Object-Oriented Approach for Quick Preparation of Learning Materials in Conditions of Digital Transformations and Rapid Updates of Scientific Results

Alexey Dukhanov  
National center of cognitive research  
ITMO University  
Saint Petersburg, Russia  
dukhanov@itmo.ru

Leonid Gorokhovatsky  
Faculty of Digital Transformation  
ITMO University  
Saint Petersburg, Russia  
leongor_us@yahoo.com

Aleksei Pershutkin  
Faculty of Digital Transformation  
ITMO University  
Saint Petersburg, Russia  
Alex_Neurocoder@itmo.ru

Sergei Koniukhov  
Faculty of Digital Transformation  
ITMO University  
Saint Petersburg, Russia  
svkoniukhov@itmo.ru

I. INTRODUCTION

Abstract—This paper examines the object-oriented approach (OOA) for annotation and defining of hybrid learning resources (HLR) based on existing information resources, and scientific object and learning object paradigms. This approach allows us to create, develop, and update learning resources rapidly. At the same time, the use of one existing resource in different applications (research/domain areas) becomes possible. The approach uses metamodeling based on reusable abstractions: reusable learning objects and research objects. We describe the usage of basic OOA concepts in the context of HLR and incorporate the domain-specific language AHLR-DL to show the possibilities of our approach.

Keywords—object-oriented approach, hybrid learning resources, information resources, reusable abstractions

Digital developments and transformations significantly influence educational processes, and scientific results and technologies emerge and update at a rapid pace. Contemporary educational trends lead to the appearance of two types of educational programs in engineering education. Programs of the first type are multidisciplinary educational programs based on cross-cutting technology and one branch (e.g., “Urban Supercomputing” [1] or “Big Data Financial Technologies” [2]). Programs of the second type are interdisciplinary programs containing the incorporation of two or more cross-cutting technologies and applications in several branches. Such programs provide many learning trajectories. For example, the program “Big Data and Machine Learning” [3] provides five basic learning trajectories (including Quantum Intelligence). Therefore, to provide students with the skills required according to educational programs of these types, teachers face designing and preparing a huge volume of learning resources. Moreover, the use of traditional approaches to design and implement learning resources is ineffective for major courses because materials become obsolete fast.

To mitigate this problem, we offer the object-oriented approach (OOA) for annotation and defining of hybrid learning resources (HLR) based on existing information resources, and scientific object and learning object paradigms. This approach allows us to create, develop, and update learning resources rapidly, and the use of one existing resource in different applications is possible. These applications depend on educational purposes (including psychological features of students), technical possibilities, and domain areas. This work-in-progress considers the basis of the approach and presents a test version of the domain-specific language (DSL) for a definition and description of HLR. We present an example of our HLR description using this language.

II. BACKGROUND AND RELATED WORKS

Our contribution is based on two paradigms: research objects [4], [5], and learning objects [6]–[10]. Both paradigms use the OOA to define informational resources and use tools to handle these resources.

The research object paradigm (ROP) gives us a technique to describe a set of links to scientific informational resources (such as computational application packages or libraries, scientific papers and reports, and technical documentation) and provides instructions or scripts to use these resources for the defined research purposes. Instructions can be presented in the form of usual text manuals or scripts prepared with the use of DSL (e.g., EasyFlow [11]). The ROP was extended in the Virtual Simulation Objects approach, which allows researchers to integrate research objects visually according to an anthology and build complex scientific informational tools [12].

The learning object paradigm (LOP) is similar to the ROP. The difference lies in the application’s purpose: education. Therefore, a learning object (or its extensions: Reusable Learning Object (RLO) [8] and Generative Learning Object (GLO) [13]) can contain manuals or tools to use related informational resources in instructional design. Each learning object contains information concerning learning prerequisites and learning objectives. These requirements are reflected in Educational Modeling Languages (EMLs) [14]. This chapter contains the high-level classification of EMLs, which allows us to formalize educational processes with the use of learning, human, managerial, and time resources. The usage of reusable abstractions (learning and software objects) in the building of
meta modeling languages is described in [15], and product design and copyrighted works — two critical resources — are taken into account in [16].

We hypothesize that contemporary international educational society is provided with a wide variety of approaches and means to formalize and build learning resources and courses. However, we could not find works devoted to joint application scientific results, which can change rapidly, and RLOs to build learning resources for interdisciplinary courses. Therefore, our work-in-progress is devoted to initiating research on the development of a meta language to annotate learning resources in applications of ICT, including cross-cutting technologies, in different research/domain areas. It could be interesting to implement dynamic binding and polymorphism to use one resource in different applications, including different domain areas.

III. OFFERED APPROACH

Our approach uses metamodeling based on reusable abstractions: RLOs and research objects [4]. We apply the OOA, especially dynamic binding and polymorphism, to annotate HLR. We use the notion «HLR» because we invoke both RLOs and information objects (including scientific application software, research objects). The OOA approach is based on three concepts: encapsulation, inheritance, and polymorphism.

A. Encapsulation

The annotation of learning resources starts with the declaration of the class with properties and methods. We called this class an Abstract Hybrid Learning Resource (AHLR). There are two types of properties: (1) parameters, which define educational and domain applications and system-technical possibilities of related software and (2) links to informational and learning resources. Educational parameters define instructional features of instances (e.g., educational degree, learning technology, representation). System-technological parameters depend on the technical possibilities method, approach (e.g., the maximum number of iterations for some numerical method). Domain parameters depend on domain area only (for example, initial number of infected agents in the SIR model, stock of raw materials in the production problem). Methods describe the use of the linked resources for educational purposes, including theoretical learning, practical learning, and evaluation of learning outcomes. Every instance of an AHLR (we consider HLR as one of the instances of this class) may describe an application of connected informational resources and learning objects to different types of learning activities and various domain areas. For example, we can create an HLR to deliver a lecture illustrating and explaining RLOs (to introduce basic concepts and terms), present PowerPoint slides, and experimental parts of scientific papers. The AHLR contains the set of links to these resources and RLOs. Here, RLOs, scientific papers, and PowerPoint slides present the theory of self-study. An application package is a tool to perform the computational part of an assessment. Note that we can create other instances of the AHLR for various domain areas using both the same and different resources depending on the domain problems or tasks.

To facilitate teachers’ efforts to describe an application of HLR, we defined structures named “Learning activities” to achieve educational purposes. Such structures contain a list of methods to realize them. Initially, an activity is based on the preliminarily prepared template (the initial list of methods), but a user may change this list if necessary. The list of activities and examples of the templates are presented in Part IV of this paper.

Since every AHLR and its instances only contain links to RLOs, informational resources, or ROs, any changes in related objects are automatically reflected in an AHLR. The author of an AHLR is required to be informed about them.

B. Inheritance

To fix the designed AHLR’s contents and create a new child class to extend the area of application, we use the inheritance concept. A child AHLR may contain overridden and new parameters or methods to change and supplement the application of related instances. For example, we may design the class for teaching the simplex method in general and create two child’s classes for teaching the same algorithm in financial and food domains (with each class complementing the new links to required informational and learning resources). In addition, the inheritance allows us to create a library of AHLRs according to an existing hierarchy, including scientific branches structure. For example, we may describe AHLRs devoted to optimization theory (OT). Suppose that a parent AHLR is defined in terms of present basic concepts and terms of the OT, including the context of optimization and the problem statement, affecting factors, boundaries, and optimum solution seeking. We can then create a child AHLR to provide students with skills in solving one-dimensional optimization problems with the use of critical points analysis. Next, we can devote the children to one-dimensional problems with unimodal, convex, or Lipschitzian objective functions.

C. Polymorphism

The use of polymorphism allows us to design learning resources to teach using general-purpose methods and technologies, including cross-cutting technologies. We design the base class with abstract methods. To provide students with skills in cross-cutting technology in general and the application of this technology in different research/domain areas, we create the child AHLR, which contains same-name methods with required executable contents. For example, we could design classes to provide students with skills in clustering. Let the base class contain the abstract method to present the related algorithm on the lecture (“Show”). In this case, we may create three children for general learning and study the algorithm’s application in financial and social domains. It can be observed that the method “Show” is defined in each child differently, including the usage of different RLOs, data sources, and application packages. The related links are defined with the use of dynamic binding.

D. Approach to Implementation

To annotate HLR, a user only has to have a text editor. The annotation of a parent AHLR takes 15–20 minutes in the case of an experienced user having the related JSON template, required context parameters, links to existing RLOs and ROs. Child AHLRs are described much faster (the mean 5–7 minutes for one) because, as a general rule, it only requires changes of some context parameters, activities, and methods, as well as adding links to extra RLOs, informational resources/objects, and ROs if necessary.
To access RLOs, we can use learning management systems based on standards that are widely known in educational society (e.g., SCORM, xAPI), Fig. 1. The access to basic informational objects and resources, like texts of scientific papers or technical reports, may be direct (a link is enough) or through related common services (e.g., IEEE Xplore, YouTube, Google Drive, etc.). However, usage of ROs, including application packages for (advanced) scientific computing, is more effective with the use of platforms/frameworks for the management of distributed computing in heterogeneous computational environments like Pegasus [17] and CLAVIRE [5]. Such powerful computational means are useful for research collaborations and allow us to grant students direct remote access to unique scientific services through the web interface. Scientific teams in collaboration with teachers can customize such services to learning requirements. It replaces the design and development of related executable learning resources. In addition, students faced real digital objects instead of related learning “translations.” To execute a scientific service, parameters for system-technical and domain applications may be used.

It is assumed that HLR will be generated on the basis of AHLR’s instructions with the use of xAPI-specification. This specification will allow us to collect learning experience data (including progress status, interim test results, etc.) and personal user data (e.g., including mouse cursor motions). These data will be a base for further pedagogical analysis.

IV. PEDAGOGICAL ASPECTS

Pedagogical aspects are the same, related to encapsulation, inheritance, and polymorphism. Encapsulation allows us to save a training object from unnecessary information, which is based on scoring, to separate the parameters of a training object, important for a particular user (student), from unimportant ones, and choose the one that is appropriate specifically to its educational goals and objectives. It is important that the same object can be used by another user (trainee) in the interests of realizing his, probably completely different, educational trajectory. Encapsulation simplifies the process of searching for information for the user and allows him to form the competency directly demanded by him at the moment.

Inheritance is a tool that allows us to create learning resources for each student performing a specific educational task in a team—for example, when performing a task by a group or when studying a common topic in one class. Each educational resource-child contains all the attributes of the parent resource (so each student has complete information about the general), but also contains the special features that are important for a particular student within the context of his specific personal features and task.

Polymorphism allows us to provide students from different professional contexts with identical content. For example, the same subject from mathematics can be presented from different points of view to a trainee-physicist, trainee-engineer, and trainee-programmer. This opportunity is especially important when we talk about the educational context of students with various profiles and specific educational needs (Fig. 2).

All of the above makes the learning process unique for each student through the individualization of educational trajectories. The important aspect is that all of the principles can be realized through using a special kit of definitions of the learning resource, which includes educational activities and methods (the methods’ example and the list activities are shown below). By correct annotation of the learning resource with suitable meanings of activities and methods, we can obtain the desired educational result in each particular educational case. We can simply change the settings if we find that the information, generated by the resource, suits the different activities and methods (for different educational needs).

Methods’ Examples: Show() - presentation of resources with explanation; QA_Session() - question and answer session (may be used as a quick check for materials consolidation or as assessment mean); Grant_Access() - access granting to RLOs/ROs/Information resources (objects) for students; Problem_Statement() - problem statement for student/students’ team before practical activity (assignments, projects, etc.); Consult() - consultation; Performing() - problem/task performance; Check() - check of student’s problem/task performance results; Review_Session() - problem/task performance results’ review (after action review); Reflection() - after action reflection (session, where students and teacher(s) present and discuss their impressions, opinions, feedbacks after learning actions; draw conclusions).

![Fig. 1. Options of access to RLOs, ROs, and information resources.](image1)

![Fig. 2. The example of usage HLR based on different child AHLRs for students with various profiles.](image2)
Activity examples: Theory Learning (Grant_Access(), Show(), QA_Session(), Reflection()); Practical Assignment (Problem_Statement(), Grant_Access(), Consult(), Performing(), Check(), QA_Session()); Project (Show(), Problem_Statement(), Grant_Access(), Consult(), Performing(), Check(), Defense(), Review_Session(), Reflection()).

Contents of activities and behavior of methods depend on parameters of AHLR/HLR.

V. DESCRIPTION OF DOMAIN-SPECIFIC LANGUAGE FOR ABSTRACT HYBRID LEARNING RESOURCES

To show the capabilities of our approach and present opportunities for application, we made the prerelease version of the DSL, using the JSON format of data description as a basis. The current version of the language, which we call AHLR-DL, splits an AHLR description into the following blocks:

1) general: contains the name of the AHLR, learning prerequisites (skills required to start learning; defining the name of the skill and its level, for example, understanding or application), and learning outcomes;
2) parents: link(s) to parent AHLR(s);
3) parameters: contains names of parameters (including educational, problem/domain, and system-technical types);
4) links to research objects, ROs, and RLOs;
5) students' personal information: contains parameters of individual trajectory in educational programs or courses and personal traits, including psychology features (for example, the parameters of Felder–Silverman's Learning Style Inventory (LSI) [18]);
6) methods: links to methods for linked resources presentation and usage; methods may be differently defined in one or more child AHLRs; and
7) activities: learning activities description in the form of a list (please, see Part IV).

The example given below summarizes the potential uses of AHLR-DL.

AHLR "Optimization (basics)":

```json
{ "General": { "Name": "Optimization", "Learning Prerequisites": [ { "LP Skill Name": "Problem Statement Analysis", "LP Skill Level": "Apply" }, { "LP Skill Name": "Numerical Methods' Usage", "LP Skill Level": "Apply" } ], "LP Array" }, { "LO Description": [ { "Name": "OptOneDimOF", "LP Description": [ { "Learning Prerequisites": [ ] } ], "LO Array" } ], "General": { "Name": "OptOneDimOF", "LP Description": [ { "Learning Prerequisites": [ ] } ] }, "LO Description" }, { "Parameters": [ ] }, { "Parents": [ ] }, { "Links": [ ] }, { "Resources": [ ] }, { "Personal Information": [ ] }, { "Methods": [ ] } } // Methods
```

AHLR "Optimization with one-dimensional objective functions":

```json
{ "General": { "Name": "OptOneDimOF", "Learning Prerequisites": [ ] }, "LP Description": [ { "Learning Prerequisites": [ ] } ], "LO Description" }, { "Parameters": [ ] }, { "Parents": [ { "Parent AHLR": "Optimization" } ] }, { "Links": [ ] }, { "Resources": [ ] }, { "Personal Information": [ ] }, { "Methods": [ ] } } // Methods
```

The example above shows two AHLRs: "Optimization (basics)" and "Optimization with one-dimensional objective functions". The second AHLR is the child of the first AHLR. It is possible to note that links to similarly named methods are different. Here, we can define objects based on the first AHLR but bound each of them with methods of the needed class (dynamic binding).

We annotated and used approximately 50 HLR (including cross-cutting technologies in machine learning), Statistical data show this quantity reduces the otherwise time-consuming process of learning material design. In addition, students obtain the experience to use source materials (such as scientific papers, technical reports, and scientific software).

VI. CONCLUSION AND FUTURE WORKS

In this paper, we offered the OOA for Object-Oriented Approach for Quick Preparation of Learning Materials in Conditions of Digital Transformations and Rapid Updates of Scientific Results. The approach integrates two paradigms: the scientific object paradigm and the LOP. This approach allows us to create, develop, and update HLR rapidly. In addition, the use of one existing resource in different applications is possible. We have shown the way to use basic concepts of OOA (encapsulation, inheritance, and polymorphism) in the context of HLR. The methods and activities were incorporated and shown to present pedagogical possibilities of usages of the OOA. By the way, we have presented the DSL AHLR-DL with AHLR’s examples, which reflects part of the possibilities of our approach.

We intend to design, implement, and fill the repository to store and test the use of our AHLR/HLR description in the learning process. We will develop special application software to generate HLR automatically based on the descriptions of AHLRs with the use of xAPI specifications. Thus, we will receive enough data to perform an experiment to hypothesis test: the speedup of learning resources’ preparation for individualized learning trajectories without loss of educational quality. After that, we intend to raise this application to a framework for educational purposes.

ACKNOWLEDGMENT

This work is financially supported by National center of cognitive research of ITMO University.
REFERENCES


