OmniaScience

Journal of Technology and Science Education
JOTSE, 2020 – 10(1): 60-71 – Online ISSN: 2013-6374 – Print ISSN: 2014-5349
https://doi.org/10.3926/jotse.755

STUDENT MODEL INITIALIZATION USING DOMAIN KNOWLEDGE ONTOLOGY REPRESENTATIVE SUBSET

Ani Grubišić, Branko Žitko, Slavomir Stankov

University of Split, Faculty of Science (Croatia)

ani@pmfst.hr, branko.zitko@pmfst.hr, slavomir.stankov@pmfst.hr

Received May 2019
Accepted October 2019

Abstract

In intelligent e-learning systems that adapt a learning and teaching process to student knowledge, it is important to adapt the system as quickly as possible. However, adaptation is not possible until the student model is initialized. In this paper, a new approach to student model initialization using domain knowledge representative subset is described. The approach defines which concepts from domain knowledge should be included in the initial test so the system can make conclusions about what students truly know about domain knowledge. This representative subset of domain knowledge is defined using non-semantic mathematical approach based on graph theory. The initial test, created over a domain knowledge representative subset, guarantees encompassing all concepts that are relevant to domain knowledge. A two-level case study is conducted on what would be the representative subset of one selected domain knowledge. It compares semantically selected domain knowledge representative subsets (semantical analysis was done by domain area experts) to a non-semantical, mathematically selected domain knowledge representative subset. The results of the case study show that problems of inequality of semantically selected domain knowledge representative subsets are easily overcome using the presented approach.

Keywords **–** Intelligent tutoring systems, adaptive e-learning systems, adaptive courseware, domain knowledge, ontology, initialization of the student model.

To cite this article:

Grubišić, A., Žitko, B., & Stankov, S. (2020). Student model initialization using domain knowledge ontology representative subset. *Journal of Technology and Science Education,* 10(1), 60-71. https://doi.org/10.3926/jotse.755

---

1. Introduction

All teachers, at the beginning of a course, want to know what their students' foreknowledge is about the domain knowledge that the instructor is about to teach. This information is needed to appropriately tailor the learning and teaching process. However, all teachers face the same problem: namely how to assess students' foreknowledge? How many questions should be given to students so that one can accurately examine what students truly know about specific aspects of domain knowledge? What domain knowledge concepts should be included in the pre-assessment questions? What kind of questions should be used?

In intelligent e-learning systems that adapt to student knowledge, it is necessary to start adapting the learning and teaching process as soon as possible. Ideally this adaptation should occur immediately after initial student model is built (Corbett & Anderson, 1994). Without an appropriate student model initialization, the entire learning and teaching process in intelligent e-learning systems can become ineffective. This phenomenon can be likened to a physician that attempts to treat a disease without knowing the symptoms of disease (Glaser & Nitko, 1970).

Typically, initialization is accomplished using some form of an initial test or questionnaire. As with all knowledge testing, the main issues that surround an initial test or questionnaire occurs in question generation, and more importantly, in determining the subset of the domain knowledge that represents pre-requisite student foreknowledge. If an instructor wants to be 100% sure that they have asked all the right questions, they need to ask at least one question about every concept in the domain knowledge. However, such an initial test would be too long and burdensome for the student and thus be ineffective for student model initialization (Aïmeur, Brassard, Dufort & Gambs, 2002).

In this paper, a new approach to student model initialization using a representative subset of domain knowledge is described. The approach defines which concepts from the domain knowledge ontology should be included in the initial test in order to accurately identify what the student truly knows about the domain knowledge. The selected concepts can be thought of as a proxy of the entire domain knowledge.

Every unique domain knowledge has its own representative subset. In our approach, the selection of concepts for the representative subset is done using an algorithm (presented in this paper), regardless of the content associated with the domain knowledge. Consequently, the problems associated with semantically selected domain knowledge representative subsets done by field experts are avoided. When field experts are utilized to develop the domain knowledge relative subset, the result is expert to expert variability in the representative subsets (i.e. experts will differ in what they define to be the representative subset due to the fact that concepts are selected based on their semantical meaning). The inherent variability associated with semantically selected representative subsets is problematic because student model initialization relies on the use of single representative subset for a particular set of domain knowledge. As a result, the development of a non-semantic, mathematical approach for determining a domain knowledge representative subset would be an important contribution to the literature. Such an approach should be able to be utilized for any set of domain knowledge that is of interest to the instructor and would include automatic selection of concepts that are foundational to the given set of domain knowledge.

The main research question considered here is how to select a representative subset of the domain knowledge in a uniform and efficient manner, regardless of the domain knowledge under consideration. The selection process is best thought of as an optimization problem. In optimization problems, solutions which are optimal or near-optimal, with respect to some pre-identified criteria, are sought after (Rothlauf, 2011). In computer science, an optimization problem is the problem of finding the best solution from all feasible solutions. In our case, we seek to find a subset of a large data set that best represents the original data. Utilizing the same notation as (Duckworth & Wormald, 2010), we let P denote the set of domain knowledge and P* denote a subset, where we seek to find the subset P* that optimally represents P.

The remainder of the manuscript is organized as follows: in Section 2 we discuss the issue of student model initialization and review how others in the literature have addressed model initialization; in Section 3 we describe the process of selecting a representative subset of the domain knowledge – this representative subset will be used for student initialization; we present an experimental evaluation of the proposed model compared with three recent sampling algorithms; we describe how this approach was implemented in one particular intelligent e-learning system.

2. Related Work

Before presenting the contributions of our work to the field of the student model initialization in intelligent e-learning systems, we review how others in the literature have approached student model initialization.

Initial tests/questionnaires (also known as pre-tests or preliminary tests) are commonly used for student model initialization. These tests are typically created manually by teachers for the particular domain knowledge at hand. When the domain knowledge changes, it follows that the questions for the initial test have to be created and selected again (Vištica, Grubišić & Žitko, 2016).

More than twenty years ago, Self (Self, 1994) discussed the necessity of optimizing the sequence of questions used in the student model initialization process. Namely, in order to control the length of the initial test, he used the "concept neighborhood" idea: if concepts A and B are in the same neighborhood, mastery of A implies mastery of B. This idea is one of the main premises of our approach.

The CLARISSE (Aïmeur et al., 2002) uses an intelligent pre-test where the questions are focused on the "important" concepts. The selection of the "important" concepts, that is, the selection of questions related to the "important" concepts, is done by cluster analysis. The starting set of questions is manually created by the instructor. After using a clustering algorithm, the starting set of pre-test questions is reduced. The authors emphasize the trade-off between the number of questions in the pre-test and the accuracy in the student model initialization process. The main drawback is the fact that the instructor has to manually create a large set of questions for each domain knowledge. The larger set of questions has to be tested by students, and then the cluster analysis reduces the large set to a pre-test. This process can be rather time consuming.

In the Web-EasyMath (Tsiriga & Virvou, 2004), the initial test contains a manually selected representative set of questions that cover the whole domain being taught. Results are then used to calculate the similarity between the student and other students that have already used Web-EasyMath in order to determine the stereotype category of the subject with respect to her/his knowledge level of the domain being taught. There are four stereotypes which reflect the levels of student knowledge: novice, beginner, intermediate and advanced. This process suffers due to the manual creation of the initial test because the creation and subsequent selection of questions has to be done for each new domain knowledge. The same approach is used in (González, Burguillo, Llamas & Laza, 2013).

In the SIETTE (Conejo, Guzmán, Millán, Pérez-de-la-Cruz & Trella, 2004), a student model is initialized using a pre-test of the whole domain knowledge based on a hierarchically structured curriculum and the "complete assessment mode". The mechanisms used to carry out the selection of the most suitable questions are based on a psychometric theory known as Item Response Theory (IRT). The IRT focuses on the individual items (questions), as opposed to classical theory of testing which focuses on the test as a whole. The main disadvantage of this initialization process is the fact that questions in the pre-test are manually created and they must cover the entire domain knowledge, thus making the pre-test too burdensome on the student.

The LS-Plan (Limongelli, Sciarrone, & Vaste, 2008) uses the Knowledge Space Theory (course concepts are mod-elled as atomic elements of knowledge) and the Felder-Silverman's Learning Styles Model. The student model is initialized using an initial questionnaire that is prepared by the instructor. Like other methods discussed, the main disadvantage is the fact that questions in the pre-test are created and selected manually. A similar approach to LS-Plan is used in Wayang Outpost, a geometry tutor that helps students learn to solve geometry problems. This geometry tutor uses 28 manually created problems in a pre-test (Ferguson, Arroyo, Mahadevan, Woolf & Barto, 2006).

In the DEPTHS, an intelligent tutoring system for learning software design patterns (Jeremic, Jovanovic & Gasevic, 2009), three basic categories of the students' characteristics are used: personal

data, performance data and individual preferences, and teaching history. The student model is initialized based on the student's self-assessment. The system then subsequently assigns the student one of the following stereotypes: beginner, intermediate or advanced (expert). The major shortcoming of this approach is a subjective self-assessment that cannot give valid pre-dictions about student's knowledge.

In the adaptive e-learning system based on LearnSquare (Esichaikul, Lamnoi & Bechter, 2011), the student model initialization is based on pre-test results analyzed by the Dempster-Shafer (DS) theory (general framework for reasoning with uncertainty). This process includes: evidence extraction, evidence combination and student model initialization. The results from the pre-test answers are input to the DS formula for determining the level of the student's knowledge in all domain knowledge concepts. The creation of questions and their selection is not described.

None of the aforementioned systems utilizes an automatic detection of concepts from the domain knowledge for inclusion into the representative subset. To the best of our knowledge, there is no approach that enables completely automatically generation of questions based on the domain knowledge structure, that is not domain related (for example, algebra problems).

3. Domain Knowledge Ontology Representative Subset

Many intelligent e-learning systems use domain knowledge presented in the form of a conceptual graph, network or ontology (Tangjin & Xianhon, 2010). By using these structures as a representation of domain knowledge, one can think about exploiting the structure in order to automate the selection of a representative subset. The problem of selection of a representative subset then becomes an optimization problem which we develop in this section.

Each area of human activity can be presented as a series of properly related concepts (Grubišić, Stankov & Peraić, 2013: page 5363). The main structural elements of the conceptual model are the concepts and relations (Lee, Hendler, & Lassila, 2001). Domain knowledge is then expressed as a structure with concepts and relations between the concepts. As the direction of the relation between concepts has to be indicated, the terms subconcept and super-concept are used (Grubišić et al., 2013).

In order to clearly indicate for each relation in the ontology which concepts it connects and what the nature of that relation is, Definition1 is presented (from Grubišić et al., 2013: page 5365)

> **Definition 1**
>
> Let set $E_{CON}=\{K_1,\ldots,K_n\}$, $n\geq 0$, be a set of concepts, set $E_{REL}=\{r_1,\ldots,r_m\}\cup\{has\_subtype, has\_instance, has\_part\}$, $m\geq 0$, a set of relations and $\emptyset_E$ an empty element. **Domain knowledge (DK)** is a set of triplets (K_1, r, K_2) that define that the concepts K_1 and K_2 are connected with relation r. In this way we define that the concept K_1 is **superconcept** of concept K_2 and that concept K_2 is **subconcept** of concept K_1.

A relation 'has_subtype' is used for hierarchical categorization of concepts and presents a generalization that enables property inheritance between superconcepts and subconcepts. A relation 'has_instance' is used for connecting a concept that presents a class with a concept that presents an instance of that class. A relation 'has_part' is used for the structural breakdown of the concept into its parts.

3.1. Domain Knowledge Representation

Since the fundamental elements of the domain knowledge are concepts and relationships between them, graph theory is used as a mathematical base for the visualization of domain knowledge. Therefore, a directed domain knowledge graph on which all the rules from the graph theory apply is defined furthermore.

Definition 2

For domain knowledge DK we define directed **domain knowledge graph DKG=(V,A)** where the set of vertices is $V=E_{CON}$ and a set of edges $A=\{(K_1,K_2)|\exists(K_1,r,K_2)\in DK,\ r\neq\emptyset_E,\ K_1\neq K_2\}$ is equal to a set of ordered pairs of those concepts from the domain knowledge that are related.

The set of concept K_x's superconcepts is a set $\boldsymbol{SuperK_x}=\{K\in E_{CON}|\exists(K,r,K_x)\in DK,\ K\neq K_x,\ r\neq\emptyset_E\}=\{K\in V|\exists(K,K_x)\in A,K\neq K_x\}$. The number $\boldsymbol{supK_x}$ is equal to the number of elements in the set $SuperK_x$ and denotes the number of concept K_x's superconcepts.

The set of concept K_x's subconcepts is a set $\boldsymbol{SubK_x}=\{K\in E_{CON}|\exists(K_x,r,K)\in DK,\ K\neq K_x,\ r\neq\emptyset_E\}=\{K\in V|\exists(K_x,K)\in A,\ K\neq K_x\}$. The number $\boldsymbol{subK_x}$ is equal to the number of elements in the set $SubK_x$ and denotes the number of concept K_x's subconcepts.

The vertex from DKG is called a ***root*** if it has no superconcepts and has subconcepts.

The vertex from DKG is called a ***leaf*** if it has superconcepts and has no subconcepts.

The algorithm for finding paths (if any paths exist) between the two vertices K_x and K_y in DKG is based on a standardized depth-first graph search with backtracking that starts from the vertex K_x.

Definition 3

A ***graph unit*** $\boldsymbol{C_i}$ is the largest connected subgraph of DKG whose vertex set contains only one root. The root of the graph unit $\boldsymbol{MaxVertex_{Ci}}$ is called a **central root** of the graph unit $\boldsymbol{C_i}$.

Any vertex in the graph unit, except the central root, has one or more immediate predecessors. The graph unit is, therefore, a connected subgraph where at least one path exists between every vertex and the central root (except the central root itself). In the graph unit there are no isolated vertices. Thus, each root of the domain knowledge defines one graph unit.

Our algorithm for finding a path between the two vertices can be used for finding the longest path in the graph unit, as well as the central root of the graph unit, as defined in the following definition.

Definition 4

The longest path in the graph unit is a $\boldsymbol{PathK_{Ci}MaxVertex_{Ci}}$ with length $\boldsymbol{MaxLen_{Ci}}$.

A domain knowledge can be extensive; therefore, it is critically important to define a subset of the domain knowledge that sufficiently represents the entire set of domain knowledge. More specifically, the representative subset of domain knowledge should include all the concepts and relationships that are relevant for comprehension of the domain knowledge.

There are two approaches in determining the representative subset of domain knowledge: (i) semantic analysis of concepts and relations in the domain knowledge and (ii) using mathematical methods from graph theory. Both approaches have their advantages and disadvantages.

A semantic analysis requires that a group of individuals with expertise in the domain knowledge determine which concepts and relations are critical for comprising the representative subset. This approach clearly depends upon the heuristics and expertise of the experts who are selected for the job. This representative subset is more "alive" and "real" and better suites the real situation, because it is done differently for each domain knowledge, but the existence of non-uniformity of proposed representations present a challenging problem (see the case study results in the next section).

For a given set of domain knowledge, it is likely that each expert will come up with a different representative subset since experts utilize semantics to develop their respective subset. As such, the diversity in semantic representative subsets is due to expert to expert variability. Given that the use of

experts will generally not result in a mutually agreed upon unique representative subset, a mechanized, non-semantic mathematical determination of the domain knowledge representative subset would be ideal, especially if the process can be utilized regardless of the set of domain knowledge of interest. Our approach does not rely on semantics of concepts and relations, but rather is mathematically based using methods from graph theory. In addition, our approach can be utilized for any set of domain knowledge.

Our idea for determining the domain knowledge representative subset has also been utilized in the fields of social networks, citation networks and communication networks (Vištica et al., 2016). In these areas, graph sampling or graph reduction is used. Two groups of authors, (Leskovec & Faloutsos, 2006) and (Krishnamurthy, Faloutsos, Chrobak, Lao, Cui & Percus, 2005), have discussed this issue and each of them has their own categorization of methods.

Leskovec and Faloutsos (2006) divide the graph sampling methods into three groups: (i) methods based on a random selection of vertices, (ii) a random selection of edges, and (iii) methods based on exploration that simulates random walks or virus propagation.

Krishnamurthy et al. (2005) divide the graph reduction methods into three groups: (i) deletion methods that delete vertices or edges from the graph until a desired size is reached, (ii) contraction methods that shrink the neighboring vertices until a desired size is reached, and (iii) exploration methods that include a certain number of vertices, thus maintaining the characteristics of the original graph.

Studies have shown that the best sampling methods are based on random walks (RW) and forest fires (FF) that define a representative sample comprising 15% of the original graph (Leskovec & Faloutsos, 2006). Random walks are used when someone wants the sample to have similar properties as the original graph (scale-down goal) and when someone wants the sample to be connected. Forest fires are used when someone wants the sample to be similar to what the original graph was when it was the size of a sample (back-in-time goal), that is, it longitudinally observes the development of the graph (Leskovec & Faloutsos, 2006). Studies have shown that the graph reduction methods, such as exploration by breadth first search (EBFS) and exploration by depth first search (EDFS), can reduce the size of the graph by 70% while still retaining the important aspects of the original graph (Krishnamurthy et al., 2005).

After analyzing the sampling methods presented by Leskovec and Faloutsos (2006), we chose exploration sampling, because in our approach we do not want to randomly select vertices and edges as we want a representative subgraph to be connected. "Exploration" sampling methods include Random Node Neighbor (RNN), Random Walk (RW), Random Jump (RJ) and Forest Fire (FF). A common approach in exploration sampling is a random selection of the first vertex and then exploring the peaks in its vicinity.

Since we want to preserve the characteristics of the original graph and we want a representative subgraph to be connected, we use exploration methods from those discussed by Krishnamurthy et al. (2005). These methods include: an Exploration by Breadth First Search (EBFS) and an Exploration by Depth First Search (EDFS). The usual approach in the exploration methods is a random selection of the first vertex and then crossing over the graph according to the selected sampling method.

Since we want our sample to be a connected subgraph that has similar properties as the whole domain knowledge graph, we use a combination of random walks (RW) and exploration by depth first search (EDFS). We now introduce mathematical definitions and methods that serve as the foundation for the selected graph sampling and reduction methods.

Let G be a domain knowledge graph that has n vertices. The aim is to create a representative subgraph S with n_0 vertices, $n_0 < n$, which will be most similar to the graph G, that is, we want S to have similar properties as G.

In our approach it is required that the random walk is the longest path from selected starting vertex (depth first search). For each domain knowledge graph unit, we start random walk from its central root

MaxVertexCi, that is, for graph unit the combination of random walk (RW) and exploration by depth first search (EDFS) will lead to $PathK_{Ci}MaxVertex_{Ci}$. Now, the next definition is proposed:

Definition 5

Representative subgraph RepDKG of directed domain knowledge graph is the union of the longest paths in all DKG graph units, ie.

$$RepDKG = \bigcup_{i=1}^{n} PathK_{C_i} MaxVertex_{C_i}$$

Representation of domain knowledge RepDK is a subset of domain knowledge such that $RepDK=\{(K_1,r,K_2) \in DK \mid K_1,K_2 \in V_{RepDKG}\}$.

This graph sampling approach is mathematical because the semantic meaning that is embedded in the ontology structure does not directly influence the mechanism for representative subset selection. There is some indirect influence of the ontology semantic meaning that is included in the selected representative subset, but we do not take it into account.

3.2. Experimental Evaluation of the Proposed Model

In order to assess how representative the subset RepDKG is, we evaluated whether the sampled graph is able to preserve the distributions of several characteristic topological graph properties such as degree, path length and clustering coefficients (Ahmed, Neville & Kompella, 2011), (Cem, Tozal & Sarac, 2013). We have compared the statistics of our subgraph with the statistics of the subgraphs that are gained using a variety of sampling techniques such as Fire Forest Sampling (FFS) (Leskovec & Faloutsos, 2006), Snowball Sampling (SS) (Lee, Kim & Jeong, 2006) and Metropolis-Hastings Sampling (MHS) (Lu & Bressan, 2012).

In this section, we evaluate the efficacy of our sampling algorithm, compared to the other sampling algorithms mentioned above, on real data set - a citation network cit-HepPh (available from snap.stanford.edu). We chose the citation network since this is the most well-known example of acyclic graphs - the cited works are older than the citing work.

For the purposes of this graph analysis we have used Gephi version 0.8.2 (https://gephi.org/). All sampling algorithms created a sample that retained 37% of original nodes, but the number of selected edges ranges from 27% (our approach and metropolis) to 46% (forest fire) of original edges. All sampling algorithms, but our own, sampled subgraphs that have no connected components, which exist in the original graph. Our approach had an average path length of 6.2, which is the closest to the original graphs average path length of 11.69. The density of the original and the sampled subgraph using our algorithm are closest (original graph=0.000353, our approach=0.000454). The closest clustering coefficient, to that of the original graph was that resulting from the metropolis algorithm (Table 1).

Graph or subgraph	Nodes	Edges	No. CC	Avg. path	Density	Clustering coefficient
Cit-HepPh	34546	421578	61	11.69	0.0007	0.143
representation	12725 (37%)	112147 (27%)	90	6.20	0.0007	0.134
metropolis	12782 (37%)	115787 (27%)	1	5.42	0.001	0.144
forrest fire	12891 (37%)	195691 (46%)	1	4.49	0.001	0.151
snowball	12876 (37%)	185497 (44%)	1	5.04	0.001	0.158

Table 1. Characteristics of original graph and its sampled subgraphs

Our experimental evaluation is primarily along three main characteristics—degree, path length and clustering coefficient (as in (Ahmed et al., 2011) and (Leskovec & Faloutsos, 2006)). We have measured the performance of a sampling algorithm by how well the sampled subgraphs preserve the cumulative distribution function of each of these three characteristics.

From the distribution figures (Figure 1), it can be observed that all of the compared algorithms are accurate at preserving degree distributions, as very little differences exist. Our algorithm shows its ability to estimate the amount of low and high degree nodes, as well as Metropolis algorithm. The path length distribution reveals that our algorithm has a higher fraction of long path lengths, which is not surprising since our approach relies on finding the longest path. In the case of clustering coefficient, our algorithm shows a higher fraction of low clustered nodes since it explores only 2 nodes from the neighbors of the observed node (its immediate predecessor and its immediate follower). It also tends to miss several edges among the sampled nodes.

This analysis compared the behavior of our approach to graph sampling with three of the most popular sampling algorithms in the literature. Taking into account the nature of our algorithm (it finds the longest path in a graph unit) and the fact that it is the only sampling algorithm that does not base itself on random node selection in any segment, our sampling approach performs admirably in all comparisons made. In future research, we hope to conduct similar analyses using the presented graph metrics on larger, more complicated graphs to determine the relative effect of scale on each of the competing approaches.

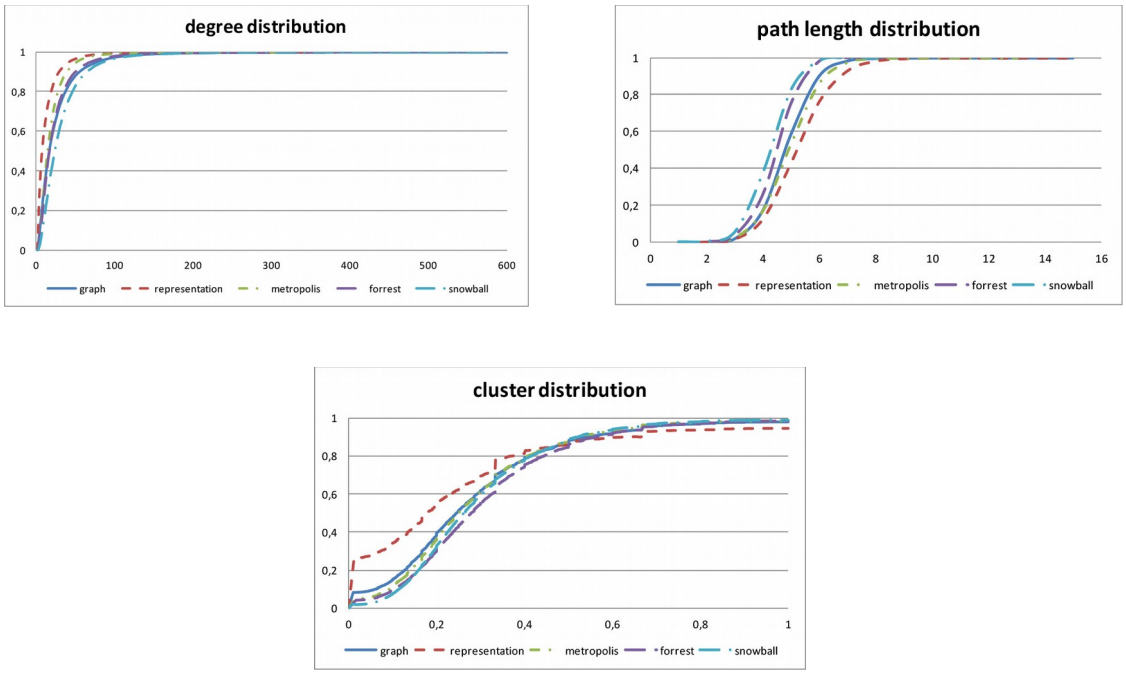

Figure 1. Degree, path length and cluster distribution

3.3. A Student Model Initialization in the AC-Ware Tutor

The described sampling approach was implemented in a system called Adaptive Courseware Tutor (AC-ware Tutor), an intelligent tutoring system with fully automated generation of courseware elements for learning and teaching, dynamic selection and sequencing of courseware elements and it realizes adaptation to student's knowledge, as it's the most important feature. This system automatically generates an initial test over a representative subset of the domain knowledge (Grubišić, Stankov & Peraić, 2013). We now present how the described approach can be used for initialization of a stereotype student model in the AC-ware Tutor (Grubišić et al., 2013).

Stereotypes are especially appropriate when someone wants to make initial assumptions regarding student's knowledge level of the domain knowledge being taught (Tsiriga & Virvou, 2004). In the AC-ware Tutor, the accustomed pedagogical terminology and the Bloom's knowledge taxonomy (Bloom, 1956) are combined to define students' stereotypes (Grubišić et al., 2013). We defined five knowledge stereotypes: novice, beginner, intermediate, advanced and expert. The number of stereotypes corresponds exactly to custom pedagogical practices that assess students in a range from 1 to 5 (or F, E to A).

Testing courseware elements in the AC-ware tutor contain questions generated over one subset of the domain knowledge. A graph DKG'=(V',A') is a subset of the domain knowledge that is defined according to the subset DK'. If testing courseware element is the initial test, then DK'=RepDK, DKG'=RepDKG.

Testing courseware elements contain certain number of questions generated using templates. Those templates have four difficulty levels closely related to stereotypes and the Bloom's taxonomy. Each difficulty level examines a certain knowledge level: the first level templates test knowledge recall, the second level templates test knowledge comprehension, the third level templates test knowledge application and the fourth level templates test knowledge analysis, synthesis and evaluation (Grubišić et al., 2013).

Each question tests knowledge about relations between concepts, as well as knowledge about the concepts themselves. Answers are scored on a scale from 0 to 4 points. Regardless of the question, if the answer is "I don't know", then the score is 0 points. Incorrect answers are scored as 0 points, while a correct answer scores as many points as the question difficulty level is.

The initial test has a minimum of two and a maximum of three iterations. The first set of questions in the initial test is generated based on the question templates from the third difficulty level (L=3). The concepts that are included in the questions that the student has answered incorrectly become an input for generation of questions based on templates from the second difficulty level (L=2) and the concepts that are included in the questions that the student has answered correctly become an input for generation of questions based on templates from the fourth difficulty level (L=4). The concepts that are included in the questions based on templates from the second difficulty level that were answered incorrectly become an input for generation of questions based on templates from the first difficulty level (L=1).

After the initial knowledge test is generated over the representative subset of domain knowledge, the initial student stereotype is determined, and the student model initialization is finished. Then, the AC-ware system can adapt the process of learning, teaching and testing students' knowledge to an estimate of the student's current knowledge level.

We have compared the student model initialization approach in the AC-ware Tutor with the initialization done by the teacher with ten years of experience in teaching "Introduction to programming" course. A paper based initial test (scored from 0 to 100), written by the classroom instructor was administered to the same students that used the AC-ware Tutor. The classroom instructor evaluated results from the initial test and subsequently assigned a stereotype using the following scale: score 0-19 corresponds to stereotype novice, 20-39 beginner, 40-59 intermediate, 60-79 advanced, 80-100 expert.

We have analyzed if there was a statistically significant difference between the distributions of stereotypes assigned through these two initialization approaches (Table 2). The experiment involved 33 undergraduate students that enrolled course "Introduction to programming". Their initial stereotype was determined by the AC-ware Tutor and by their teacher based on pre-test results. The resulting Chi-Square test of independence ($p=0.1518$) suggest that the results of the student model initialization done by the AC-ware Tutor are not statistically significantly different from the initialization of the same students done by the teacher, but the benefits of automatization of that process are enormous for the teacher.

	Stereotype				
Frequency	**Novice**	**Beginner**	**Intermediate**	**Advanced**	**Expert**
Observed (AC-ware Tutor)	6	1	5	3	4
Expected (real teacher)	0	1	11	6	1

Table 2. Observed and expected frequencies of stereotypes

4. Conclusions

In this paper, a new approach for developing a representative subset of domain knowledge is presented. The approach enables initialization of the student model using a non-semantic mathematical approach based on graph theory. As a result, the proposed methodology avoids common issues associated with semantically chosen representative subsets including the time and cost of utilizing a team of experts as well as the inherent variability in expert to expert representative subsets. It is difficult for human experts to agree about the importance of the concepts (what is important to one human may or may not be important to others) and difficult to select a final subset that truly represents the set of domain knowledge. Therefore, the main advantage of developing a "machine made" representative subset is the fact that it can be created in a short time period (i.e. the time it takes for the computer algorithm to run) whereas the creation of a unique "human made" representative subset takes weeks due to the diversity of human views on domain knowledge. In this way initial tests in intelligent e-learning systems for different domains can be created in a second.

Representative subsets of domain knowledge help one to determine the concepts of domain knowledge on which to generate an initial test used to test student's knowledge about some domain knowledge. The initial test, created over the domain knowledge representative subset, guarantees that it will encompass all concepts and relationships that are relevant to a domain knowledge. The future work in the area of non-semantic mathematical approach to defining representative subsets of domain knowledge, will involve comparing this model to different domain knowledge graph sampling and reduction methods, in order to experimentally verify the correctness of this approach. Furthermore, the future research will definitely focus on combining this mathematical approach with some semantic metric in order to avoid rigid representative subsets. It is also necessary to study the robustness of our method on a larger scaled problem and whether in these larger problems, our method can be improved by hybridizing it with elements of semantic approaches.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This paper describes the results of research being carried out within project 177-0361994-1996 Design and evaluation of intelligent e-learning systems within the program 036-1994 Intelligent Support to Omnipresence of e-Learning Systems, funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

Acknowlegment

Special thanks to Dr. Timothy Robins, University of Wyoming, for suggestions how to improve the paper.

References

Ahmed, N.K., Neville, J., & Kompella, R. (2011). Network Sampling via Edge-based Node Selection with Graph Induction. *Technical Report TR-11-016*, Dept of Computer Science, Purdue University.

Aïmeur, E., Brassard, G., Dufort, H., & Gambs, S. (2002). CLARISSE: A Machine Learning Tool to Initialize Student Models. In Cerri, S.A., Gouardères, G., & Paraguaçu, F. (Eds.), *Intelligent Tutoring Systems* (718-728). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-47987-2_72

Bloom, B.S. (1956). *Taxonomy of educational objectives. The classification of educational goals*. Handbook I Cognitive Domain. Green, New York, NY: Committee of College and University Examiners, Longmans.

Cem, E., Tozal, M.E., & Sarac, K. (2013). Impact of Sampling Design in Estimation of Graph Characteristics. In *International Performance Computing and Communications Conference*. San Diego, USA. https://doi.org/10.1109/PCCC.2013.6742788

Conejo, R., Guzmán, E., Millán, E., Pérez-de-la-Cruz, J.L., & Trella, M. (2004). Siette: a web-based tool for adaptive testing. *Int Journal of Artificial Intelligence in Education*, 14, 29-61.

Corbett, A.T., & Anderson, J.R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model User-Adapt Interact*, 4, 253-278. https://doi.org/10.1007/BF01099821

Duckworth, W., & Wormald, N.C. (2010). Linear Programming and the Worst-Case Analysis of Greedy Algorithms on Cubic Graphs. *The Electronic Journal of Combinatorics*, 17(1).

Esichaikul, V., Lamnoi, S., & Bechter, C. (2011). Student Modelling in Adaptive E-Learning Systems. *Knowledge Management & E-Learning: An International Journal (KM&EL)*, 3(3), 342-355. https://doi.org/10.34105/j.kmel.2011.03.025

Ferguson, K., Arroyo, I., Mahadevan, S., Woolf, B., & Barto, A. (2006). Improving Intelligent Tutoring Systems: Using Expectation Maximization to Learn Student Skill Levels. In Ikeda, M., Ashley, K.D., & Chan, T.W. (Eds.), *Intelligent Tutoring Systems*, 453-462. Springer Berlin Heidelberg. https://doi.org/10.1007/11774303_45

Glaser, R., & Nitko, A.J. (1970). *Measurement in Learning and Instruction.*

González, C., Burguillo, J.C., Llamas, M., & Laza, R. (2013). Designing Intelligent Tutoring Systems: A Personalization Strategy using Case-Based Reasoning and Multi-Agent Systems. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 1(4), 41-54.

Grubišić, A., Stankov, S., & Peraić, I. (2013). Ontology based approach to Bayesian student model design. *Expert Systems with Applications*, 40(13), 5363-5371. https://doi.org/10.1016/j.eswa.2013.03.041

Grubišić, A., Stankov, S., & Žitko, B. (2013). Stereotype Student Model for an Adaptive e-Learning System. ICIIS 2013: International Conference on Information and Intelligent Systems. Venice, Italy. *Special Journal Issue on Advances in Information and Intelligent Systems, World Academy of Science, Engineering and Technology* (76), 20-27.

Jeremic, Z., Jovanovic, J., & Gasevic, D. (2009). Evaluating an Intelligent Tutoring System for Design Patterns: The DEPTHS Experience. *Educational Technology & Society*, 12(2), 111-130.

Krishnamurthy, V., Faloutsos, M., Chrobak, M., Lao, L., Cui, J.H., & Percus, A.G. (2005). Reducing large internet topologies for faster simulations. *Networking 2005*, 328-341. https://doi.org/10.1007/11422778_27

Lee, S.H., Kim, P.J., & Jeong, H. (2006). Statistical properties of sampled networks. *Physical Review*, E 73(1). https://doi.org/10.1103/PhysRevE.73.016102

Lee, T.B., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 34-43. https://doi.org/10.1038/scientificamerican0501-34

Leskovec, J., & Faloutsos, C. (2006). Sampling from large graphs. Proceedings of the *12th ACM SIGKDD international conference on Knowledge discovery and data mining* (631-636). https://doi.org/10.1145/1150402.1150479

Limongelli, C., Sciarrone, F., & Vaste, G. (2008). LS-Plan: An Effective Combination of Dynamic Courseware Generation and Learning Styles in Web-Based Education. In Nejdl, W., Kay, J.,. Pu, P., & Herder, E. (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems* (133-142). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-70987-9_16

Lu, X., & Bressan, S. (2012). Sampling Connected Induced Subgraphs Uniformly at Random. Scientific and Statistical Database Management. *Lecture Notes in Computer Science*, 7338, 195-212. https://doi.org/10.1007/978-3-642-31235-9_13

Rothlauf, P.D.F. (2011). Optimization Problems. In *Design of Modern Heuristics* (7-44). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72962-4_2

Self, J. (1994). Formal Approaches to Student Modelling. In: Greer, J.E., & McCalla, G.I. (Eds.) *Student Modelling: The Key to Individualized Knowledge-Based Instruction. NATO ASI Series (Series F: Computer and Systems Sciences,* 125). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-03037-0_12

Vištica, M., Grubišić, A., & Žitko, B. (2016) Applying Graph Sampling Methods on Student Model Initialization in Intelligent Tutoring Systems. *International Journal of Information and Learning Technology*, 33(4). Emerald Group Publishing. https://doi.org/10.1108/IJILT-03-2016-0011

Tangjin, J., & Xiahong, W. (2010). Intelligent tutoring system based on computing conceptual graphs. In *2010 International Conference on Artificial Intelligence and Education (ICAIE)* (60-62).https://doi.org/10.1109/ICAIE.2010.5641488

Tsiriga, V., & Virvou, M. (2004). A Framework for the Initialization of Student Models in Web-based Intelligent Tutoring Systems. *User Modeling and User-Adapted Interaction*, 14(4), 289-316. https://doi.org/10.1023/B:USER.0000043396.14788.cc

Published by OmniaScience (www.omniascience.com)

Journal of Technology and Science Education, 2020 (www.jotse.org)