Information Resource based on Scientific Software as a core of Interdisciplinary Learning Resources

Maria Karpova, Vadim Shmelev, Alexey Dukhanov
Department of High-Performance Computing
ITMO University
St. Petersburg, Russian Federation
karpovams@niuitmo.ru, vad1611@yandex.ru, dukhanov@niuitmo.ru

***Abstract* — This paper introduces the individualized interdisciplinary learning resources based on scientific software, that usually serves for solving domain problems (optimization, simulation etc.). Integrating this type of software in learning process is a problem that we are solving by suggesting a new way to describe this software and introduce a framework that helps to form learning resource from it. We propose the concept of creating new advanced type of generative learning objects (GLOs) that changes not only structure of content presentation for different purposes (described by the contexts' parameters) but also it changes the way of learning and evaluating. The object-oriented programming (OOP) paradigm is used for combining concepts of GLO and learning objects (LO) based on information object to formulate our proposed approach. In this article we describe the principles of generating the interdisciplinary learning resource for a master program's learning course. Also we provide example of scientific software (transport schedule optimization) and show how it should be described for being used in learning process.**

Keywords — interdisciplinary learning object, generative learning object, reusable learning object learning object, object-oriented paradigm

I. Introduction and Literature Review

Interdisciplinary skills that are connected with creating and usage of scientific software are on demand today. As a consequence of this, there is a need for subject specialists who are able to work with software that is specific to their domain and programmers who are knowledgeable about domain concepts on which they work. Using scientific software in learning process in master programs enables students to obtain interdisciplinary skills, but there are many different ways on how to use it for educational purposes and often instructors find it difficult to integrate them in learning courses. For facilitating this process , we propose a new type of learning object – interdisciplinary learning resource (ILR), that allows exploring interdisciplinary software from different points of view (as a programmer, system analytic or domain specialist) on various complexity level (using program features and components, analyzing program and algorithm structure, deal with code). Also, different learning styles and student specific should be taken into account. Thus, there are many parameters of ILR and principles for creating it. But to perform it well we need to have a framework that allow to unify description of ILR's core – scientific software. There are two concepts that can help with it – OOP and GLO. So we need to find a place of scientific software in the concept of reusable learning objects (RLO) and the way of using OOP and GLO approach for ILR description.

This work was financially supported by the Ministry of education and science of the Russian Federation.

A. RLO and scientific software as a core

The concept of the reusable learning object (RLO) became widely known because of it's clear approach to describe 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. The ILR being an advanced type of LO has type interdisciplinary – it can be used in 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], the description of these and others approaches can be found in [3[, [4]). In simple case - information object type is defined by the kind of information that it is concerned: concept, fact, procedure, process, principle. The lowest level of learning object granularity in [3], presented by content assets (items) those classified by the type of media – text, audio, image, etc. and they are used for creating information objects. In the case of LOs based on software, authors bypass the topic of an information object or information resource (collections of information objects [5]) as a core of learning object and use concept GLO instead of it [14], [15], [16], [17]. I

B. OOP and GLO approach for LO

Some of the attempts to describe learning object (LO) through the concept of OOP was unsuccessful and [9] criticized them ([6], [7]) because they only bring complexity to the description of the LO concept and do not facilitate practical use of LO. Indeed [6] and [20] explain the application of OOP concept by reusing LO, and [8] also suggest to do it through metadata tags. The [7] is criticized for a poor description of the practical usage because it only states that LO as objects has methods (typically rendering and assessment) and properties (content and relationships to other resources).

Nevertheless, there are some more detailed descriptions how to use OOP concept together with the concept of RLO. The first step for clearness made in [10] explain how to look at learning object as at instance of OOP class, where

LearningObject is the superclass of all learning objects. The [10, 7] use term learning resource (this term is used instead of the information resource) as a core of LO. In [10] exist such properties as *Version* and *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. However, the paper provides only an example of a hierarchy of learning activities types strictly connected with LO. Thus, the methods of usage of the described principles in LO constructing are weakly represented.

The generative learning object (GLO) concept proposed by [18], [19] define GLO as: "an articulated and executable learning design that produces a class of learning objects." By [20] GLO is a special category of LOs with a higher degree of reuse and repurposing.

To use GLO (for making an instance of RLO), the context in which it will further be used, should be analyzed. The contexts declared in [13] are following thematic context, pedagogical context, learner context, organizational context and historical/statistical context. The didactical, organizational and technical contexts are used in [17]. Authors of [14,15,16] do research in the field of teaching Computers Science that is close to interdisciplinary software. The context of LOs in [14] is described by the TPACK (Technological Pedagogical Content Knowledge), it consists of 3 main domains: content, pedagogical and technological. Those domains are similar to contexts from [17]. These authors in [15] 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 type).

In conclusion of this part of literature review we add that the topics of OOP for LO and GLO are in the center of attention of many researchers, and exists the set of basic ideas on how to apply them for LO construction. The rules on how to apply OOP concept to LO does not exist, but there are some attempts. This could be because LO is simple and there was no reason to systemize it. In contrast, ILR is complex and principles of OOP are useful to describe it better. GLO concept has applicable principles and technologies, although there exist specific ones for computer science domain (that is close to our computational science domain) they cannot be simply used for ILR.

In this article, the scientific software, like modeling or simulation, serves as a source for ILR construction – from one scientific program (our information resource) many variations of information resources (IR) and ILRs may be formed. The principles of OOP and idea of GLO are used in new concept of description information resource formed from interdisciplinary software, to facilitate the process of creating ILR. In this paper we concentrate on the framework of information resource's description and take in account GLO concept and briefly describe application of OOP concept. The next purpose is to describe how to build ILR from this type of resource, like it is done for LO from five information object types in [12], this will be done in future works.

Our approach is useful for describing content of interdisciplinary courses of master programs for students with different academic background, for students whose role is between specialist in specific domain and programmers.

In section II presented core ideas of our approach: specific way of the description of the scientific software – generative information resource and IR, and our concept of ILR, our view on OOP aspects. In section III an example of scientific software is considered and some interesting details are provided.

II. Approach desciption

The scientific software can be applied to educational purposes in various disciplines by different ways. In this work we discuss three of possible user's roles: domain expert, system analytic and programmer. For applying this type of software in education a lot of efforts should be done, like choosing part or way of representation related to the topic of learning course, integrating it in learning approach, creating proper tasks, adaptation for the students etc. The solution is using RLO approach but scientific software is more complex then usual information object that is the core of usual RLO and we named it information resource (IR). As consequence the learning object created from IR is a learning resource that is interdisciplinary because of the feature of scientific software that integrates IT domain and problem specific domain. Thus we name it interdisciplinary learning resource (ILR).

The first step before creating ILR is to make a description of content of each available representation form of scientific software that serves as a generative information resource.

A. Interdisciplinary Software as an Information Resource

The information resource based on scientific software is characterized by forms of software representation and by domains: by related subjects belonging to the interdisciplinary domain of the problem solving by scientific software and by professional roles.

Several forms of software representation exist: in [16] learning resource (program for robot) three-way of LO presentation are used: code, graphic interface and meta-body for it (code for "drawing" graphical interface). Three coding type of software model (that define forms of appearance) considered in the article [17]: symbolic (source code, documents of API, development related text), drawing (UML diagrams) and pictures (screenshots of GUI and screen video of using software). All these forms concern Computer Science domain. Since our information resource is software, we use some of them or their analogies.

Main representation forms of scientific software are as follows: program as executable file, set of diagrams (UML, IDEF, etc.), presentation of source code as a workflow. Attention should be paid to the last form that allows understanding how the program or program complex works that lead to rising developer skills (just programmer is not highly demanded in interdisciplinary context). Workflow concept clearly shows structure of composite applications that fairly often used in science, especially when calculations are performed in the cloud and some packages have some data dependencies. For this purpose, we use the developed by our department platform CLAVIRE [21] that is very useful when distributed cloud computing is needed (if there are a lot of data or computations). In other cases, Microsoft Windows Workflow Foundation (WF) technology is enough. It allows describing structure of composite or standalone application and its components.

For describing scientific software by domains there is need to analyze the content of representation form. The content is described by keywords relating to specific subject (this relation may be set by expert or by domain ontology). Moreover, we suggest to mark the parts of scientific software in representation forms by these keywords and by related professional roles (in this research we discuss following roles: domain expert, system analytic and programmer). For set of diagrams and workflow representation forms it is easy to distinguish the parts: each diagram, each activity of WF or each package of CLAVIRE. For the executable form the parts are as follows: if graphical user interface exists these parts are just available to user actions, but often scientific software is without graphical user interface, in this case parts are distinguished by the sets of input parameters (problem domain IT domain specific).

Generative information resource (GIR) has two parameters that influence the content of IR in which way the scientific software will be used:

1) Available representations
2) Related domains: subjects (with keywords) and professional roles

GIR contain list of available way of interactions that are dependent on mentioned parameters. Each representation allows users to perform different typical action for example editing or creating new diagram, creating code based on set of diagrams, running executable solution with set of input parameters, adding or changing package/activity in workflow.

IR is an inherited subclass of GIR and have the attribute Version that is used to describe the combination of the type of representation form and the set of values of subject and role specific attributes. These possibilities to interact with information resource provided to users are redefined in IR in method "use" that is based on ways of usage representation forms that describes which parts of representation forms are used and how.

On Fig. 1 presented overview of proposed concepts of IR and GIR

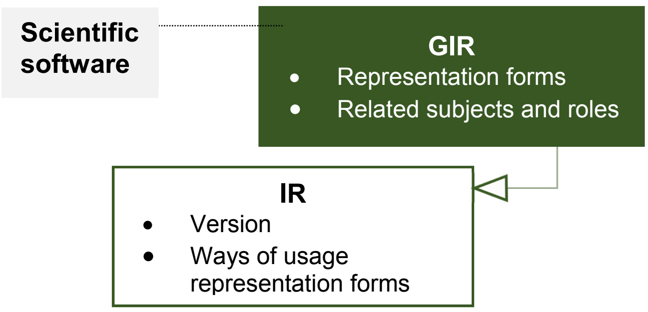

Fig. 1 Generation IR from scientific software

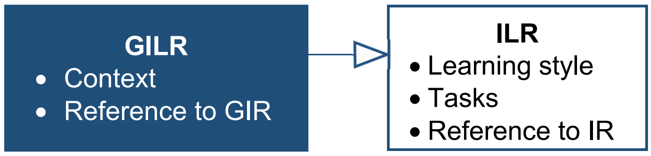

Fig. 2 Generation IR from scientific software

B. *Concept of Information Learning Resource*

GIRL is used to describe available options of the values of the contexts parameters for selected GIR. ILR has fixed values of contexts parameters and its content (tasks that form a learning process and an assessment) is defined by learning style that depends from parameters of student and teacher learning preferences and goals. The types of tasks are formulated inside GIRL for the chosen GIR. The specific tasks are defined in ILR based on described in IR's method "use" possible interaction. The tasks combined with learning style define the behavior of ILR's method "learn" (that available for students). The method "show" (for teacher) intended to demonstrate of scientific software in learning purposes and only depend on IR's method "use".

The superclass learning resource (*LR*) has an attribute *Contexts* that consists a set of three contexts – pedagogical, technical and content that contain contexts' parameters. These parameters influence the values of attributes *Representations*, *Subjects* of *IR*. To choose IR particular discipline and profession role should be defined as content parameters. The pedagogical parameters influence the learning approach and assessment way, while the technical parameters are connected with way of usage representation form.

The process of creating and application ILR is following:

1) Create GIR (from scientific software)
2) Set contexts parameters of GILR.
3) Find proper IR and define "use" method
4) Create an ILR
   a. Choose one of learning style
   b. Formulate the tasks for it
5) Use created ILR in class

Scientific software can be used for studying various subjects after it was described like GIR. IR created once can be used again for studying same subject but with other pedagogical parameters (e.g. for another group of students, for selected student with his specific learning parameters or by another teacher). Created ILR can be used in the same condition (set by context parameters values)

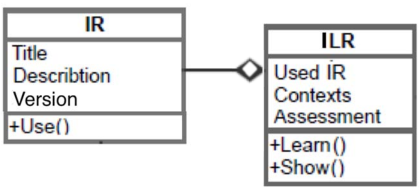

Fig. 3. IR class and ILR class

III. IMPLEMENTATION IN PRACTICE

We consider an example of GIR (based on interdisciplinary scientific software) that offer optimal path finding between bus stops and public bus scheduling. This object allows learning of at least the following topics: graph theory, vehicle routing problems, parallel computing, evolutionary computing, complex systems design, algorithm theory, discrete mathematical models, and urban transport system monitoring and control. The available roles are follows: programmer, analytic, domain specialist.

The IR consists of two parts: generating pairs of bus stops and searching shortest paths between pairs on a dynamic scheduling optimization part. The graphical user interface is presented at Fig. 6.

Package for public transport routes optimization takes files with following data: graph of roads with passing times data bounded to each edge, time moments, in which routes passing times change, information about routes and bus stops.

Public transport schedule optimization takes files with the following parameters: parameters of the routes' and public vehicles, passing time between two bus stops at certain moments, a density of passengers on the bus stops, genetic algorithm parameters, parameters for parallel application launch (number of nodes, cores on each node, etc.).

At Fig.4 presented VSO workflow representation in CLAVIRE that is better for analyzing changeable parameters. Fig. 5 depicts workflow representation in WF, which is good for changing algorithms or implementation of activities in the workflow.

The application (as ILR) can be launched in a variety of learning ways [22, 23] (different implementation of method "learn"):

- As a black box without examining the structure of application. Only input and output data can be observed.
- With access to the workflow, which allows editing the order of its elements.
- With access to each element

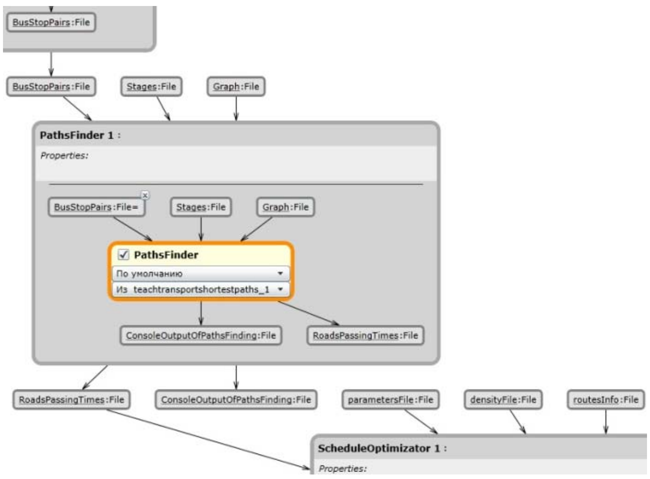

Fig. 4. VSO workflow representation of path finding and transport schedule optimization scientific software.

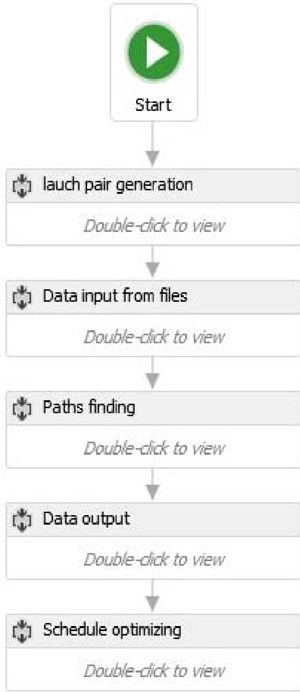

Fig. 5. Workflow representation in WF

On step of generating ILR the tasks will be created as follows:

- Varying genetic algorithm parameters to investigate the convergence of the solution.
- Changing input domain parameters for observation of data dependencies, for example, the relationship of time between buses and their capacity.
- Tuning parameters of launching optimization package on a cluster to reduce the calculation time.
- Rewriting or changing some parts of the code to speed up the package or eliminate bottlenecks

Only the most interesting part of this scientific software was presented in this section. The explanation of principles of creating content of ILR will be presented in future works.

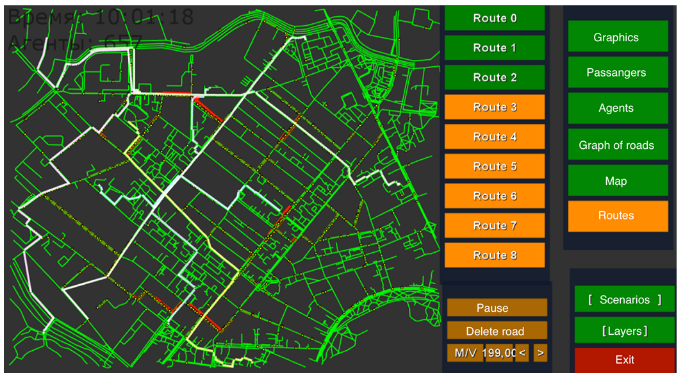

Fig. 6. GUI of the executable representation

IV. Conclusions And Future Works

Our approach is useful for instructors of interdisciplinary master programs where the students have a different academic background. The solution consists of applying principles GLO and RLO concepts together in the process of creating learning resource from scientific software. Thus, the new concept of GIR appears and IR becomes reusable. The separation information resource based on scientific software and learning resource allowed to reach the higher level of reusability through using the potential of the complexity of the problem domain that allows creating interdisciplinary learning resources (that combine specific role in IT, subject related for problem domain) for different disciplines. The OOP implementation together with GLO concept helps to unify description of learning materials of that helps to create the instrument for automated generation of ILR from interdisciplinary software that fulfills repository of the exclusive learning materials easier and faster than it will do an expert without such automation tool. Another feature of ILRs that they are based on the real scientific software that developed in scientific group thereby students are learning to work with it and get experience that will help them to continue research of scientific group and facilitate joying them.

In the future, we develop a system of relations between different context parameters and transfer the approach on specific types of interdisciplinary software, that will allow us precise features of ILRs. Another goal is to observe teaching approaches and learning styles to create a base of typical tasks variations depending on it.

Acknowledgment

This work was financially supported by the Government of the Russian Federation

References

[1] S. Currier, L.M. Campbell, "Evaluating learning resources for reusability: the" dner & learning objects" study," In ASCILITE, 2002 Dec, vol. 8, pp. 139-148.

[2] CISCO S. "Reusable learning object strategy: Designing and developing learning objects for multiple learning approaches," White paper, CISCO, 2003.

[3] A. Claudine, and K.M. Ezra. "Improving Learning Object Reuse Through OOD: A Theory of Learning Objects." Journal of Object Technology, vol. 9.6, 2010, pp. 51-75.

[4] K. Verbert and E. Duval, "Towards a global component architecture for learning objects: A comparative analysis of learning object content models," In Proceedings of the EDMEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 202-208. 2004.

[5] S. Currier and L.M. Campbell, (). Evaluating 5/99 content for reusability as learning objects. VINE: The journal of information and knowledge management systems, 2005,35(1/2)

[6] C. Quinn and S. Hobbs, "Learning objects and instruction components. ," Educational Technology & Society,vol. 3(2), pp.13-20, 2000.

[7] R. Robson, "Object-oriented Instructional Design and Applications to the Web." In EdMedia: World Conference on Educational Media and Technology, vol. 1999, no. 1, pp. 698-702, 1999.

[8] B. Bannan-Ritland, N. Dabbagh and K. Murphy, Learning object systems as constructivist learning environments: Related assumptions, theories, and applications. 2000.

[9] M. Sosteric and S. Hesemeier, "When is a Learning Object not an Object: A first step towards a theory of learning objects," The International Review of Research in Open and Distributed Learning, vol. 3(2), 2002

[10] P. Mohan and C. Brooks, "Engineering a future for web-based learning objects," InWeb Engineering 2003 Jul 14, pp. 120-123, Springer Berlin Heidelberg.

[11] C. Chrysostomos and G. Papadopoulos. "Applying Object-Oriented Principles to the Analysis and Design of Learning Objects," The European Conference on e-Learning, 2007.

[12] C. Barrit, 2001. CISCO Systems Reusable Learning Object Strategy-Designing Information and Learning Objects Through Concept, Fact Procedure, Process, and Principle Templates. *Version 4.0. White Paper*.

[13] P. Mohammed and P. Mohan, "Contextualizing learning objects using ontologies," Computational Intelligence. 2007 Aug 1;23(3):339-55.

[14] R. Burbaite and K. Bespalova, "Model-driven processes and tools to design GLO for CS education. InComputers in Education (SIIE), International Symposium on 2014 Nov 12, pp. 139-144, IEEE.

[15] R. Burbaite, K. Bespalova, R. Damasevicius andV. Stuikys, "Context Aware Generative Learning Objects for Teaching Computer Science," International Journal of Engineering Education. 2014 Jan 1;vol. 30(4), pp. 929-36, IEEE

[16] V. Stuikys, K. Bespalova and R. Burbaite, "Feature transformation-based computational model and tools for heterogeneous meta-program design," InComputational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on 2014 Nov 19, pp. 185-190.

[17] J. Magenheim and O. Scheel, "Using Learning Objects in an ICT-based Learning Environment," InProceedings of E-Learn 2004 Nov 1 (pp. 1375-1382).

[18] T.L. Boyle,. "To boldly GLO – towards the next generation of learning objects," World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004, pp. 28-33.

[19] R.L. Morales, "A case in the design of generative learning objects (GLOs): applied statistical methods," World Conference on Educational Multimedia, Hypermedia. 2005.

[20] C.B. Chirila, "A survey on generative learning objects," In The International Scientific Conference eLearning and Software for Education, vol. 2, p. 550. " Carol I" National Defence University, 2015.

[21] Konstantin V. Knyazkov, S. V. (2012). CLAVIRE: e-Science infrastructure for data-driven computing . Journal of Computational Science, 504-510.

[22] A. Dukhanov, M. Karpova, K. Bochenina, 'Design virtual learning labs for courses in computational science with use of cloud computing technologies," Procedia Computer Science. 2014 Dec 31; 29:2472-82.

[23] A. Dukhanov, T. Trofimenko, M. Karpova, L. Bezborodov, A. Bezgodov, A. Bilyatdinova, A. Lutsenko, "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. 2015 Dec 31; 51:1957-66.

[24] A. Holzinger, J. Smolle and G. Reibnegger Learning objects (LO): An object-oriented approach to manage e-learning content. Encyclopedia of Informatics in Healthcare & Biomedicine. Hershey (PA): Idea Group .Fererence. 2006:89-98.