Graph–based Domain Model for Adaptive Learning Path Recommendation

Dade Nurjanah
School of Computing
Telkom University
Bandung, Indonesia
dadenurjanah@telkomuniversity.ac.id

Muhammad Fiqri
School of Computing
Telkom University
Bandung, Indonesia
iffiqri@gmail.com

***Abstract*— Adaptive educational hypermedia (AEH) has been an alternative tool to replace conventional learning tools. It mainly offers adaptive navigation and presentation based on a wide range of characteristics, preferences and understanding of learners. The adaptation is enabled by a user model which records users' characteristics, domain models which represent all concepts taught in a course and relationships among them, and goal and adaptation models that infer domain and user models to produce adaptation. The main adaptation offered in AEH deals with where a learner can go next. However, it does not inform learners how far they are from the goals. This paper addresses the problem of finding the shortest path to reach a learning goal. We propose a modified Dijkstra algorithm applied to a graph-based domain model. It takes account of learners' knowledge, the weight of topics in a course and the influence scores of each topic to learn the other topics to predict the possibility of success in learning a topic. Based on the success possibility scores, learning paths are recommended to students. Since students' knowledge is progressive, the success possibility scores are dynamics and they will result in a dynamic learning path adaptive to students' progress.**

Keywords—domain model, user model, adaptive learning path, graph, weights, influence scores, learners' knowledge, Dijkstra.

I. INTRODUCTION

Fundamental changes have happened in education. Learning nowadays tends to apply constructivism, rather than instructivism, which requires learners to be active by doing self-directed learning. With the availability of advanced information technology, it is now possible to conduct learning through online media. The technology has enabled learning processes to be done remotely without direct face-to-face meetings between teachers and learners. In fact, not all learning systems can meet the users' needs, since some of them apply conventional approaches. Conventional approaches were commonly applied in classrooms and each learner was considered to have the same goals, prior knowledge and capabilities. They ignore the differences and uniqueness of learners by imposing equal treatment on all learners, such as the same learning flow and materials.

Adaptive educational hypermedia (AEH) has been developed as an alternative tool to replace conventional learning tools, such as general hypermedia, which applies the principle of "one-size-fits-all" [1][2]. AEH mainly offers adaptive navigation and presentation based on a wide range of characteristics, preferences, and understanding of learners [3]. The adaptation is enabled by a user model which records users' characteristics, domain model which represents all concepts taught in a course and relationships among them, and goal and adaptation models that infer domain and user models to produce adaptation.

The main adaptation offered in AEH deals with where a learner can go next. Adaptive navigation is performed by the inference process of the user model. One of the widely used methods for the process is Bayesian Network method, which infers the student's knowledge to produce adaptive navigation [4]. With adaptive navigation learners will be directed to the next concept, problem, or topic that is most suitable for them. However, although adaptive navigation directs learners to the learning goals, it does not inform learners how far they are to the goals.

In AEH, besides adaptive navigation and adaptive presentation, self-navigation is another function offered. It enables learners to navigate learning by exploring knowledge space and hyperspace. Along with the benefits learners can get by self-navigation as they can explore related and interesting concepts, there is also the possibility that they could get lost in exploring hyperspace and knowledge space. Furthermore, they do not get a view of what is the suitable learning path for them to achieve the learning goals. The most suitable shortest learning path aims to provide a view of the way to reach learning goals from the current status.

This paper addresses a 'finding shortest path' problem to reach a learning goal. We proposed a modified Dijkstra algorithm, for which students' knowledge is used as its parameter. The remainder of this paper is organized as follows: Part 2 discusses related work, Part 3 presents the prototype designs and interfaces, Part 4 discusses the research method and the experiment, Part 5 discusses experiments and results and Part 6 presents conclusions.

II. RELATED WORK

A. User Model

The ser model represents the characteristics of learners, which comprise a combination of many attributes, such as learners' experience, education level, knowledge, skill, abilities, grades, interests, and learning styles. Attributes used in user model can be varied, depending on the types of adaptation that will be produced in adaptive educational hypermedia. This makes the user model unique for each learner. Generally, information about users' characteristics consists of personal information, such as name, age, gender, skills, preferences, knowledge, learning goals, and activities and habits during learning [2][3]. Other data that are often used in modelling user are individual traits or specific characteristics of users, for example, personality (introvert / extrovert), cognitive factors, and learning style [2]. The specific characteristics of users are usually used in user model for tools with a high level of adaptation and personalization [2].

There are many types of information which can be considered in adaptive learning, such as background knowledge**,** pre-requisite skills or experience, and user interests [5], user objectives /goals, disabilities, academic background, cognitive capacities, and learning environment [6], personal information, preference/individual traits, and learning history [7], learning styles [8], and personality [9]. Recently, research on user models has been focused on engaging learners by making user models open for learners to access. This is called an Open Learner Model (OLM) [10][11]. OLMs can be viewed or accessed by the learners they represent or by other people, such as teachers, peers, or parents. OLMs give learners control and responsibility in learning processes and provide navigation to suitable materials, exercises, problems, activities, or tasks. By knowing their own current status of knowledge and progress, they will know how to interact and collaborate well with peers.

B. Domain Model

Research on domain models has largely focused on how domain knowledge is organized. There are three most commonly used approaches, namely pedagogically-, cognitively- and mixed-oriented approaches [12]. Pedagogically-oriented approaches organize domain knowledge in the form of a structured or hierarchical set of topics [13]. They describe the linkages between sub-materials in the domain knowledge in the form of a tree. With a pedagogically-oriented approach, students and teachers will get the whole picture of all materials in a hierarchical structure. The disadvantage of this approach is that a deep hierarchy potentially makes students frustrated or bored as they spend much time on learning a topic with all its underlying sub-topics.

On the other hand, the cognitively-oriented approach organizes domain knowledge in the form of a graph of concepts that comprehensively organizes concepts and all relationships between them [14]. From a graph of concepts, students will get a view of why it is important to learn certain concepts before learning a particular concept. Furthermore, this method is favourable since it can support students to explore learning material concepts without any restraint. However, along with this advantage, this method also potentially makes a student get lost when exploring a graph of concepts and fail to reach a learning goal.

A mixed method that combines pedagogically- and cognitively-oriented approaches is proposed to solve the disadvantages that occur in the previous methods. In this method, a pedagogical approach is applied in upper level concepts, such as topics and sub-topics. Afterwards, the cognitive approach is applied to organize concepts under a sub-topic [12].

C. Adaptive Learning Path Recommendation

In an adaptive learning tool, during interaction between the tool and a learner, the tool keeps observing the learner's progress or actions and saves the data into the user model. The goal model will compare the learner's progress with the learning goal and then the adaptation model will find the most suitable shortest path to reach the goal [15]. Finding shortest paths to reach the goal has been studied in previous research. Zhao and Wan [16], for example, proposed an algorithm to select the most suitable shortest learning paths to reach the learning goal. A course is modelled as a graph in which nodes represent knowledge units (KUs) and two nodes in the graph are connected by prerequisite relationships. A connection between two nodes has weights which represent time and effort needed to learn the target KU from the source KU. The path recommended involves the least time or effort that learners need to spend.

Another study has improved Zhao and Wan's work by combining shortest path recommendation and what they called eliminating and optimized selection (EOS) [17]. The main principle is that the shortest path recommendation will be suitable for learners if at first they get suitable learning objects. The shortest learning path is formed over learning objects that have been selected previously. Another study improved the research by applying a greedy algorithm, a classic method to find the shortest path [18]. It modelled a course in a graph and applied node elimination and selection processes to find the shortest path. It maintains learner parameters, including goals, financial statement, environment and language, and learning object parameters, including costs, concepts, environment and language. Research on this area is open in terms of improved algorithms and the diversity of learning or learner parameters used.

The aforementioned work maintained user model and learning path recommendation was performed based on user model and learning goals. A different approach was presented by Henning et al. [19]. They avoided a user model that potentially brought stereotyping to learners. Instead, they developed a pedagogical ontology and a tool that applies self-paced learning with free navigation. The tool keeps observing learners, any progress made by learners is mapped to the pedagogical ontology and the tool recommends an appropriate concept and content to be learnt.

Another previous study which proposed a method for finding the shortest learning path was by Mumey et al. [20]. In this study, students were clustered by finding similarity in the courses and marks already taken and gained. On the other hand, a model of hidden prerequisites was developed. It revealed hidden prerequisite relationships between two courses by finding correlation between marks gained by students in the two courses. The final process was mapping clustering to the prerequisite model to produce a learning path for each cluster.

Research on shortest learning path recommendation does propose new methods, but also various parameters to be recorded in user model and considered in performing the learning paths. Research by Marco Temperini et al., for example, applied learning styles, learners' knowledge and individual goals [21]. It is enhanced by taking into account learner profiles for composing a learning path recommendation [22]. Both research approaches offer an adaptive learning path for an individual learner and applied the classic model of adaptive learning systems.

III. Our Work

We design a graph-based domain knowledge and user model that records learners' learning progress. We apply Dijsktra algorithm to find the suitable shortest learning path for students based on student model and domain knowledge model.

A. Definition and Assumption

In order to have a similar understanding of the proposed model, several definitions and assumptions need to be clearly defined.

- A concept can be seen as a knowledge unit that must be learnt by students. Learning material is learning content linked to concepts. It could be lecture notes, exercises, examples, problems and the like.
- Knowledge is an understanding gained by students after learning one or more concepts. Knowledge of a certain concept can be a prerequisite for other concepts. We assume that the number of concepts is very large and each concept cannot be a prerequisite for itself.
- A learner is a user interacting with learning concepts in order to gain knowledge from an initial concept to a targeted set of concepts. We assume that learners who have learnt a concept will gain targeted knowledge through their interaction with learning materials. We also assume that a learner who has not fulfilled a set of prerequisite knowledge to learn a concept should not be allowed to learn the concept, because they will not achieve good results.
- Adaptive learning path is the most suitable shortest learning path for a student based on his/current knowledge current knowledge.

B. Graph-based Domain Knowledge Model

We have developed a prototype with a domain model for Data Structure, a course for Computer Science undergraduate students in three universities in Indonesia. It consists of 15 topics. We modelled the course in a graph G = (V, E) as a directed graph for domain model. Each node corresponds to a learning concept. Each node has a weight representing the time needed to learn the concept. For example, if the time allocated to learn concept *u* is more than the time needed to learn concept *v*, then the weight of *u* is more than the weight of *v*. The weights of nodes are shown in Table 1.

Two nodes are connected if there exists a prerequisite relationship between them. So that, an edge *e* = (*u*, *v*) means that having knowledge about concept *u* is needed to learn concept *v* or *v* is a post-requisite of *u*. We define a weight for each edge, *w*(*u,v*), which represents the connectivity strength between concept *u* and *v*, and influence score, *is(u,v)*, which means the advantage of having knowledge of *u* to learn *v*. We interviewed the lecturers teaching the course to identify connectivity between concepts. The connectivity between concepts is described in Figure 1.

TABLE I. CONCEPTS TAUGHT IN DATA STRUCTURE CLASS

#	Concept	Index	Percentage
1	Introduction	1.49	10.67%
2	Abstract Data Type	0.77	5.51%
3	Linear Linked List	1.25	8.93%
4	Double Linked List	0.36	2.58%
5	Circular List	0.36	2.56%
6	Multilink List	0.49	3.48%
7	Stack	0.73	5.20%
8	Queue	0.73	5.23%
9	Recursion	0.82	5.87%
10	Tree	2.73	19.46%
11	Binary Tree	1.12	7.99%
12	Heap Tree	0.66	4.68%
13	*Graph*	1.30	9.28%
14	DFS	0.60	4.27%
15	BFS	0.60	4.27%

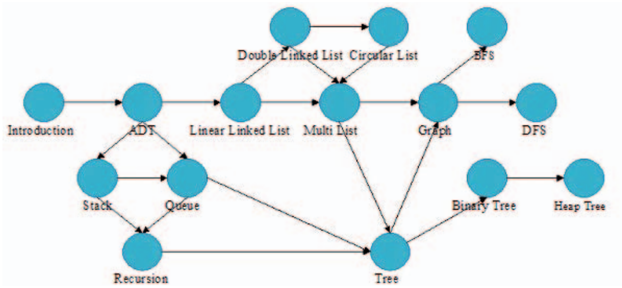


Fig. 1. The domain model of Data Structure course

We calculated edge weights by considering the weights of corresponding nodes The weight of edge *e*(*u,v*), which is the advantages of having knowledge of concept *u* to learn concept *v*, is calculated using the following formula:

$$w(u,vk) = \frac{d(vk)}{\sum_{i=1}^{n} d(vi)} \tag{1}$$

where *v1...vn* are targeted concepts that can be learnt after learning concept *u* or *u* is pre-requisite of concept *v1...vn*. The domain model graph after applying the formula to all nodes is as follows:

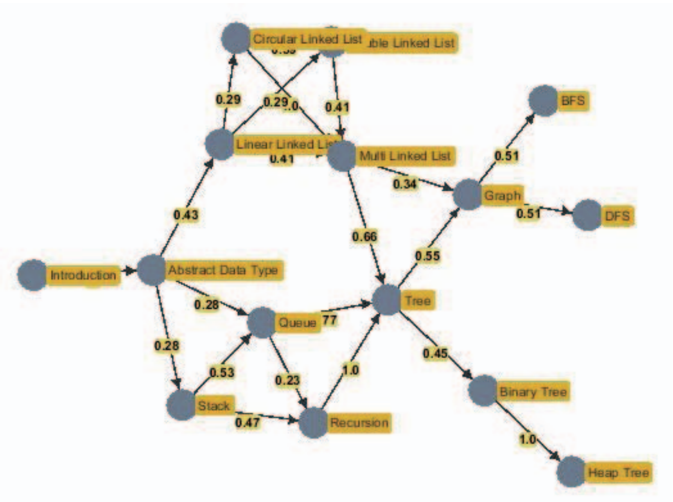


Fig. 2. Graph with edge weights which represent connectivity strength between concepts

C. User Model Inclusion for Dynamic Shortest Paths

Shortest path finding is important to inform students which concepts they must learn before learning a certain concept. The shortest path can be based on the influence scores that will result in the same path between two concepts for all students, because the scores are static all the time and for all students. In adaptive systems, for example adaptive learning systems, a user model is considered in recommendation or decision making. The attributes recorded in the user model can be varied; one of them is students' knowledge or competence. Since knowledge or competence is progressive, its inclusion in finding shortest paths will result in dynamic shortest paths adaptive to students' knowledge.

We applied parameters related to learners and domain knowledge for finding the most suitable and shortest path of learning to make it dynamic and adaptive to learners. The parameter applied is learners' knowledge of each concept which is between 0 and 1. A user model is represented in a vector:

$$K(u) = (k(v1), k(v2), k(v3), \ldots, k(v15)) \qquad (2)$$

$$0 \leq k(v) \leq 1$$

where $K(u)$ is a vector user (u) knowledge, $k(v1)$ is the student's knowledge on topic vi. Every time a student has mastered a concept, the influence scores between the concept and all its post-requisite concepts will be updated. For each $e(vi,vj)$:

$$is(vi,vj) = w(vi,vj) * k(vi) \qquad (3)$$

with $is(vi,vj)$ representing the influence of having knowledge on concept vi to learn concept vj, $w(vi,vj)$ is the connectivity score between concept vi and concept vj, and $k(vi)$ is the student's knowledge of concept vi. Since the student's knowledge is dynamic and different among students, the influence score combination is dynamic and unique for each student.

D. Modified Dijkstra Algorithm for Adaptive Learning Path

The function to find the most suitable shortest learning path in this research is inspired by the direction function provided in Google Maps. While Google Maps recommends the shortest path from a city to another city which is the same for all users, an adaptive learning tool can implement this function to find a learning path between two concepts based on students' knowledge. Hence, the learning path is called the most suitable shortest path. The path gives a student a view on how far they are from reaching a particular concept according to their current knowledge. The distance means concepts that must be learnt before learning the target. Our work is inspired by Google Maps and Dijkstra Algorithm, a classic shortest path finding algorithm.

Given the source node, vs, and the target node, vt, the shortest path finding algorithm tries to find a path

$$(vs, v1, v2, \ldots, vn, vt)$$

with minimum $|is(vs,v1), is(v1,v2), is(v2,v3),\ldots, is(vn,vt)|$. It consists of several steps, as follows:

1. Once a student has learnt a concept c, recalculate the influence score of each (c,vi).
2. For each concept node vi which becomes the post-requisite of some other concepts, including c ((c,vi) is exists), there exists $is(vj,vi)$, with $0 \leq j \leq$n. Normalize all $is(vj, vi)$ scores; this is called the success probability of vi, $sp(vi)$.
3. Dijkstra Algorithm takes the normalized $sp(vi)$ scores into account in determining the next concepts. Concept vi will be chosen if $sp(vi)$ is maximum.

In case a student does not yet have any learning experience, the algorithm will use weight scores which will give the same path to all students.

IV. Experiment and Result

We developed a tool prototype for finding the most suitable shortest learning path and designed two experiments. The first experiment aimed to prove that the domain model, user model and the shortest path algorithm will produce adaptive learning paths. Furthermore, we tested whether the tool could analyse different user models and produce an adaptive learning path or not. The second experiment aimed to get a view on whether the recommended shortest learning paths are suitable for learners. In this second experiment, we observed whether the learners followed the paths or not when they were given the freedom to navigate in hyperspace. In this paper, we discuss the first experiment and its results.

In this experiment, we invited a number of Data Structure class students majoring in Computer Science. The experiment started with a pre-test to build a user model, that is, the current knowledge participants. In real systems, user models will be gradually developed during the learning process; hence the pre-test is probably not necessary, unless teachers need to know the knowledge background of students. In the pre-test, participants were required to solve 15 problems, one each per concept. Once the pre-test had been done, the graph of the user's knowledge was updated. An example of a student model after the pre-test is shown in Figure 3.

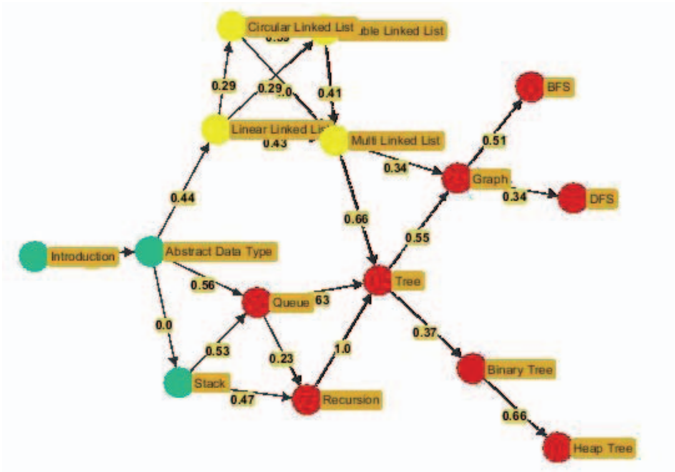
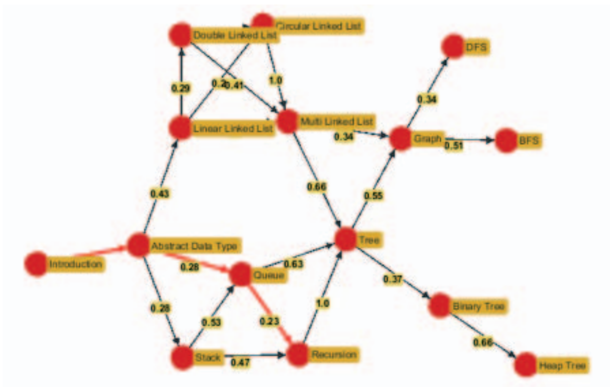


Fig. 3. Graph with connectivity strength

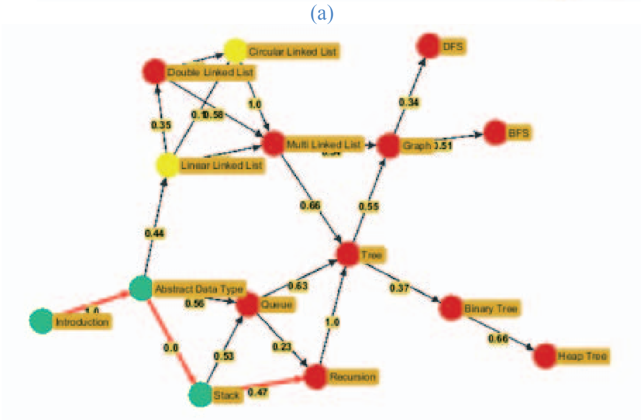
We compared the shortest paths recommended to different participants at an early stage of learning when participants had been just assessed from pre-test results. Furthermore, we compared the recommended shortest paths in the middle early stage of learning when participants had learnt some concepts.

1. *Comparison of learning paths in an early stage of learning*

   Figure 4 shows recommended learning paths given to two students at early stage of learning.



(a)



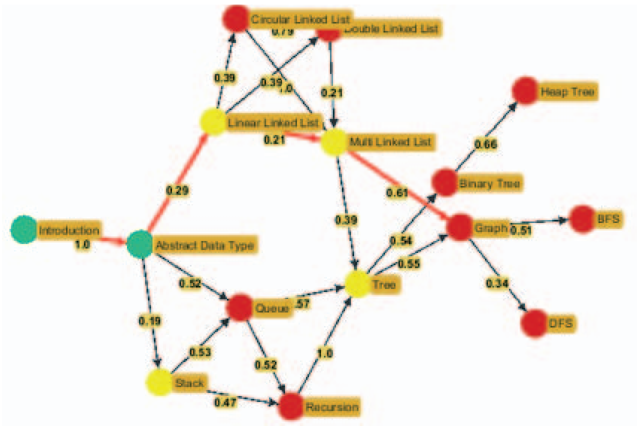
(b)

Fig. 4. (a) A recommended path for a student without any background knowledge (b) A recommended path for a student with some background knowledge

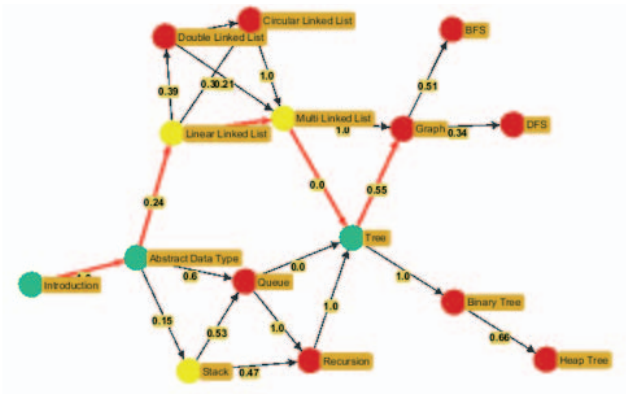
In this experiment, some students do not have any mastery of any material, while others have fully mastered a few materials and partially mastered a few other materials. The experiment shows the difference of learning paths recommended. In Figure 4, as regards user models, the green nodes are concepts that have been mastered by the students, and the yellow nodes are concepts that have been learnt by the students but not yet mastered.

2. *Comparison in the middle of learning when students have learnt some concepts*

   In this experiment, students have made some progress in mastering different concepts. The experiment aims to prove that the user model has updated and students have received different recommended paths based on their progress. Figure 5 shows different learning paths recommended to students with different progress.



(a)



(b)

Fig. 5. Recommended paths for students with different progress

From the first and the second experiments, the proposed algorithm can recommend adaptive learning paths that are considered the most suitable shortest path for students.

However, further experiments or expert review are needed to validate the results.

V. Conclusions

This research aims at improving the ability to select the most suitable shortest learning path for a learner. In order to achieve this, we apply a modified Dijkstra algorithm. The first step of our approach is to construct a graph-based domain model to represents learning concepts, prerequisite relationships and the weights of learning concepts in the course and connectivity score. The former is based on time that must be allocated for learning a concept in comparison with the whole time needed to learn the course, while the latter represents the connectivity strength between a concept and its post-requisites. The second step is to combine the weights of learning materials and the mastery level of learners which results in influence factors. It represents the influence of mastering one concept in mastering the next concept. The final step is to apply a shortest path algorithm to select an adaptive path which is suitable and shortest.

Three parameters are used, including learners' knowledge, the connectivity score and the influence score. It has been proved that the three parameters can produce different learning paths for different user models. Further experiments, however, are needed. Firstly, the algorithm must be applied to a larger network of concepts and applied to a larger number of students: thus the user models will be more varied. Further experiment is also needed to validate the recommended shortest paths.

References

[1] P. Brusilovsky, J. Eklund, and E. Schwarz, "Web-based education for all: a tool for development adaptive courseware," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1–7, pp. 291–300, 1998.

[2] P. Brusilovsky, "Adaptive hypermedia," in *Ten Year Anniversary Issue*, Kluwer Academic Publishers, 2001, pp. 87–110.

[3] A. Kavcic, "The role of user models in adaptive hypermedia systems," in *Electrotechnical Conference, 2000. MELECON 2000. 10th Mediterranean*, 2000, vol. 1, pp. 119–122 vol.1.

[4] D. Nurjanah, "Knowledge management in intelligent tutoring system," in *Indian International Conference on Artificial Intelligence*, Hyderabad, India, 2003.

[5] C. Fröschl, *User modeling and user profiling in adaptive e-learning systems: an approach for a service-based personalization solution for the research project AdeLE (adaptive e-Learning with eye-tracking)*. VDM Verlag Dr. Müller, 2008.

[6] A. C. Martins, C. V. de Carvalho, and E. Carrapatoso, "User Modeling in Adaptive Hypermedia Educational Systems," *Educational Technology & Society*, vol. 11, no. 1, pp. 194–207, 2008.

[7] P. Brusilovsky and E. Millan, "User models for adaptive hypermedia and adaptive educational systems," in *The Adaptive Web*, Springer Verlag Berlin, 2007, pp. 3–53.

[8] S. Graf, Kinshuk, and L. Tzu-Chien, "Supporting teachers in identifying students' learning styles in learning management systems: an automatic student modelling approach," *Educational Technology & Society*, vol. 12, no. 4, pp. 3–14, 2009.

[9] E. Siakas and A. . Economides, "Adaptive learning: mapping personality types to learning styles," presented at the INSPIRE, Thessaloniki, 2012.

[10] S. Bull and J. Kay, "Open learner models as drivers for metacognitive processes," *Azevedo V Aleven Eds Int. Handb. Metacognition Learn. Technol. Springer N. Y.*, no. 349–3659, 2010.

[11] S. Bull and J. Kay, "SMILI?: A framework for interfaces to learning data in open learner models, learning analytics and related fields," *Int. J. Artif. Intell. Educ.*, vol. v26, no. n1, pp. 293–331, Mar. 2016.

[12] M. Liang, J. Guerra, and P. Brusilovsky, "Building multi-layer social knowledge maps with Google Maps API," presented at the Workshop on Semantic and Adaptive Social Web (SASWeb 2012), Montreal, Canada, 2012.

[13] P. Brusilovsky, I.-H. Hsiao, and Y. Folajimi, "QuizMap: open social student modeling and adaptive navigation support with TreeMaps," presented at the The 6th European conference on technology enhanced learning: towards ubiquitous learning, Palermo, Italy, 2011.

[14] S. Sosnovsky and P. Brusilovsky, "Layered evaluation of topic-based adaptation to student knowledge," presented at the The Fourth Workshop on Evaluation of Adaptive Systems at the 10th International User Model Conference, Edinburgh, UK, 2005.

[15] Y. Gong, J. E. Beck, and N. T. Heffernan, "How to construct more accurate student models: comparing and optimizing knowledge tracing and performance factor," *International Journal of Artificial Intelligence in Education*, vol. 21, no. 1–2, pp. 27–46, 2011.

[16] W. Zhao and L. Wan, "A shortest learning path selection algorithm in e-learning," presented at the The Sixth International Conference on Advanced Learning Technologies (ICALT'06), Kerkrade, The Netherlands, 2006.

[17] A. Marwah and J. Riad, "A shortest adaptive learning path in eLearning systems: Mathematical view," *Journal of American Science*, vol. 5, no. 6, pp. 2–42, 2009.

[18] D. Guillaume, N. Belacel, and F. LaPlante, "Graph theory based model for learning path recommendation," *Information Sciences*, vol. 251, p. 10–21., 2013.

[19] P. . Henning *et al.*, "Learning pathway recommendation based on a pedagogical ontology and its implementation in moodle," in *DeLFI 2014 - Die 12. e-Learning Fachtagung Informatik*, Bonn: Gesellschaft für Informatik, 2014, pp. 39–50.

[20] brendan mumey, S. Yaw, C. Fastnow, and D. J. Singel, "Finding pathways to student success from data," presented at the The 7th International Conference on Computer Supported Education, Lisbon, Portugal, 2015.

[21] A. Sterbini and M. Temperini, "Adaptive construction and delivery of web-based learning paths," presented at the The 39th IEEE International Conference on Frontiers in Education Conference, San Antonio, Texas, USA, 2009, pp. 1223–1228.

[22] A. H. Muhammad, Q. Zhou, G. Beydoun, D. Xu, and J. Shen, "Learning path adaptation in online learning systems," presented at the 2016 IEEE 20th International Conference in Computer-Supported Cooperative Word in Design (CSCWD), United States, pp. 421–426.