An Architecture Model of Recommender System for Pedagogical Design Patterns

Nasrin Dehbozorgi
Department of Software Engineering
Kennesaw State University
Marietta, GA, USA
dnasrin@kennesaw.edu

Alan Norkham
Department of Computer Science
Kennesaw State University
Marietta, GA, USA
anorkham@students.kennesaw.edu

***Abstract*—This work in progress research-to-practice paper proposes an architecture model for active learning design pattern retrieval based on the concept of conversational recommender systems (C-RS). In earlier work, we proposed an object-based pedagogical design pattern model to generate abstraction of multiple unique implementations of collaborative active learning practices. This model helped in formalizing the problem-solution pairs that the educators face in collaborative active learning such as students' engagement in teams and class activities, social skill improvement, and assessment issues. Based on this model multiple patterns in different categories were developed through a systematic pattern development cycle combined with deductive elicitation methods and workshops. In this study, we propose an architectural model for an interactive pattern recommendation system. We use NLP models to extract the context of the problem that each pattern addresses. The extracted aspects are combined with the dimensions of each pattern based on attributes of the developed pattern model. The proposed architectural model is based on the model, view controller (MVC) pattern. In this model, the user interacts with the system through the 'view' layer. The 'model' layer stores the pattern language in the form of a graph in which each node of the graph represents one pattern, and the edges denote the relationship between the patterns. The 'controller' handles text analysis by NLP algorithms based on user input. One of the main features of our object-based design pattern model is its emphasis on team attributes by which different dimensions of collaboration in engineering education are addressed. The implementation of the developed architectural model would enable the instructors to identify possible solutions to the pedagogical challenges they face in teaching engineering classes where teamwork is an integrated part of the learning process.**

Keywords—Pedagogical design patterns, MVC pattern, Software Architecture, Recommender systems, NLP, Active learning, Collaborative learning

I. Introduction

Student-centered pedagogical practices have been applied by multiple educators in engineering education. One of the known forms of student-centered learning is collaborative active learning in which students mostly work in the form of low-stake teams [1],[4] to learn from each other and develop their interpersonal skills [1]. Active learning in engineering education improves students' computational thinking and problem-solving skills as well as long-term knowledge retention by 70% [2].

Successful implementation of active learning requires goal-oriented pedagogical practices that are developed based on learning theories and best practices [3]. On the other hand, the diversity of the existing practices makes it challenging to formalize and disseminate them. Pedagogical design patterns are similar to the concept of design patterns in software engineering and have been applied in the educational domain to formalize successful solutions to solve recurring problems [3].

Pedagogical design patterns enable educators to disseminate their expertise and design ideas in a formalized language and also provide a framework for comparing design decisions [3].

The commonality across most of the existing pedagogical design patterns is their narrative style which is based on Alexander's design patterns format [5]. The narrative structure of these patterns causes two problems: 1)it takes time to read it and decide if they are applicable in the given context, 2) finding the patterns that are most relevant to the context may require domain knowledge.

To address the first problem in earlier work [5] we proposed an object-based design pattern model with a metadata section and two levels of attributes. The multiple dimensions and attributes of the proposed model allow easier indexing and navigation through the patterns and led to the creation of a pattern language using concept maps which is a graph network of patterns [5]. The object-based model allows developing multiple objects of the same abstract pattern class which reduces redundancy and overlap as the number of patterns grows. Fig.1. shows the developed pattern model presented in the earlier work [5].

Although the developed model [5] reduced the amount of narration (text) in each pattern and made indexing and navigation through the patterns easier, still identifying the pattern that is most relevant to the context may be challenging for some educators or designers with limited domain knowledge about active learning.

Our findings show most of the existing repositories of design patterns are either available in the form of handbooks or books [5] or are accessible from online repositories [6]. We further identified that expanding the application of the existing patterns requires some platforms that help educators to identify the pattern (problem-solution pairs) quickly without having to review all available patterns one by one.

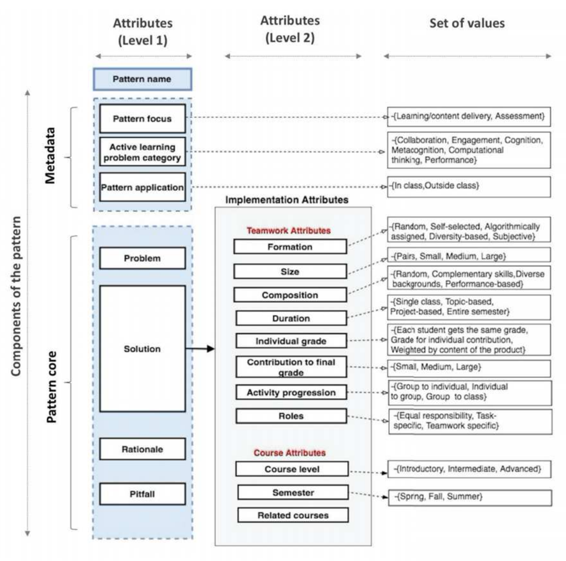

Fig. 1. Design Pattern Model[5]

The aforementioned observations motivated us to design a recommender system (RS) to help educators in decision-making for applying the patterns that best fit their needs. Most recommender systems require a large amount of data for training the model or otherwise they face cold start situations. In the context of design patterns, limited data is available about the patterns and their application for training the model. Therefore we need to apply the appropriate framework for recommending patterns that are independent of the size of existing data. Different types of RSs exist from which we select the Conversational Recommender Systems (C-RS) in this study.

The rest of this paper is categorized as follow: in section two we review the existing types of recommender system and the main features and drawbacks of each type are discussed, in section three we argue why we adopt C-RS, discuss its advantages in the context of pedagogical design patterns and propose our architectural model for design pattern recommendation system. Finally, we conclude with the plan for future work to develop the actual system.

II. Review of Recommender Systems

Recommender systems are a sub-class of information filtering systems that have been widely applied in multiple domains such as e-commerce, healthcare, and education. Although many different approaches have been developed to implement recommender systems, due to the complexity of this research topic and increasing demand for its application in diverse domains, they are still a rich research area. The focus of this study is not an in-depth analysis of existing RS models. However, we present an overview of the main categories of existing RSs, and base on the pros and cons of each type we rationalize our choice of C-RS for this study.

A. Content-Base Filtering(CB-RS)

In content-base filtering, an item's features are used to make a recommendation to the user based on the user's requirements [8]. In such systems, as the metadata about the user's requirement (i.e. the user's profile historic data) increases, the recommendations are more accurate and better suited for the user. In CB-RS, where no data is available from the new user, the recommender system initially recommends popular and highly rated items [8]. In these systems, when the RS elaborates on why a certain item is being recommended, it gives the user an understanding of the RS which will potentially develop user trust in the system over time [9]. However, one of the shortcomings of such systems is over-specialization which happens over time, where there is no novelty in the recommended items. Over-specialization is a problem in content-based filtering because the system tends to recommend items with only high ratings based on the user's history. Another drawback of content-based filtering is the reduction in the accuracy of recommendation due to the inability to extract all of the attributes of an item in reference to the user's intent. [10],[11].

B. Collaborative Filtering (CF-RS)

In collaborative filtering, the RS uses the collective interest of a group to recommend items. The idea is that groups that have similar backgrounds and interests, tend to consume the same type of items. This technique does not require content analysis the way CB-RS does since it depends on the group's interest [11]. In these systems, the lack of content analysis gives an opportunity for more diverse recommendations compared to CB-RS, helping alleviate the issue of over-specialization. CF-RS also has higher accuracy in recommendations due to its independence of content analysis. In these systems, as the number of data increases, the issue of scalability becomes a problem because of the computational requirements of the increased data, furthermore, to keep the user engaged, the RS needs to be relatively fast [12]. Another issue in these systems is unrated items and new users with no profile information [13]. These problems are known as sparsity and cold start respectively [9].

C. Context-Aware (CA-RS)

Context-Aware Recommender Systems add a third pillar to the user and item dimension. The system implements a contextual element in regard to the environment, time of day, and location. The accuracy and precision of the recommendation surpass CB-RS and CF-RS systems [9]. These systems heavily rely on machine learning methods to deal with large amounts of data, and the recommendations are still possible without explicit knowledge of contextual information. The contextual inferred data can be collected seamlessly through device sensors [9], but if the device does not contain sensors, capturing and using contextual data is difficult in CA-RS [14]. There is no standard for contextual data and the usefulness of the data depends on the domain and the context in which the system is used. [9]. Another issue in CA-RS is that contextual data are limited and the users' trust in the system affects the availability of the data [14].

D. Conversational Recommender Systems (C-RS)

Conversational Recommender System is a turn dialogue approach between the user and the system that consists of alternating responses. The system is typically either question and answer based or end-to-end conversations. Similar to other recommender systems, C-RS has its advantages and disadvantages. In C-RS, the recommendation is task-oriented and consists of a user and system alternating dialogue. C-RS is comparably a robust system since the dialogue management system can address user preferences, explanations, and process user feedback, which produces high accuracy and flexibility for recommended items [15]. C-RS can also be applied to multiple domains while the other recommender systems tend to be restricted to specific domains because the dialogue allows the user to ask a variety of topics without requiring much data about the user's history and preferences [16]. Cold start can also be alleviated because the system has the opportunity to clarify the intent of the user by asking a follow-up question. The data limitation is not a major issue in systems where contextual datasets are lacking since C-RS constantly builds the datasets through the user-system interaction.

There are also challenges and disadvantages of the C-RS to consider. Transparency of the user intents may reduce processing speeds in C-RS. These systems don't perform in real-time and need processing from the user's end. Since in the recommendation the depth is high and breadth is narrow, the chances of recommending items beyond user expectation or novelty are lessened. Interpretation of user input is an added task in these systems and requires additional resources for natural language processing, which might decrease the efficiency of such systems depending on input modality [17]. Since C-RS systems need to determine when to ask questions or make recommendations, the time factor can impact the availability of such systems [10]. All of the above factors contribute to the users' trust, acceptability, and satisfaction with the C-RS.

In this research, we adopt the C-RS due to its advantage of alleviating cold-start situations, performing with limited user data, and building up dataset via user interaction with the system.

III. Proposed Architecture Model

In this section, we propose our architecture model for building a pedagogical design pattern recommender system. The goal of this system is to help users identify the best patterns (i.e., solutions) that address the problems they face in teaching active learning classes. The users of this system would be instructors and educators who are interested in implementing different student-centered pedagogical practices in their classes.

The architecture model of is system is based on the Model, View, Controller (MVC) pattern. MCV is one of the most common design patterns applied in software systems, in which the emphasis is on human-computer interaction [19]. This pattern provides high cohesion by separating the functionalities of the system and maintains low coupling, which addresses the linkage problem between system components and maximizes the usability of the system.

This pattern divides the system into three components that are dependent and connected together, and present the data to the user based on the user input into the system. Fig.1. visualizes how different components of the MVC pattern interact with each other.

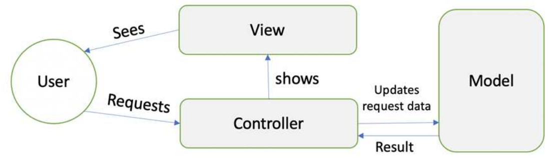

Fig. 2. Components of the MVC pattern

The 'model' component encapsulates the domain-specific knowledge, state of the application, and operations that change the states [18]. The view component presents the data to the user through a user interface and the controller passes the transactions to the model for execution.

The heart of C-RS is the dialogue management system that interprets the user input, sends it to the processing and reasoning unit (model), and finally presents the output to the user.

In our proposed system, the user interface, or 'view', prompts the user to select and input the key terms that most correlate with the problems for which they seek to find the solution. The user input is passed to the dialogue management system, which is the 'controller'.

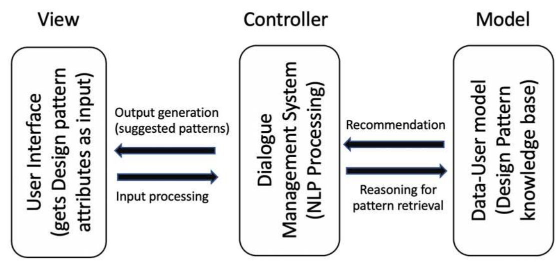

Fig. 3. Logical viwof the proposed architectural design

The role of the dialogue management system is to process the user input through NLP methods and pass it to the 'model' of the system. These terms are analyzed with respect to the meta-data, dimensions, and attributes of the pedagogical design patterns we identified in the previous study based on the existing theories of cognition and social constructs that impact performance in the educational setting [5],[3]. Some examples of these attributes are low-stake teams, mini-lectures, class activity, pair programming, assessment, etc. The 'model' includes the knowledge base, user modeling system, and reasoning engine which identifies the appropriate pattern to be delivered to the user. Fig. 3 shows the logical view of the proposed architectural design.

IV. Conclusion and future work

During the past decade, many researchers have focused on the cognitive theories and how the integration of social construction of knowledge in collaborative environments can lead to higher performance or retention rates [3]. These novel studies led to emerging new ways of content delivery and student-centered learning in educational settings.

Pedagogical design patterns were introduced as a means to formalize the emerging practices. Many educators, designers, and higher education institutes have contributed to developing numerous pedagogical design patterns. The developed patterns are mostly available on online repositories or published in different venues. The large variety of the existing patterns makes accessibility, pattern selection, and retrieval process time-consuming for the users. The focus of this study is how the extensive previous research can be applied and practiced in engineering education in an effective manner. We suggest the conversational recommender systems as a tool to cue educators implement novel practices in engineering education.

In this study, we propose a conversational recommendation system architecture based on the MVC pattern to recommend the patterns to the educators. MVC pattern is one of the most common software design patterns that lead to the creation of architecture with high cohesion and low coupling between components. The application of this pattern also helps in developing an effective user interaction experience. Given that our proposed system would be based on the C-RS user interaction plays a big role in the system process and performance and thus the MVC would be a good fit for this system.

The plan for the near future includes the following phases: 1) Identification and selection of the problem areas that the educators face in engineering active learning classes, 2) selection of a set of design patterns that are most commonly used in each problem domain, 3) Running NLP methods on the existing patterns to extract the context of the patterns for the indexing purpose. This step will also include the application of the meta-data attributes in our developed object-based pattern model to tag the context of the patterns, 4) Development of the dialogue management system by using NLP algorithms which is the core of the C-RS, and 5) Developing the model, view, and controller layers of the system.

At the last step we will be evaluating the accuracy and effectiveness of the developed system. The developed recommender system will be applied by two educators of the software engineering classes who are familiar with the active learning methods as well as pedagogical design patterns. Both quantitative and qualitative data analysis based on the educators' feedback will be applied to measure the effectiveness of the application of this system and improvements will be applied accordingly.

We believe the development of such a system would enhance the application of pedagogical design patterns and creates a platform to exchange design ideas in a structured format. This will have a positive impact on facilitating the application of novel pedagogical practices in engineering education.

REFERENCES

[1] Dehbozorgi, N., Maher, M. L., & Dorodchi, M. (2020, October). Sentiment analysis on conversations in collaborative active learning as an early predictor of performance. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

[2] Dehbozorgi, N. (2017, August). Active learning design patterns for CS education. In Proceedings of the 2017 ACM Conference on International Computing Education Research (pp. 291-292).

[3] Dehbozorgi, N. (2020). Sentiment Analysis on Verbal Data from Team Discussions as an Indicator of Individual Performance (Doctoral dissertation, The University of North Carolina at Charlotte).

[4] Dehbozorgi, N., & MacNeil, S. (2019, October). Semi-automated Analysis of Reflections as a Continuous Course. In *2019 IEEE Frontiers in Education Conference (FIE)* (pp. 1-5). IEEE.

[5] Dehbozorgi, N., MacNeil, S., Maher, M. L., & Dorodchi, M. (2018, October). A comparison of lecture-based and active learning design patterns in CS education. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-8). IEEE.

[6] Maher, M., Dehbozorgi, N., Dorodchi, M., & Macneil, S. (2020). Design Patterns for Active Learning. *Faculty Experiences in Active Learning: A Collection of Strategies for Implementing Active Learning Across Disciplines*, 130-158.

[7] Dwivedi, S., & Roshni, V. K. (2017, August). Recommender system for big data in education. In 2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH) (pp. 1-4). IEEE.

[8] Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. Recommender systems handbook, 73-105.

[9] He, C., Parra, D., & Verbert, K. (2016). Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. Expert Systems with Applications, 56, 9-27.

[10] Jannach, D., Manzoor, A., Cai, W., & Chen, L. (2020). A survey on conversational recommender systems. arXiv preprint arXiv:2004.00646.

[11] Cui, L. Z., Guo, F. L., & Liang, Y. J. (2018, October). Research overview of educational recommender systems. In Proceedings of the 2nd International Conference on Computer Science and Application Engineering (pp. 1-7).

[12] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009.

[13] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web (pp. 291-324). Springer, Berlin, Heidelberg.

[14] Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., & Duval, E. (2012). Context-aware recommender systems for learning: a survey and future challenges. IEEE Transactions on Learning Technologies, 5(4), 318-335.

[15] Barria-Pineda, J. (2020, July). Exploring the Need for Transparency in Educational Recommender Systems. In Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (pp. 376-379).

[16] Narducci, F., Basile, P., Iovine, A., de Gemmis, M., Lops, P., & Semeraro, G. (2018, October). A Domain-independent Framework for building Conversational Recommender Systems. In KaRS@ RecSys (pp. 29-34).

[17] Yanes, N., Mostafa, A. M., Ezz, M., & Almuayqil, S. N. (2020). A Machine Learning-Based Recommender System for Improving Students Learning Experiences. IEEE Access, 8, 201218-201235.

[18] Grove, R. F., & Ozkan, E. (2011, May). The MVC-web design

[19] Feng, X., Liu, W., & Jiang, Y. (2020, June). A model-driven system interface design method based on MVC pattern. In Journal of Physics: Conference Series (Vol. 1549, No. 4, p. 042075). IOP Publishing.