.com/diagram.png)

**Figure 1. The scheme of abstract learning resource design**

Research objects with different representations are conjugated in frames of ALR. The main scheme of abstract learning resource design is showed in Figure 1. The basic materials (the content of informational resources) remain placed at initial locations (e.g. databases of publications) while ALR operates with links to IRs. The methodical materials for theoretical and practical learning, and for an assessment of learning outcomes are prepared depending on the particular context parameters. Changes in informational resources (e.g. modification of source code of software) are mapped in corresponding cloud services (and learning resources) automatically. Along with IRs and context parameters, ALR includes a virtual function which identifies the behavior of ALR when it is used in education. The particular implementations of ALR are determined by the overriding of this function in derived classes. This explains how the principles of OOP are incorporated in the scheme of ALR design.
As HLRs are deployed on the basis of cloud computing platform, the forms of representations of executable products reflect those available in the platform itself. In this study, we use CLAVIRE [16] eScience infrastructure platform as a basis for deployment. CLAVIRE provides the following forms of IOs: package, SWF, Virtual Simulation Object (VSO) [19] and VSO project. VSO is a different form of representation of SWF incorporating models from different domains within a single system. VSO project is a composition of VSOs. Available software and models are combined in HLR with hypertext materials. The latter can be developed in a short time having a set of scientific papers, software, and experimental documentation.

To embed the software into cloud computing platform, one needs to prepare the formalized description of input and output data (1) and the rules of transformation of a basic package into cloud package including constraints of correctness (2). In CLAVIRE, embedding of packages is performed using the specialized language of package description — EasyPackage. Working with packages is also available via graphical user interface (GUI) of Package Manager. Hypertext materials are supported with PDF viewer. Creation and usage of cloud packages and SWFs in CLAVIRE is performed with Domain-Oriented Interface and SWF editor. They provide graphical user interfaces: (i) to create SWFs from a given packages specifying their relations and the rules of interconnections of input and output data of different packages, (ii) to launch experiments with existing SWF. A student can try different scenarios of usage of SWF paradigm from launching a single package with varying parameters to embedding his or her own packages into the platform. VSO is supported in CLAVIRE by VSO designer component, and VSO projects — by VSO Builder. The access to informational objects (including executable ones) could also be supported via traditional Learning Content Management Systems (LCMS), for example, Moodle. In such a case, interfaces of CLAVIRE platform are embedded into LCMS using LTI.

Summarizing, design of HLR includes the following steps:

(i) embedding of software packages into cloud environment (if necessary);
(ii) creation of the list of cloud services and hypertext materials (i.e. IOs) required for HLR;
(iii) preparation of informational resources;
(iv) determination of technical, domain, educational contexts;
(v) determination of methodical recommendations for theoretical and practical training;
(vi) setting the links to existing learning objects;
(vii) implementation of learning resources on the basis of ALR.

The proposed approach allows operating with simple building blocks (IOs and IRs) on the high level of abstraction. Due to this, it maintains reusability of embedded packages, quick response to changes in IOs (in semi-automatic mode), native support of the results of research in different forms of representations and the opportunity to develop a family of learning resources from a single ALR by overriding of the scenarios of usage of its content in education. This explains its appropriateness for fast design and deployment of learning resources on the basis of scientific results (i.e. Q1 from the Introduction). As for Q2, proposed approach of HLR design uses cloud computing platform not only as a location of embedded packages but also as a binding basis of the learning environment providing to the students a unified access to methodical materials, methods, models, algorithms, software, computational resources and assessment tools. Students can not only use tools developed by senior researchers but also create their own models and embed them into the platform to share them with other participants of research and educational processes. Doing this, they repeat the main stages of knowledge co-creation process in which they will be involved in their professional life and, as a result, they improve the skills which are preferable for their further employers.
4 Experimental study

The goal of the experimental study was to demonstrate implementation of the proposed approach to HLR design and to show to which extent its usage allows facilitating the development of learning materials. Presented results were obtained during preparation of LRs for Master programs in Computational Science (High-Performance Computing Department of ITMO University). Our approach is oriented to a fast translation of scientific results into educational practice, and in the first example of its implementation we consider the results on simulation of vehicle evacuation as the starting point for HLR design (Figure 2). The description of these results can be found in [20].

Figure 2. Implementation of hybrid learning resources: simulation of vehicle evacuation
According to (2), packages \texttt{stsim} and \texttt{ptsim} were embedded into CLAVIRE. Both of these packages are used to simulate the evacuation of vehicles in a city (the difference is that \texttt{ptsim} provides parallel implementation). On the basis of these two packages, we prepared five informational objects with different usage scenarios. The next step according to Figure 1 is the preparation of IR using links to IOs. To obtain HLR from IR, it is necessary to provide the values of context parameters for different types of contexts and develop methodical recommendations. The domain and technical context include parameters of algorithms for optimal path finding, a number of agents, a number of processors (for parallel version), a city map and navigation parameters \textit{pa}. Figure 2 illustrates three possible implementations of HLR with different assignments of parameters of the ALR.

The first example shows how a single package can be used as a basis of HLR design. The second example aims to show the usage of SWF and VSO forms of representing the scientific results for the same purpose. Corresponding IR is based on two packages that offer optimal path finding between bus stops for public bus scheduling. This IR can be used as a part of HLRs in a variety of ways [18, 21] (different implementation of method ‘learn’): (i) as a black box without examining the structure of application, (ii) with support of editing of workflow structure, (iii) with support of source code modification. The parameters of context thus are determined according to the problem to be solved. Figure 3a shows the scheme of SWF, and Figure 3b shows its VSO representation in CLAVIRE.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{swf_vso.png}
\caption{a) The scheme of SWF; b) VSO representation of SWF}
\end{figure}

Figures 2, 3 represent examples of HLR which were designed with the use of proposed approach. During its application in the preparation of materials for Master programs in Computational Science, we prepare six ALRs for the interdisciplinary problem of simulation of vehicle evacuation. The summary of results is presented in Table 1. In comparison with traditional methods of preparation of methodical materials, proposed approach allows designing learning resources 3.9 times faster (in average). Using only two cloud packages (\texttt{stsim} and \texttt{ptsim}), 20 learning resources for 10 different subjects were developed. This confirms high reusability of executable objects embedded into a cloud platform. If necessary, the translation of existing learning resources into a new discipline is performed by tuning educational and domain contexts which take no more than 5–7 minutes per HLR.

The experience of applying proposed approach to the design of hybrid learning materials for educational programs in Computational Science shows that it is appropriate to perform fast translation of scientific results into higher education. This becomes possible according to following distinguishing characteristics of the design process: (i) high reusability of IOs that is once embedded in a cloud computing platform (see Figure 2), (ii) polymorphic implementation of HLR by variation of context parameters, (iii) using primary scientific materials without their reprocessing by a teacher.
An approach to hybrid learning resource design for training professionals in Computational Science

Klavdiya Bochenina et al.

Table 1. ALR for simulation of vehicle evacuation

| No | Scenario                              | Subjects                                                                 | A number of uses of package | Time consumption (approach/traditional method) |
|----|---------------------------------------|-------------------------------------------------------------------------|-----------------------------|-----------------------------------------------|
| 1  | Demonstration                         | City transport, vehicle routing methods, operation research, graph theory, system analysis | 5                           | 15/120                                       |
| 2  | Parametric demonstration (PD)         | City transport, emergency evacuation, decision support systems, system analysis | 4                           | 30/120                                       |
| 3  | PD (parallel performance)             | Urgent computing, eScience technologies                                 | 2                           | 40/160                                       |
| 4  | Practical work (parallel performance) | Parallel computing, eScience technologies                              | 2                           | 40/120                                       |
| 5  | Practical work (optimal routes)       | Emergency evacuation, decision support systems, system analysis, vehicle routing methods | 4                           | 40/120                                       |
| 6  | Project (navigation of vehicles)      | Emergency evacuation, decision support systems, vehicle routing methods | 3                           | 45/180                                       |
|    | Total                                 |                                                                         | 20                          | 210/820                                      |

5 Conclusions

The fast obsolescence of learning materials in dynamically developing, knowledge-intensive areas such as Computational Science raises new challenges in learning resource design. In this paper, we propose an approach to the design of hybrid learning resources which uses principles of OOP and cloud computing technologies to support the process of translation of scientific results into the educational process. This approach is focused on high reusability of informational objects embedded into cloud computing platforms which is achieved by separation of the content of a learning resource and the context of its usage. We demonstrate how an abstract learning object can be created step by step using links to informational resources, methodical descriptions, and context parameters, and how the usage of ALR could facilitate the process of learning resource design. The native support of executable objects (as scientific software and scientific workflows) by modern cloud platforms allows combining in ALR theoretical information with parametrized computer models. This provides the wide range of opportunities for the flexible and fast design of actual learning resources on the basis of the research results. The latter was confirmed during the implementation of proposed approach in High-Performance Computing Department of ITMO University. The results of the experimental study demonstrate that design of hybrid learning resources with the proposed approach is several times faster than using traditional methods while keeping the high reusability of packages embedded into a cloud platform.

Acknowledgements

This work was partially supported by Russian Foundation for Basic Research (project 15-29-07034_16). This paper is financially supported by Government of Russian Federation, project "Big data management for computationally intensive applications" (project #14613).
References

[1] Currier Sarah and Lorna M. Campbell. Evaluating learning resources for reusability: the "dner & learning objects" study. ASCILITE, 8:139–148, 2002.
[2] CISCO, Syst. Reusable learning object strategy: Designing and developing learning objects for multiple learning approaches. White paper, CISCO; 2003.
[3] Allen Claudine A., and Ezra K. Mugisa. Improving Learning Object Reuse Through OOD: A Theory of Learning Objects. Journal of Object Technology, 9(6):51–75, 2010.
[4] Burbaitė Renata and Kristina Baspalova. Model-driven processes and tools to design GLO for CS education. In Comp. in Education (SIIE), International Symposium, p 139–144. IEEE, 2014.
[5] Renata Burbaitė et al. Context Aware Generative Learning Objects for Teaching Computer Science. Intern. J. of Eng. Ed., 30(4):929–936, 2014.
[6] Quinn Clark, and Samantha Hobbs. Learning objects and instruction components. Educational Technology & Society, 3(2):13–20, 2000.
[7] Bannan-Ritland Brenda, Nada Dabbagh, and Kate Murphy. Learning object systems as constructivist learning environments: Related assumptions, theories, and applications; 2000.
[8] Robson Robby. Object-oriented Instructional Design and Applications to the Web. In EdMedia: World Conference on Educational Media and Technology, 1999(1):698–702, 1999.
[9] Sosteric Mike, and Susan Hesemeier. Engineering a future for web-based learning objects. In Web Engineering, pages 120–123. Springer Berlin Heidelberg, 2003.
[10] Robson Robby. Object-oriented Instructional Design and Applications to the Web. In EdMedia: World Conference on Educational Media and Technology, 1999(1):698–702, 1999.
[11] Chrysostomou Chrysostomos, and George Papadopoulos. Applying Object-Oriented Principles to the Analysis and Design of Learning Objects. The European Conf. on e-Learning; 2007.
[12] Tom Boyle, Dawn Leeder, and Howard Chase. To boldly GLO — towards the next generation of learning objects. World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pages 28–33, 2004.
[13] Raquel Morales, Dawn Leeder, and Tom Boyle. A case in the design of generative learning objects (GLOs): applied statistical methods. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications; 2005.
[14] Mohammed Phaedra, and Permanand Mohan. Contextualizing learning objects using ontologies. Computational Intelligence, 23(3):339–355, 2007.
[15] IJsbrand A. Bringing Digital Science Deep Inside the Scientific Article: the Elsevier Article of the Future Project / A. IJsbrand, S. Atzeni, H. Koers, Z.-s. Elena // Lib. Q. – 2014. – T. 24 – № 4.
[16] Konstantin V. Knyazkov, Sergey V. Kovalchuk, Timofey N. Tchurov, Sergey V. Maryin, and Alexander V. Boukhanovsky. CLAVIRE: e-Science infrastructure for data-driven computing. Journal of Computational Science, 3(6):504–510, 2012.
[17] Bechhofer S. Research Objects: Towards Exchange and Reuse of Digital Knowledge / S. Bechhofer, D. De Roure, M. Gamble, C. Goble, I. Buchan // Nat. Preced. – 2010.
[18] Alexey Dukhanov et al. Approach to Automation of Cloud Learning Resources’ Design for Courses in Computational Science Based on eScience Resources with the Use of the CLAVIRE Platform. Procedia Computer Science, 51:1957–1966, 2015.
[19] Kovalchuk S. V. et al. Virtual Simulation Objects concept as a framework for system-level simulation //2012 IEEE 8th International Conference on E-Science.
[20] Karbovskii V., Voloshin Daniil V., Karsakov A., Bezgodov Alexey, Zagarskikh A. Multiscale agent-based simulation in large city areas: emergency evacuation use case // Procedia Computer Science. - 2015. - Vol. 51. - pp. 2367-2376
[21] Alexey Dukhanov, Maria Karpova, and Klavdiya Bochenina. Design virtual learning labs for courses in computational science with use of cloud computing technologies. Procedia Computer Science, 29:2472–2482, 2014