Aggregating of Learning Object Units Derived from a Generative Learning Object

Vytautas ŠTUIKYS

Software Engineering Department, Kaunas University of Technology
Studentų 50, LT-51368 Kaunas, Lithuania
e-mail: vytautas.stuikys@ktu.lt

Ilona BRAUKLYTĖ

Informatics Engineering Department, Klaipėda University
Bijūnų 17, Klaipėda, Lithuania
e-mail: ilona@ik.ku.lt

Abstract. Aggregating and sequencing of the content units is at the core of e-learning theories and standards. We discuss the aggregating/sequencing problems in the context of using generative learning objects (GLOs). Proposed by Boyle, Morales, Leeder in 2004, GLOs provide more capabilities, focus on quality issues, and introduce a solid basis for a marked improvement in productivity. We use meta-programming techniques to specify GLOs and then to automatically generate LO units on demand. Aggregating of the generated units to form a compound at a higher granularity level can be performed in various ways depending on the selected criteria or their trade-offs (e.g., complexity, granularity level, semantic density, time constraints, capabilities of modelling the learning process, etc.) that enable to evaluate units in advance. We describe aggregating as an internal sequencing of the content units derived from a GLO. Our contribution is a formal graph-based model to specify the problem when the variability of LO units is large. First we formulate the problem and consider properties of the proposed model; and then we analyze a case study, implementation capabilities, and evaluate the approach for e-learning.

Keywords: learning object (LO), generative learning object (GLO), granularity level of LO, aggregating model, sequencing model.

1. Introduction

In general, learning or teaching process consists of three interrelated elements: competence, activity/process and topic/content. The content is to be learned or delivered within the process which is restricted in time dimension and can be split in time-related phases (e.g., due to the physical and methodological reasons). On the other hand, the content itself is not a monolithic structure, rather it consists of pieces that are known in e-learning as learning objects (further LOs or LO) (McGreal, 2004; Northrup, 2007). One can image that LOs of a given topic are linked (usually implicitly) with learning/teaching phases within the process. LOs and aggregating/sequencing/packaging of the content are the

most essential terms through which we express the fundamental issues (i.e., pedagogy-related, psychology-related aspects) of the process: *what* is to be learnt/ taught and *in which way* (i.e., logical sequence) the process is to be delivered and organized. These terms and the processes they describe are at the focus of e-learning standards (e.g., LTSC, 2002; LOM, 2002; SCORM, 2004) and learning theories such as the one proposed by Wiley (2000a, 2000b).

Typically LOs are dealt with (or should be dealt with) from the *reuse perspective* because the reuse strategy can ensure a higher productivity, better quality and wider capabilities and effectiveness of use. Indeed today technology advances enable teachers and course designers to create the content in a variety of versions. Modifications, changes and adaptations of the content are common reuse activities. The need for adaptation increases with technology advances and expansion of the e-learning domain. If adaptations are done ad hoc, this may lead to the uncontrolled growth of similar versions causing additional difficulties in storing, sharing and reusing. If adaptations can be done automatically, we have a more powerful kind of reuse, called *generative reuse*. Recently Boyle, Morales, Leeder *et al.* (Boyle *et al.*, 2004; Morales, 2005) have proposed the concept of *generative learning objects* (GLOs), which is based on separating the learning design from the instantiation of the LO content and using templates as a generative technology. The approach provides more capabilities at a larger extent, focuses on quality issues, and introduces a solid basis for a marked improvement in productivity.

We have extended the capabilities of GLOs by introducing meta-programming, a more powerful generative technology (Štuikys and Damaševičius, 2008). Using this technology a designer can, at a larger extent, firstly to specify a family of the related LO units by describing various aspects (through the use of meta-data relevant to the topic, which support modifications of the content). And then, a user can automatically generate from the GLO specification either a concrete LO unit on demand depending on the meta-data values or the whole family (sub-family) of the LO units. Thus GLOs (a) enlarge the space of variants of the content to be delivered substantially, and (b) increase the role of aggregating and sequencing of the content at the larger extent.

The aim of the paper is to consider the aggregating and sequencing problems of the content units (i.e., LOs) within the given topic, which are derived from the specification of the GLO. Our contribution is a formalized graph-based model which brings a solid background for solving the aggregating/sequencing problem when the space of LO units is large. Sequencing is described as a partial case of the aggregating problem (that explains why we use the compound term "aggregating/sequencing" in the paper). The proposed model requires first to evaluate LO units quantitatively, and then it serves as a basis to construct and solve the problem. Furthermore, the model enables to optimize the solution in a delivery process (if criteria of the process are well-defined and an optimization technique is selected). To our knowledge, the aggregating/sequencing problem in the context of GLOs that are described using meta-programming techniques is considered for the first time in the e-learning literature.

The rest of the paper is organized as follows. Section 2 analyzes related works. Section 3 introduces the basic assumptions to the problem under consideration. Section 4

presents definitions of the terms that are important to understand the introduced model. Section 5 formulates the aggregating/sequencing problem. Section 6 describes some properties of the model. Section 7 provides a case study by series of examples in order to approve the theoretic statements of the approach. Section 8 summarizes and evaluates the results. Section 9 provides conclusions and outlines the future work.

2. Related Works

Since there is a wide spectrum of publications on LOs in the e-learning literature and we deal with a specific topic in the paper, we restrict ourselves with those LO-related publications that either outline the context of the topic or have direct links with the topic. As a result, we use the following categorization scheme of the analyzed sources: 1) general issues (context) related to LOs, such as definition, reuse, taxonomy and design; 2) GLO-related sources including their technological support; 3) Sequencing-related sources (pedagogy and technology related).

Stream 1. Taxonomy for definitions, characteristics and applications of LOs can be found in Rossano *et al*. (2005). Papers Nugent *et al*. (2005), Nortrup (2007) analyze the design, development and validation of learning objects. The paper Altun and Askar (2008) focuses on the granularity aspects and affirms that ontologies provide guideline for instruction designers to address the issue of granularization. Their model proposes a separation of learning expectations as concepts and skills based on their ontological relations. The paper Huddlestone and Pike (2005) proposes a four-tier model (strategic reuse, operational reuse, contextual reuse, structural reuse) for making reuse happen in practice. IEEE Standard for Learning Object Metadata is presented in (LTSC, 2002). The paper Memmel *et al.* (2007) discusses various approaches to LO oriented instructional design.

Stream 2. As it has been already stated, Boyle *et al.* Boyle *et al.* (2004), Morales *et al*. (2005) are pioneers of the GLO concept. The authors interpret this kind of the e-learning content as the next generation of learning objects because of their capabilities to ensure better quality and higher productivity (Boyle *et al.*, 2004). The other paper Boyle *et al.* (2008) provides a conceptual framework that can be used to understand, to author and to adapt GLOs. The authors argue that a major pedagogical issue is the desire of tutors to adapt and not merely reuse learning object. In this context, a technological support to specify and implement adaptations and modifications becomes extremely important. The paper Štuikys and Damaševičius (2008) suggests 1) specifying GLOs at a higher abstraction level using feature diagrams and 2) implementing GLOs using meta-programming techniques. One can learn more about feature diagrams in Schobbens *et al.*, (2006) and more about meta-programming techniques in Sheard (2001). Kramer (2009) considers a process for generating Interactive Learning Objects from configurable samples.

Stream 3. The Wiley's sequencing approach is connected to instructional design theories. Sequencing of LOs occurs at 3 levels: sequencing of work models, sequencing of case types, and sequencing of specific problems (Wiley, 2000a, 2000b). The basic principles are "from simplest LOs" to "more complex LOs" and LOs should be "sequenced

according to their level and type, and in order to promote transfer when possible". In SCORM®2004 a new book was added, titled "Sequencing and Navigation", which describes approaches to control the way the learner interacts with the learning object based on the IMS Simple Sequencing specifications (IMS PCK and SS, 2003). The paper Alvarez and Montesinos (2006) considers the sequencing problem at the tool level. In order to better understand our approach presented in this paper one can learn more about basics of the graph theory and sequencing-related algorithms from Cormen *et al.* (2001).

The related works should be considered as a context to our approach that is described through assumptions, definition of basic terms, problem and model statement, analysis of properties of the aggregating/sequencing model and a case study.

3. Assumptions

The assumptions relate to the properties of GLOs, which are due to the use of heterogeneous meta-programming techniques. They are as follows:

1. The given GLO is designed for reuse in a much wider context than it is needed for a concrete application. In other words, GLOs describe the space of relative units of learning for possible case uses.
2. The content is highly structured into related pieces called *LO units* (i.e., *units of learning*) that are generated on demand through the specification of meta-data values by the user. More specifically, structuring of LOs is also known as a granularization in e-learning. We assume that units of learning are derived from a GLO.
3. It is possible to re-generate units of learning with new aspects (meta-data values), if the GLO specification is complete. Otherwise, the GLO is to be extended or re-designed.
4. It is possible to prune the generated LO units or even to introduce new variants of units externally (i.e., from other sources).
5. In general, a LO unit (either it was derived from the GLO or was introduced externally) has a value which may be considered from different views (social, pedagogical, topic-related, etc.). Even if a user (i.e., teacher or learner) do not estimate the value in advance explicitly, he/she uses some estimates implicitly (e.g., user always estimates the time slot needed for delivering or learning).
6. In general, aggregating/sequencing of the content is undependable of the way in which the LOs are created and upon the space of their units. But when the space of the content variants enlarges (as it is in the case of using GLOs), the problem of aggregating/sequencing becomes more complicated and the necessity to deal with it increases dramatically. The more complicated problem the greater need for its formalization and automation is.
7. Our intention is to focus on technology-based theoretical aspects of aggregating/sequencing problem; however, pedagogy-based aspects are seen as fundamental requirements though they are not always presented explicitly in the paper.

4. Definitions

Here we define the terms used in our approach. Since it is difficult to define two basic terms, i.e., Generative Learning Object (GLO) and LO unit only from the one perspective precisely, we use a multidimensional scheme for the definition of the terms here. This scheme has already been applied in Štuikys *et al.* (2009). Note that the terms LO and LO unit are treated as synonymous throughout the paper.

DEFINITION 1. In general, Learning Object is defined as small, stand-alone, mediated unit of a content that can be reused in multiple instructional contexts, serving as building blocks to develop higher-level compounds (e.g., lessons, modules, etc.) (Nugent *et al*., 2006). Wiley (2000a) defines a learning object clearly as "any digital resource that can be reused to support learning". When reused, such units are combined in various ways leading to the great variability of the learning content.

DEFINITION 2. From the technological perspective, GLO is a higher-level program (meta-program), i.e., an executable specification developed using some generative technology. In that aspect, GLO can be also conceived as a program generator allowing generating LO units on demand (meta-designer's view).

DEFINITION 2A. From the structural viewpoint, GLO is a compound (also can be called meta-program or meta-specification) that consists of two interrelated parts: meta-data (meta-interface) and meta-body. Meta-data are for specifying the meta-parameter values to support generation (this semantics of the term "meta-data" differs from the one used in e-learning literature, where the term describes storing, searching and sharing aspects of LOs). Meta-body describes the content variants dependent on meta-parameter values (see Section 7, for examples).

DEFINITION 2B. From the instructor's (teacher's) and learner's viewpoint (pedagogical perspective), GLO is a set of related LO units (or a qualified in somewhat way sequence of LO units).

DEFINITION 2C. From the methodological viewpoint, GLO is a highly reusable structure that enables ensuring higher productivity and quality of LOs, thus focusing not only on component-based reuse but also moving methodological efforts in the e-learning domain towards generative reuse, which focuses on variability of the domain and generation.

DEFINITION 2D. From the e-learning perspective, GLO is an extension of the LO concept, the fundamental concept of e-learning, in the technological, methodological and pedagogical aspects, thus bringing new capabilities (e.g., in terms of higher productivity and quality for e-learning and also rising new challenges) that should be understood and studied.

DEFINITION 3. LO unit is a concrete LO that is coded within the GLO specification in a specific way using some technology (e.g., meta-programming).

DEFINITION 3A. *Derivative LO* is a learning unit derived from the given GLO through the generating process that is supported by a given generative technology.

DEFINITION 3B. *Model* of the derivative LO is a set of (meta-)parameter values which are extracted from the pre-specified meta-data (meta-interface) of the GLO in such a manner that for each meta-parameter there is identified the only one its value. Since the set of parameter values describes the functionality of the LO very abstractly without details about the content, the model is treated as a high-level model.

DEFINITION 4. *Set of derivative* LOs (further it is denoted as a set S) is a set of LO units generated from the given GLO and then selected for a particular context of use. The *selection* is needed because, in general, the GLO may specify a much wider space of LO units that are needed for the given context.

DEFINITION 5. Aggregating of learning objects through *sequencing* is an arrangement of the learning units within the selected derivative set S according to some relation order between any pair of LO units (further we treat the selected derivative set simply as a set S).

DEFINITION 6. *Relation order* among two objects L_i and L_j is a propositional relation with the values *true* (or 1) if the objects are semantically linked and, from the pedagogical view, should appear one after another, and with the value *false* (or 0), otherwise (see also Definition 10).

DEFINITION 7. Role of a LO is identified as a property of the LO, which is important to the external system or actor to express some particular aspects of using the LO. Two LO units among the selected set of LOs have specific roles. These LOs are called as *initial LO* and *terminal LO*, adequately.

DEFINITION 8. *Initial LO* L_O ($L_O \in S$) is the one which is to be shown (learnt) first in the sequence S within the learning process that per se is identified as a manageable (by an external actor, i.e., learner, teacher, or learning system) sequence of LOs. The role of the initial LO can be changed in the concrete context of use.

DEFINITION 9. *Terminal LO* L_T ($L_T \in S$) is the one which is to be shown (learnt) at the end of the sequence of LOs within the learning process (or its phase). For example, terminal LO L_T can be understood as a process of providing the summary or conclusions of a topic. The role of the terminal LO can be changed in the concrete context of use.

DEFINITION 10. Two LOs L_i, L_j are called to be *directly ordered* in a given sequence S ($L_i, L_j \in S$) if there is such a logical relation that object L_j appears in the sequence only after object L_i, otherwise they are *not directly ordered* or they have not a logical sequence. In terms of the propositional logic, two *directly ordered* LOs L_i, L_j have a relation $R(L_i, L_j) = $ *true*, otherwise $R(L_i, L_j) = $ *false*.

DEFINITION 11. Two LOs L_i, L_j are called to be *mutually ordered* in a given sequence S ($L_i, L_j \in S$) if there is such a logical relation that object L_j appears in the sequence after object L_i, and object L_i appears in the sequence after object L_j. In terms of the propositional logic, *two mutually ordered* LOs L_i, L_j have a relation $R(L_i, L_j) = R(L_j, L_i) = true$.

DEFINITION 12. Weight w_i^k of LO L_i, which is denoted as $w_i^k(L_i)$, is a measure for estimating and expressing some quantitative characteristics of the given LO L_i, where index k indicates a characteristic type from the set P ($k \in P$) of the given attributes (e.g., granularity level, semantic density, interactivity level, etc. (LTSC, 2002)).

DEFINITION 13. *Aggregating/sequencing model* of the given set S of LOs is a directed graph $G(V, U)$, where the set of nodes V represents a member of S, i.e., $\forall_{i,j}(v_i \equiv L_i \& v_j \equiv L_j)(L_i, L_j \in S)$, and the set of directed branches $U(u_{ij} = (v_i, v_j); v_i, v_j \in V; u_{ij} \in U)$ represents the ordering relationship, i.e., the branch exits only if two LOs L_i, L_j either are *directly ordered or are mutually ordered*. (It is important to note that mutual ordering is introduced for generalization purposes admitting its value in other context, but not in this paper).

DEFINITION 14. Aggregating/sequencing model $G(V, U)$ is said to be *weighted* if its node v_i ($v_i \in V$) is marked by the weight w_i^k. The node-weighted sequencing model is denoted as $G(V^w, U)$.

We identify two kinds of aggregating/sequencing model: the node-weighted graph $G(V^w, U)$ and the arc (branch)-weighted graph $G(V, U^w)$.

DEFINITION 15. *Aggregate* is a set of LO units that lay on the route from the initial LO to the terminal LO.

DEFINITION 16. *Higher-level granularity aggregate* is a compound of a lower-level aggregates that are formed through the concatenation of two or more different routes from L_O to L_T within the model.

Equivalence property of two models $G(V^w, U)$ **and** $G(V, U^w)$. Let be given the initial graph $G(V^w, U)$ without loops (see Fig. 1a). Next, we perform the following transformations of the graph $G(V^w, U)$: 1) we draw the arc from the node L_T to the node $L_O(L_O, L_T \in S)$; 2) we form the model $G(V, U^w)$ by labelling branches with weights as shown (see Fig. 1b). Now we can proof the following statement.

Statement. *Two models* $G(V^w, U)$ *and* $G(V, U^w)$ *are equivalent against some evaluation function that evaluates weighted routes.*

The proof is demonstrated by an arbitrary model (see Fig. 1). Indeed it is easy to construct 1) all possible routes from L_O to L_T that are evaluated by the sum of node weights and 2) all possible arc-weighted loops that starts and ends at the node L_O. And
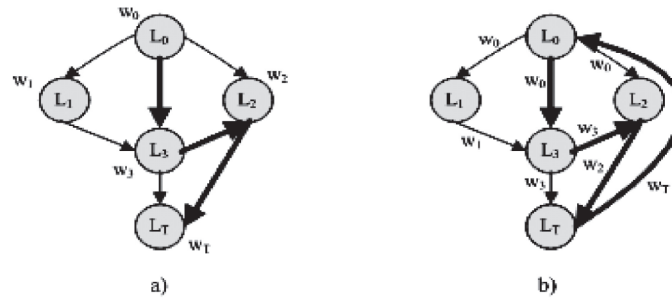
Fig. 1. Formal model of a delivery process that is modelled by the route from L_O to L_T (*a*) and by the loop that starts and ends at the node L_O (*b*).

then, by comparing evaluating functions, one can be convinced that they are equal against the function value ($w_0 + w_3 + w_2 + w_T$, see the bolded route and loop in Fig. 1a and 1b, respectively).

Now it is possible to formulate the aggregating/sequencing problem precisely as it follows below.

5. Problem Statement

Given the connectivity matrix C of a weighted model $G(V^w, U)/G(V, U^w)$, which is derived from the set S (the latter is derived from the given generative LO (GLO)). Two nodes in the model, v_O and v_T ($v_O, v_T \in V$), are identified as initial and terminal nodes respectively, where $v_O \equiv L_O$, $v_T \equiv L_T$ ($L_O, L_T \in S$). If the model $G(V, U^w)$ is used then we assume that there is the arc connecting v_T with v_O. Then the problem is formulated as follows.

*To find such a **route** D from the node v_O to the node v_T in the model $G(V^w, U)$ or (find such a **loop** D that begins and ends at the node v_O in the model $G(V, U^w)$) that satisfies the following conditions*:

1) $x_{ij} = 1$, *if and only if the path goes from the node v_i to the node v_j ($v_i, v_j \in V$), otherwise $x_{ij} = 0$*;
2) $D = \mathbf{opt}(\sum_i \sum_j w_{ij}^k c_{ij} x_{ij})$ ($k \in P$, $k = 1, 2, \ldots, |P|$), *where P is a set of weights that identifies the value of LOs; $C = \|c_{ij}\|_{n \times n}$, where C is the asymmetric connectivity matrix of the graph $G(V, U)$, $\forall_{i \neq j} c_{ij} = 1$, if and only if there is a (directed) branch from the node v_i to the node v_j ($v_i, v_j \in V$); otherwise $c_{ij} = 0$; also $\forall_i c_{ii} = 0$, $n = |V|$ (n – number of nodes)*.
3) *In the model $G(V^w, U)$ there $\exists u_{Oj} \& \exists u_{jT}$ such that $x_{Oj} = 1$ and $x_{jT} = 1$ $\forall j$ ($O < j < T$).*
4) $\forall_{k,i} (w_i^k \geqslant 0)$ *(if the node-weighted model is used).*
5) $\forall_{k,i,j} (w_{ij}^k \geqslant 0)$ *(if the arc-weighted model is used).*

6) $F(w_{ij}^k) \leqslant F_0$, *where $F(w_{ij}^k)$ is a constraint function dependent on weights w_{ij}^k; and F_0 is the fixed constraint value for the function (e.g., the duration of an academic hour if the time is selected as a constraint argument, etc.).*

Reader should not be confused by two slightly different notations used: w_i^k and w_{ij}^k. The first means the weight of the node L_i, while the second means the weight of an output branch (arc) from the node L_i to the node L_j. See also the numerical example in Section 7 (Example 6, for more details).

6. Properties of Aggregating/Sequencing Model

The model introduced in the paper expresses specific aspects of LOs that are slightly different from those LOs that are not derived from the given GLO but are composed in some other way, for example, in *ad hoc* manner. Of course, some properties of the sequence are orthogonal and do not depend on the way in which they were created. The aim of the Section is to identify the specific properties as well as common properties of series of LO units used in some teaching or learning environment. The emphasis is given to semantic of using the whole series of LOs (i.e., an aggregate) rather than a particular unit.

1. Since the model is based on the use of GLO, some properties of LOs within the model are inherited from the GLO. Namely, the model describes the features of LOs units that are highly interrelated and have a low granularity level (i.e., the units are small objects). Next the cardinality number of the model, that is, the number of vertex $|V|$ of the graph $G(V^w, U)$ may vary in a wide extent because of the GLO can be seen as an evolving entity and new features can be easily added and then a new set of units can be automatically generated.

2. The aggregating/sequencing model describes a family of the selected LO units that are arranged according to the needs and intention of a teacher/course designer or a learner. More precisely, the model identifies the possible sequences of the related learning objects; a particular sequence as an aggregate can be identified and then can be used in a given context. The whole model describes a variety of possible variants (i.e., the whole package) for reuse meaning a wide scope of use.

3. A LO derived from the model have three important properties: *the role, the value, and the order* (or a position within the formed sequence). The properties are interrelated in somewhat way. For example, the *initial LO* is the first, and the *terminal LO* is the last in the sequenced list of the LOs described by the model.

4. The role of the *initial* and the *terminal* LOs is to specify the beginning and the end of a content delivery process because any well-defined process should first be instantiated, and then, after some time should be terminated. But more importantly, these objects may also be responsible for decision making procedures.

5. The *initial* and the *terminal* LOs can be introduced in the model, when the model is created, in two ways: either these are generated from the given GLO directly (if such a feature was anticipated in the GLO design specification), or it is introduced by an actor externally (if such a feature was missed when the GLO was designed).

6. The initial node v_O and terminal node v_T have the following property: $j = 1 \equiv j = v_O \& j = n \equiv j = v_T$, where $j = 1, 2, \ldots, n$.

7. The *role* of the initial LO is to specify such a common data as the topic title, the aim of the topic, the capabilities or possible variants of the sequence. The *role* of the terminal LO is either to summarize the essential result of learning/teaching, to present some conclusions or to describe some states for the management action (e.g., to initiate the repeat of process from the beginning).

8. The value of a LO within the model is estimated by the weight. In general, the weight is *a vector* which coordinates may identify various attributes such as time needed to present, size (granularity level), complexity of the LO, readiness or level to which the given object supports the paradigm 'learning by doing', etc. By 'learning by doing' we mean modelling capabilities (e.g., push a bottom and record some result). Some attributes, such as complexity, may differ from the learner's and teacher's viewpoints. Thus the weight $\overrightarrow{W} = \{w_i^k(L_i)\}$, $k \in K$, where K is a set of pre-defined attributes of LOs.

9. The aggregating/sequencing model, i.e., either the graph $G(V^w, U)$ or $G(V, U^w)$ is said to be correctly constructed if the following conditions are valid:

   a) graph G is connected;
   b) two nodes $v_O, v_T \in V$ are identified as initial and terminal ones, where $v_O \equiv L_O, v_T \equiv L_T$ $(L_O, L_T \in S; v_O, v_T \in V)$;
   c) nodes/arcs are weighted and coordinates of the weight vector for each node/arc is ordered according to the same ordering rule (criteria);
   d) there are no mutually ordered nodes (in order to avoid internal cycles, see Definition 11).

10. Arcs, i.e., oriented branches of the graph, which pre-specify possible routes in the model, are introduced by the course designer keeping in mind learning/teaching theories (Wiley, 2000a). But it should be also taken into account that ordering of the graph highly depends on the context, i.e., on the internal structure of the content to be designed.

11. The weights of the nodes satisfy the following conditions (here we consider the node-weight model):

    a) $\overrightarrow{W} = \{w_i^k(L_i)\}, k \in K$;
    b) $\forall_{i,k}\{0 \leqslant w_i^k(L_i) \leqslant w_{i_o}^{k_o}\}$, $w_{i_o}^{k_o} \geqslant 0$, i.e., some coordinates may be zero; where $w_{i_o}^{k_o}$ is a pre-defined value specific for each LO (node) and for each coordinate of the LO (node), $k \geqslant 1$ (i.e., as a partial case the weight may have the only one coordinate).

12. Weights are calculated according to the pre-specified rules for each criterion (e.g., time, complexity, semantic density, size, readiness to implement paradigm 'learning by doing', etc.). Some of them may be very simple. For example, if a given LO supports the paradigm 'learning by doing' (it is usually clear from the context, say, when it is needed to solve a problem) then the coordinate is equal to 1, otherwise it is equal to 0. More specifically, the calculation can be seen as a *planning task* of the content delivery (LTSC, 2002).

13. The route in the model from the initial vertex (node) v_O to the terminal node v_T is estimated by the same coordinate at a time. The sum of the coordinates (i.e., the sum of the selected weights of vertexes in the route) is a function whose value is to be either minimized or maximized depending on the context. This depends on the selected coordinate or the intention of the teacher/learner. Thus the multiple routes can be identified, each expressing different aspects of learning.

14. Formally the optimized route (or loop) (i.e., the aggregate) in the given model models a learning or teaching sub-process in terms of content delivery. A set of optimized routes (loops) (i.e., compound of aggregates) models the whole learning or teaching delivery process at a higher abstraction (granularity) level, thus creating possibilities to automate the process.

15. The identification of routes (loops) in the aggregating/sequencing model can be also viewed as a weight-based integrating process of combining the smallest LOs (in other words, derivatives from the given GLO) into a higher-level compound (e.g., the content of a lecture or the other form of teaching) according to the pre-specified rules relevant to the given context of learning or teaching.

16. Some properties of the aggregating/sequencing model (e.g., a number of units for the use in a concrete context, weights of units or partially an order for the formation of sequences) may be obtained from the specification of the given GLO if such features have been anticipated in the design phase.

7. A Case Study to Illustrate and Approve the Introduced Theoretical Statements

Let be given the initial specification of a GLO (in our case "Sorting algorithms"), which is described using the notation of feature diagrams (see Fig. 2 and Štuikys and Damaševičius (2008), for more details). The model specifies at the high abstraction level (i.e., in the human readable form) the following aspects of GLO: scope, commonality and variability; essential features of the topic, their relationships and constraints. Note that not all constraints are depicted in the model (for simplicity reasons). For example, to explain the principles of sorting it is needed to use the random arrangement of arrays (*Principles* <require> *Random),* while teaching/learning effectiveness of the algorithms require all kinds of population (there is no constraints).

The aim is to explain how the theoretical statements work in the context of their use in some virtual setting. The case study provides analysis of examples. As examples are illustrative we try to simplify the problem when describing its implementation. Thus not all constraints of the initial specification (Fig. 2) are implemented in the examples. But examples we present here cover all basic concepts and processes introduced in the previous Sections. For instance,

EXAMPLE 1 (see Fig. 3) explains the technological implementation of the given specification (Fig. 2). The structure of the GLO 'Sorting algorithms' consists of meta-data (meta-interface, Fig. 3a) and meta-body (Fig. 3b). The latter is implemented using: a) **Open PROMOL** as a meta-language (Štuikys *et al*., 2002) to describe the variability
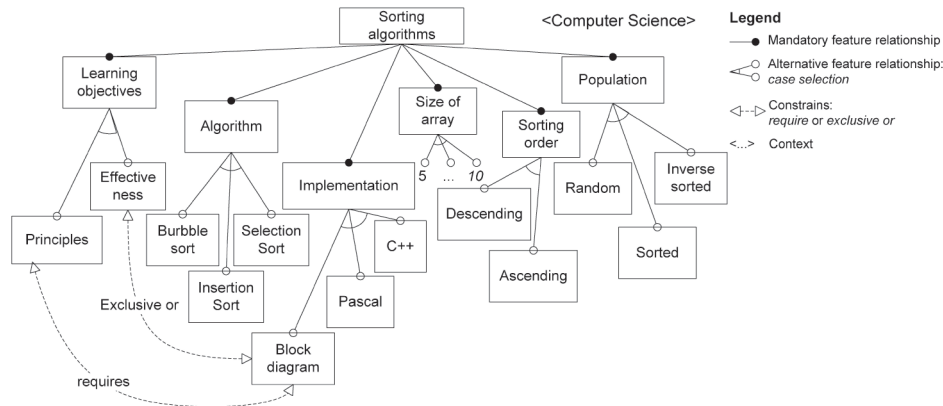
Fig. 2. Feature-based model of GLO "Sorting algorithms" (taken from a course of Computer Science).

and b) **JavaScript/HTML** to describe the visibility of LO units (while generated) via Internet. The implementation details, however, are hidden for simplicity reasons (i.e., the specification is presented as a *black–box* model without implementation details). Using the specification, one can generate about 648 derivative LO units ($648 = 2 \cdot 3 \cdot 3 \cdot 6 \cdot 2 \cdot 3$, where numbers are variants for each meta-parameter, see Fig. 3a; note that this calculation is made ignoring the constraint: "*Principles* <require> *Block-diagram*" because the constraint is not implemented in the meta-interface in Fig. 3). Such a large number of LO units tells us that the specification of the GLO was designed for reuse in a wide context which overpasses the needs of a concrete context.

EXAMPLE 2 explains an adaptation of the initial GLO for a narrowed context of use. One can understand the narrowing procedure by comparing meta-interfaces (cf. Fig. 3a and Fig. 4). Note that the specification (Fig. 4) has been changed by adding meta-parameters for generating the beginning (i.e., the initial LO) and the end (i.e., the terminal LO) of a LO unit.

EXAMPLE 3 presents a model of a derivative LO derived from the simplified GLO (see Fig. 3). The model is described as a set of all meta-parameters with one value for each meta-parameter: {EXAMPLE 3: begin := 1; goal := 1; algorithm := 1; language := 1; size := 5; order := 1; pop := 1; end := 1;}. A set of such models (see EXAMPLE 3) enables to specify an aggregate of LO or the pre-specified routes in the given graph, as it will be explained later.

EXAMPLE 4 (see Fig. 5a) gives the content of the unit the model of which is described by Example 3. The first LO unit (note that it has the internal structural units named by bolded titles, see Fig. 5a) demonstrates the following sequencing theory aspects: 1) formulation of learning objectives; 2) a more simple item is to be learnt first (e.g., principles of sorting should be learnt first because of a) sequencing logic and b) simpler array, simpler language, i.e., Pascal is simpler than C++, and c) there is no capabilities for 'learning by doing'; next unit (see Fig. 5b) should be learnt after the first since it is more complex

```
$
"Identify the learning objective"   {1- principles, 2- effectiveness}        goal:=2;
"Select an algorithm"               {1- Bubble Sort,
                                     2- Insertion Sort,
                                     3- Selection Sort}                      algorithm:=2;
"Select an implementation language" {1-diagram,2- Pascal, 3- C++}           language:=1;
"Size of array for a demonstration" {5..10}                                  size:=9;
"Select a sorting order"            {1- descending, 2- ascending}            order:=2;
"Select array population method"    {1- random, 2- sorted, 3- inverse sorted} pop:=1;
$
```

a)

```
@-   Meta-body implements the content of units using meta-constructs of
@-   a meta-language such as Open PROMOL (Štuikys et al.,2002); meta-constructs use
@-   meta-parameter values to generate/derive a particular unit;
@-   Meta-body is not shown in details because of complexity of the full specification
```

b)

Fig. 3. Meta-interface (a) and meta-body (b) of GLO "Sorting algorithms".

```
$
"Identify the beginning of a LO"    {1}                                      begin:=1;
"Identify the learning objective"   {1- principles, 2- effectiveness}        goal:=1;
"Select an algorithm"               {1- Bubble Sort,
                                     2- Insertion Sort}
                                                                             algorithm:=1;
"Select an implementation language" {1- Pascal}                             language:=1;
"Size of array for a demonstration" {5,10}                                   size:=5;
"Select a sorting order"            {1- descending}                          order:=1;
"Select array population method"    {1- random, 2- sorted }                  pop:=1;
"Identify the end of a LO"          {1}                                      end:=1;
```

Fig. 4. Meta-interface of a simplified GLO "Sorting algorithms".

(e.g., it contains learning of effectiveness of the algorithm with the capabilities 'learning by doing'). For example, two buttons '**Generate**', '**Bubble sort**' (see Fig. 5b) enable the user to model the sorting process, to write the sorting efficiency values, to compare different values for different arrays, and learn about efficiency that follows from the users actions.

NOTE 1. Though learning of sorting principles requires of using the block-diagram (see Fig. 2), the implementation in Fig. 5a is described by showing the *Pascal* program. This is done for simplicity (in order to save space in Fig. 5a and to reduce the number of units in Table 1 (also in Fig. 6)).

EXAMPLE 5 (see Table 1) enumerates LO units, their model, and weights for two criteria. It is assumed that the initial and terminal LOs, i.e., $L_O, L_T \in S$ were introduced externally, but not from the meta-specification (see Fig. 4). It is why we have 18 LO units, but not 16, as it follow from Fig. 4. Note that the initial and terminal LOs identify the beginning and the end of a sequence (aggregate) but not the beginning and the end of a concrete unit (in this implementation, we have introduced the beginning and the end within the LO unit for clearness). The numerical figures (*time in minutes*) are illustrative. Their calculation depends on the social context, i.e. depends on the teacher's intention, students' previous knowledge of the topic, etc. The other weight ('*learning by doing*')

a)

b)

Fig. 5. Derivative LO units: (a) that has no support for 'learning by doing' and (b) that has a support for 'learning by doing'.

is evaluated using the simplest form (i.e., Y/N, or 1/0). In the other context, one can estimate those capabilities using a more complex measure, such as a *level* (i.e., the extent expressed by a wider numerical range).

NOTE 2. L_1 corresponds to the LO unit in Fig. 5a. L_9 corresponds to the LO unit given in Fig. 5b.

Table 1

A matrix of weights calculated to evaluate the set of derivative LOs

Seq. #	LO's notation in G	LO model expressed through meta-parameters	Weights of the derivative LO units		
			Time in minutes	"Learning by doing" (Y/N)	Representation $\{w_{ij}^1, w_{ij}^2\}$
1	L_O	–	0	0	{0, 0}
2	L_1	{1;1;1;1;5;1;1;1}	15	0	{15,0}
3	L_2	{1;1;1;1;5;1;2;1}	10	0	{10,0}
4	L_3	{1;1;1;1;10;1;1}	20	0	{20,0}
5	L_4	{1;1;1;1;10;1;2;1}	10	0	{10,0}
6	L_5	{1;1;2;1;5;1;1;1}	15	0	{15,0}
7	L_6	{1;1;2;1;5;1;2;1}	10	0	{10,0}
8	L_7	{1;1;2;1;10;1;1;1}	20	0	{20,0}
9	L_8	{1;1;2;1;10;1;2;1}	10	0	{10,0}
10	L_9	{1;2;1;1;5;1;1;1}	7	1	{7,1}
11	L_{10}	{1;2;1;1;5;1;2;1}	5	1	{5,1}
12	L_{11}	{1;2;1;1;10;1;1;1}	10	1	{10,1}
13	L_{12}	{1;2;1;1;10;1;2;1}	7	1	{7,1}
14	L_{13}	{1;2;2;1;5;1;1;1}	7	1	{7,1}
15	L_{14}	{1;2;2;1;5;1;2;1}	5	1	{5,1}
16	L_{15}	{1;2;2;1;10;1;1;1}	10	1	{10,1}
17	L_{16}	{1;2;2;1;10;1;2;1}	7	1	{15,0}
18	L_T	–	0	0	{ 0, 0}

EXAMPLE 6 (see Fig. 6) presents the aggregating/sequencing model that corresponds to the set S of derivative LOs units ($|S| = 2*2*1*2*1*2+2 = 18$, see Fig. 5 and Table 1). For simplicity reasons, weights have the only two coordinates. We use the node-weighted graph for simplicity reasons (see Fig. 1). In general, the ordering relation, i.e., the directed arcs should be identified according to a learning theory. But a designer of the content can easily change the ordering relation (i.e., he/she can decide what arcs are to be introduced as being the most relevant to his/her context). Thus the ordering within the model by directed arcs (see Fig. 6) should be treated as illustrative. Nevertheless it expresses our view for the delivery of the topic in some virtual setting.

Using the model various optimization tasks can be considered and solved, e.g., those as follow: 1) to minimize the delivery time for explaining principles of sorting algorithms; 2) to maximize the capabilities for 'learning by doing' in learning sorting algorithms, etc. For instance, a solution of task 1 is: $L_0 + L_1 + L_T = 0 + 15 + 0 = 15$. A solution of task 2 is: $L_0 + L_5 + L_{14} + L_{13} + L_T = 0 + 0 + 1 + 1 + 0 = 2$ (see Fig. 5). In general case, one can apply an algorithm (e.g., Dijkstra algorithm (Cormen *et al.*, 2001)) for the optimal solution of the task.

How to construct and use the aggregating/sequencing model? Summarizing the whole discussion, we state that the following items are at the core of the approach: the initial set of LO units, criteria of weights to estimate the pedagogic/learning value of LOs and
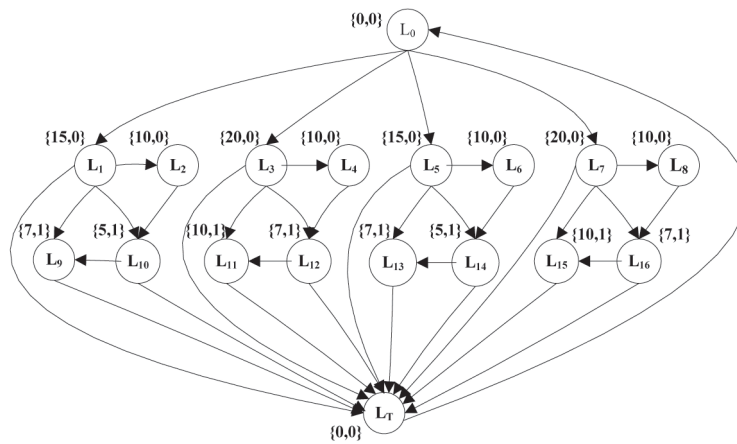
Fig. 6. The node-weighted graph as an aggregating/sequencing model that corresponds to attributes given in Table 1.

identification of an order among units. The order within the aggregated sequence is to be identified by course designer (teacher) keeping in mind the requirements of e-learning theories, recommendations of e-learning standards and, of course, the internal structure and semantics of the content. It is the responsibility of teacher or learner what aggregate to choose, when the model is used through the adequate means of managing and technological support.

8. Evaluation and Discussion

The aggregating/sequencing model we have introduced is presented as the node-weighted/arc-weighted connected graph in which nodes represent derivative LO units derived from the given GLO and arcs (directed branches) represent a logical sequence among the units for their interpreting within the learning/teaching process. The separate route (from the initial LO to the terminal LO in the model) represents one aggregate of the teaching content. All the routes specify the whole space (i.e., a package) of possible aggregates that, again, can be used to combine some aggregates into a higher-level package. The model enables to view learning/teaching as a formal process through identifying of multi-valued routes (loops) from the initial node to the terminal node according to the pre-specified rules (i.e., criteria and constraints that follow from pedagogical theories, e.g., first learn simplest items and then go to more complex ones, etc.).

Since the aggregating/sequencing model is based on properties of the GLO and on a variety of attributes (e.g., weights, roles, and ordering relations) that the model per se describes, the space for identifying the relevant sequences is very wide and optimization techniques can be applied. To do so the attribute values are to be identified first. We have suggested a multi-valued evaluation of LO units. The evaluation is based on using some principles of learning theories (e.g., Wiley, 2000a). Some evaluation attributes are

introduced along with the generative technology (i.e., meta-programming because it, e.g., enables to pre-program some capabilities to support the paradigm 'learning by doing'). A set of routes, i.e. solutions of the aggregating/sequencing problem can be obtained automatically using known algorithms (such as Dijkstra algorithm) that model the content delivery process.

Even if the weights are identified imprecisely or a teacher/learner has no intention to use the model and to optimize the process in a real setting, the awareness of boundaries of the process according to some evaluating weights is important from the theory viewpoint (e.g., at the course planning phase).

Advantages of the approach are as follows. The model (when it is used in the context of GLOs) supports reusability aspects at a higher extent because the model extends the scope of reuse and opens the possibility to automate and optimize the process. It also enables to describe explicitly the integration of technology advances with pedagogical and social issues. As a result the proposed aggregating/sequencing model is a solid basis for e-learning theory because the model explicitly (using measurable attributes) describes the fundamental aspects of teaching process and represent the aspects formally. The model can be used not only in the context of GLOs. The model is also beneficial in the case when the related LO units are constructed manually or retrieved from repositories (if the number of related LO units is sufficient).

Restrictions of the approach are as follows. The model is more relevant to such cases when the related LO units are small and their number is high enough. On the other hand, a high extent of variants leads to the increase of complexity; the latter restricts to extend the approach for its use in settings with higher-level components (LOs) (such as a set of lessons, module or the whole course). The approach requires a numerical evaluation of LO units by calculating weights, which, in turn, is a complex socio-technical task because it depends upon the social context. The latter may restrict the use of the approach in practice.

9. Conclusions and Future Work

The introduced formal aggregating/sequencing model is a result that was obtained from the analysis of learning theories and standards, as well as properties and capabilities of generative learning objects. The model describes and models the content delivery process at a higher abstraction level precisely. The model also extends the reuse aspects of learning objects because it creates wider capabilities for aggregating/sequencing and quality improvement (e.g., through a higher degree of choice of variants, modification and automation) and opens the way for optimization of managing of learning/teaching process, if the criteria of the process are well-specified. From the technology and pedagogy viewpoints, in our view, the aggregating/sequencing model creates a solid basis for contribution to the e-learning theory. Since the approach has not only many advantages but as well some difficulties and restrictions, further research is needed in order to increase the maturity level of the approach and generative learning objects per se. The anticipated work includes the weight calculations strategies, optimal sequencing, and formation of aggregates at the higher granularity level.

Acknowledgements

References

Altun, A., Askar, P. (2008). From blocks to granules: an alternative approach to designing LO. In: *Eighth IEEE International Conference on Advanced Learning Technologies* (*ICALT*), 446–448.

Alvarez, G.L.A., Montesinos, J.L. (2006). An authoring tool for learning object sequencing. In: *First Latin American Conference on Learning Objects* (*LACLO*), Guayaquil, Ecuador.
`http://gita.inf.uach.cl/publicaciones/authoring_tool_sequencing.pdf`.

Boyle, T., Leeder, D., Chase, H. (2004). To boldly GLO – towards the next generation of learning objects. In: *World Conference on eLearning in Corporate, Government, Healthcare and Higher Education*, Washington, USA.

Boyle, T., Ljubojevic, D., Agombar, M., Baur, E. (2008). The conceptual structure of generative learning objects. In: *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, *AACE*, Vienna, Austria, 4570–4578.

Cormen, T.H.Ch.E., Leiserson, R., Rivest, L., Stein, C. (2001). *Introduction to Algorithms*, 2nd edition. Section 24.3: Dijkstra's algorithm. MIT Press and McGraw-Hill, 595–601.

Huddlestone, J., Pike, J. (2005). Learning objects reuse – a four tier model. In: *People and Systems – Who Are We Designing For*, 25–31. `ieeexplore.ieee.org/iel5/10516/33295/01575286.pdf`

IMS_PCK (2003). *What is IMS Content Packaging*?
`http://www.zope.cetis.ac.uk/lib/media/whatisCP1_1_web.pdf`.

IMS_SS (2003). *Simple Sequencing Information and Behavior Model*. Version 1.0 Final Specification, IMS Global Learning Consortium.

Kramer, B.J. (2009). Generating interactive learning objects from configurable samples. In: *International Conference on Mobile, Hybrid, and On-Line Learning* (*ELML '09*). Cancun, 1–6.

LTSC (Learning Technology Standards Committee) (2002). *IEEE Standard for Learning Object Metadata*. IEEE Standard 1484.12.1. IEEE, N.Y.

LOM (2002). *Draft Standart for Learning Object Metadata*. IEEE Learning Technology Standarts Committee. IEEE 1484.12.1-2002

McGreal, R. (Ed.) (2004). *Online Education Using Learning Objects.* Routledge, N.Y.

Memmel, M., Ras, E., Jantke, K., Yacci, M. (2007). Approaches to learning object oriented instructional design. In: Koohang, A., Harman, K. (Eds.), *Learning Objects and InstructionalDesign*. Informing Science Press, 281–326.

Morales, R., Leeder, D., Boyle, T. (2005). A case in the design of generative learning objects (GLOs): applied statistical methods. In: *Proc. of World Conf. on Educational Multimedia, Hypermedia and Telecommunications*, AACE, Chesapeake, VA, 2091–2097.

Nortrup, P.T. (ed.) (2007). *Learning Objects for Instruction*: *Design and Evaluation*. Information Science Publishing, New York.

Nugent, G., Soh, L.-K., Samal, A. (2006). Design, development, and validation of learning objects. *J. Educational Technology Systems*, 34(3), 271–281.

Rossano, V., Joy, M., Rosselli, T., Sutinen, E. (2005). A taxonomy for definitions and applications of LOs: A meta-analysis of ICALT papers. *J. Educational Technology and Society*, 8(4), 148–160.

Schobbens, P.-Y., Heymans, P., Trigaux, J.-Ch., Bontemps, Y. (2006). Feature diagrams: a survey and a formal semantics. In: *14th IEEE International Requirements Engineering Conference* (*RE'06*).

SCORM (2004). *Sharable Courseware Object Reference Model. The SCORM Overview. Advanced Distributed Learning (ADL) Initiative*. SCORM 2004 3rd Edition.

Sheard, T. (2001). Accomplishments and research challenges in meta-programming. In: *2nd Int. Workshop on Semantics, Application, and Implementation of Program Generation* (*SAIG'2001*), Florence, Italy. *LNCS*, Vol. 2196, 2–44.

Štuikys, V., Damaševičius, R. (2008). Development of generative learning objects using feature diagrams and generative techniques. *Informatics in Education*, 7(2), 277–288.

Štuikys, V., Damaševičius, R., Ziberkas, G. (2002). Open PROMOL: an experimental language for target program modification. In: Mignotte, A., Vilar, E., Horobin, L. (Eds), *System on Chip Design Languages*. Kluwer Academic Publishers, 235–246.

Štuikys, V., Damaševičius, R., Brauklytė, I. (2009). How to integrate generative learning objects into teaching and learning processes. In: *Proc. of the Intern. Conf. on Information and Software Technologies*, Information Technologies. Kaunas University of Technology, 292–301.

Wiley, D.A. (2000a). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In: Wiley, D.A. (Ed.), *The Instructional Use of Learning Objects*.

Wiley, D.A. (2000b). *Learning Object Design and Sequencing Theory*. PhD Thesis, Department of Instructional Psychology and Technology, Brigham Young University.

V. Štuikys holds PhD degree since 1970 and degree of doctor habilitatis since 2003. Currently he occupies a position of a professor at Kaunas University of Technology, Software Engineering Department. His research interests include domain specific and software reuse, high level domain-specific languages, component-based programming, meta-programming and program generation, interdisciplinary aspects of automation in various domains, including e-learning systems, embedded systems, etc.

I. Brauklytė holds master degree in computer sciences from Klaipėda University since 2002. Currently she is PhD student at Kaunas University of Technology and also a lecturer at Klaipėda University, Faculty of Marine Engineering, Informatics Engineering Department. Her research interests include e-learning, learning objects, generative learning objects, meta-programming, and software reuse. She is an author of several labs and e-learning materials in virtual environment. She contributed in the world, international and republican scientific conferences of young scientists and participated in the methodological-training camps.

Mokymo(si) objektų vienetų, kurie išvedami iš generatyviojo mokymo(si) objekto, agregavimas

Vytautas ŠTUIKYS, Ilona BRAUKLYTĖ

Mokymo turinio elementų (dalių) agregavimas ir jų eiliškumas yra centrinės sąvokos, kuriomis remiasi el. mokymo teorijos ir standartai. Straipsnyje nagrinėjamos turinio agregavimo/eiliškumo problemos generatyviųjų mokymo(si) objektų (GMO) kontekste. GMO, kuriuos pasiūlė Boyle, Morales, Leeder ir jų kolegos 2004 m., yra laikomi naujos kartos mokymo(si) objektais, kadangi jie pateikia daugiau galimybių ir įgalina pasiekti aukštesnę kokybę ir didesnį našumą. Šiame straipsnyje siūloma naudoti metaprogramavimo metodus tam, kad būtų galima specifikuoti GMO, o po to ir automatiškai generuoti MO vienetus pagal naudotojų poreikį. Sugeneruotų vienetų agregavimas tam, kad būtų galima sudaryti aukštesnio granuliacijos laipsnio junginius, gali būti atliktas įvairiais būdais priklausomai nuo pasirinktojo kriterijaus arba skirtingų kriterijų tam tikros kompozicijos (pvz., sudėtingumo, granuliacijos laipsnio, semantinio tankio, galimybių modeliuoti mokymo procesą ir kt.). Agregavimo problema šiame straipsnyje yra pateikiama kaip mokymo(si) vienetų vidinio eiliškumo nustatymo uždavinys, kai tokie vienetai yra gaunami iš GMO. Straipsnio mokslinis įnašas yra formalus grafais grindžiamas modelis nagrinėjamajai problemai spręsti, kai mokymo objektų vienetų erdvė yra didelė. Iš pradžių mes formuluojame problemą pateikiant pagrindinių sąvokų apibrėžtis, po to nagrinėjame modelio savybes, analizuojame modelio realizacijos variantus ir, galiausiai, įvertiname siūlomą problemos sprendimą el. mokymui.