IJE TRANSACTIONS A: Basics Vol. 35, No. 07, (July 2022) 1317-1329

International Journal of Engineering

Journal Homepage: www.ije.ir

# A Compositional Adaptation-based Approach for Recommending Learning Resources in Software Development

M. Tayefeh Mahmoudi*[a], K. Badie[b], M. H. Moosaee[c], A. Souri[d]

[a] *Data Analysis & Processing Research Group, IT Research Faculty, ICT Research Institute, Iran*
[b] *E-Content & E-Services Research Group, IT Research Faculty, ICT Research Institute, Iran*
[c] *Computer Engineering Group, Science & Culture University, Iran*
[d] *Electrical Engineering Group, Islamic Azad University South Tehran Branch, Iran*

*PAPER INFO*

*Paper history:*
Received 01 March 2022
Received in revised form 15 March 2022
Accepted 02 April 2022

*Keywords*:
*Recommender System*
*Learning Resource*
*Case–based Reasoning*
*Compositional Adaptation*
*Semantic Similarity*
*Domain-specific Ontology*

*ABSTRACT*

In this paper, we discussed the application of a compositional adaptation approach to recommend learning resources to users in the area of software development. This approach makes use of a domain-specific ontology in this area to find those words, which are used in the technical description of the stored cases. A point peculiar with representing cases in the proposed approach is to take into account the characteristics of included learning resources, which justify the way they support the essential operations in the case of solution. In this way, only those components that comply with user's request would be considered in the final solution. In the paper, the performance of the proposed approach for recommending learning resources together with the status of user experience in his/ her interaction with the resulted recommending system, have been evaluated. Results demonstrate the fact that the learning resources through this approach are sufficiently beneficial for the users. Although the proposed approach has been applied for recommending learning resources in the area of software development, it can be equally applied to any technological area through developing domain-specific ontology for that area. This is mainly because any technological area has its own specific objects/ entities holding their own semantic similarities that finally lead to forming a domain-specific ontology for that area.

***doi**: 10.5829/ije.2022.35.07a.11*

## 1. INTRODUCTION

In recent years, there has been a significant increase in the number of learning resources in various areas, especially in areas related to design, implementation and development of a software product. This huge amount of learning resources in the internet has been created due to the growing need of developers and the rapid growth of technology in this area. The question is which parts of these resources are essential for the developers, which of them yield a better education in practice while more functional, and finally which resource is more suitable for starting or continuing the learning process according to the level of the learner's knowledge. Therefore, choice selection turns to be difficult for the learners in general, and this causes difficulties on going from one resource to other resources. Previously, traditional systems used to provide static and non-intelligent or semi-intelligent algorithms. But modern systems namely intelligent tutoring systems, make use of information such as behavior, feedback or model of users, try to increase the quality of their suggestions and promote users' knowledge, skills and engagement as well [1, 2]. Taking the above-mentioned points into account, in this paper a compositional adaptation approach to case-based reasoning is proposed to upgrade recommendation of learning resources within the scope of project development as a significant scope of activity. To retrieve appropriate cases, a sort of ontology is employed which helps find cases that are semantically similar to what users express in terms of technical requirements for implementing a particular software application. Since a

*Corresponding Author Email: mahmodi@itrc.ac.ir (M. T. Mahmoudi)

Please cite this article as: M. Tayefeh Mahmoudi, K. Badie, M. H. Moosaee, A. Souri, A Compositional Adaptation-based Approach for Recommending Learning Resources in Software Development, ***International Journal of Engineering, Transactions A: Basics,*** Vol. 35, No. 07, (2022) 1317-1329

problem situation may be complex, educational resources for software development projects need to be combined in order to yield potential solutions. Compositional adaptation is therefore believed to be a helpful means in this regard.

## 2. RELATED WORK

Nowadays, recommender systems have received remarkable attention in social networks [3] educational institutions, with particular emphasis on e-learning and online learning, especially in the area of computer science [4]. These systems have been developed to directly help learners and also instructors choose useful and relevant learning resources such as courses or exercises based on deep reinforcement learning [5]. Some systems have also been developed to meet educational requirements among the massive number of educational resources [6]. Moreover, recommender systems useful for technology enhanced learning may support and enhance learning practices in design, development and test of socio-technical innovations [7]. These systems are able to intensify teaching and practicing based on a variety of reasoning methods [8]. There also exist some hybrid recommender systems, which are adaptive to learner's preferences and are able to generate recommendations by some hybrid approaches such as content-based filtering, collaborative selection and opinion mining as well [9-11]. Another kind of hybrid recommender system also exists, which makes use of ontology and sequential pattern mining (SPM) to evaluate the domain knowledge of the learner and learning resources, and determine the learners' consecutive learning patterns [12]. Moreover, there are some adaptive/intelligent web-based educational systems, which, not only focus on adaptive presentation and navigation, but also provide intelligent solutions for problem analysis and solving as well as curriculum sequencing [13]. In this respect, a ITS named Fuse has been developed based on both fuzzy and semantic reasoning to provide learning recommendations adaptively [14]. In this way, the user-centered design has shown to be a suitable basis for developing various kinds of learning recommenders. Regarding that, supporting adaptive feedback based on students' offline information and online resources, should also be regarded [15]. These studies demonstrate that making personalized environments in learning recommendation systems is the process that must consider learners' requirements respecting the learning contexts.

Most of the aforementioned systems employ rule-based or filtering methods to recommend personalizing the contents/learning resources [16] whereas situations exist that pre-experienced cases can also help recommendations come true more effectively. In this regard, various algorithms of case-based reasoning seem to be helpful [17-19].

A brief review on employment of CBR in educational systems shows that it is, not only applicable for suggesting suitable courses of massive open online courses (MOOCs) [20], but also is useful for planning towards personalization [21], contextualization [22] as well as recommendation [23, 24] and intelligent assistance [25]. Here, case adaptation is believed to have a promising role in providing reasonable solutions. Accordingly, utilizing an appropriate compositional adaptation method may play a significant role, especially in the situations where the components of a solution are capable of being adapted separately [26], or vice versa, where a solution is not dividable into some independent components and is therefore to be combined with other solutions. Although a variety of compositional adaptation algorithms have been proposed [19, 23]. However, they seem not to be adequate in the situations where a kind of semantic similarity exists between a problem and the stored cases. Considering this point, we propose a new approach for recommending the educational resources that has the ability to consider this semantic similarity in some way.

## 3. THE PROPOSED APPROACH TO DEVELOPING CBR-BASED RECOMMENDER SYSTEM

**3. 1. Basic Idea** By a CBR-based system, we mean any system, which functions on the ground of case-based reasoning, which has the ability to provide solutions for new problems based on the similarity which does exist between the current problem situation and the problem situations experienced in the past. The peculiarity of case-based reasoning in general, is to provide additional facts for a problem situation based on the cases experienced in the past. In many cases, the components of the problem situation and those in the stored cases are entities with at- one- glance different meanings thus making the issue of case adaptation rather problematic. However, the possibility exists in such a situation that the components chosen from different cases may be incompatible with each other thus increasing the risk of obtaining a non – reasonable solution. To overcome such a problem, in our approach certain constraints are to be included in the cases to assure the validity of a solution at different stages of composition. Solution in a case comprises a set of components as well as the functionality, which is to be exhibited by it. It should be noted that each component in a case itself holds a characteristic with respect to another component in the case whose information can increase the assurance of a solution's validity in some way. In this way, the higher number of such characteristics, a higher expectation would exist for the validity of a case solution.

Taking the above discussion into account, status of similarity between a problem situation and the existing case may be assessed from both aspects of "functionality" and "components". "Functionality" is to figure out which cases are to be taken into account as cases, and "components" to find out due to which shared properties the components in a case can be linked to those in the other cases [27]. Once cases with high priority were figured out, it would then be necessary to find out first how their components can be regarded similar to those in the problem situation and then take into account the very relations, which are considered between the components in these cases. To provide such types of information, ontology-type structures developed for specific problem domains are of particular significance. With regard to an ontology-type structure, entities in terms of classes and individuals (instances or objects) are connected to each other through considering some relations which are of particular significance to the corresponding domain (Software Development in our case). These relations are basically derived from the nature of Software Development, which in turn calls for concepts such as "implementation", "target for", "kind of", "tool for", …, which are of high significance to it. Let us say incorporating such concepts as relations (between classes and individuals) provides a suitable medium for us to arrange the popular items of interest in such a way that their similarity can be found out in a reasonable manner. Obviously, greater the number of relations as well as the depth of this ontology, a higher expectation would exist for the possibility of justifying similarity between a component in the current problem and some component in a stored case. It is to be noted that, despite the fact that the complexity emerged in such a way may at one look be an obstacle to finding similarity, there are however some cases (though not so frequent) where the status of similarity between the components may not be so clear. Thus in order not to miss the chance of retrieving a relevant case, we naturally would need an ontology with higher complexity.Regarding this we should keep in mind having an ontology with high complexity doesn't necessirily mean to afford high computational cost for finding similarity for all the problem situations. What we intend to do is to provide a medium wherein alternative component in a stored case can be found for a component in the current problem through figuring out the connection which does exist between them within this ontology-type structure. Once such a connection was found, we may conclude that the two components are semantically similar to each other. Having found those components in the stored cases, they would then be worth being transferred to the problem situation as the complementary components, and in the meantime those irrelevant components that are not compatible with the requirements of the problem situation can be deleted. To keep the final solution as suitable as possible, only one component from each retrieved case ought to be added to the problem situation. It should be noted that, to validate the considered ontology and its impact on the soundness of the final results, the meaningful relations discussed above would with no doubt be of high significance, which are defined by specialists who make the ontology based on the basic knowledge of domain specific ontology and the semantic labels defining the relations between the existing nodes. The pseudo-code of the proposed algorithm is also presented in Figure 1.

**3. 2. An Example** Suppose that a user wants to develop "A Social Network Application on Mobile Platform for Consulting on Type of Restaurants in Specific Map Location". Here, situation in a case comprises a "title" with technical content, and a "technical description" with regard to the utility of this title. In the meantime, solution of a case comprises a number of components standing for appropriate learning resources, which can support the operations that are essential to realizing the case situation. Such a support is in fact a characteristic of a learning resource (as a component) which justifies how it can help a certain operation be performed in a favourable manner.

```
Procedure SEMANTIC_CBR
Input: pre-stored cases, problem situation
Set significant categories, properties or characteristics of ongoing problem situation
for each combinations of significant categories, properties or characteristics, P
  for each pre-stored case, C
 Determine semantic similarity between the components of C and P
      for each constraints S in constraint list
        if any component of C satisfy constraint S then
          if any component of P does not satisfy constraint S then
            Add the complementary component to the list of P
          end if
            Delete constraint S from the list of constraints
          end if
        end for
  end for
end for
print possible alternatives of the final solution
end procedure
```

**Figure 1.** The pseudo code of proposed algorithm for recommendation based on semantic compositional adaptation

With regard to the case retrieval process, we have to see, which cases out of those stored in the library, can support this request in some way. In our approach, in order to retrieve a case, there should exist at least one item in its "technical description" which can be identical to a keyword in the user's request. Suppose that, none of the cases has such an ability. It would now be important to check whether some words exist in the domain – specific ontology developed for this purpose, which can be semantically similar to the user's request keywords in some way. It should be noted that a domain-specific ontology has the ability to replace the keywords of the user's request with those words in the "technical description" of a case, which, though not exacly the same are semantically similar to them. To construct a domain-specific ontology, significant aspects of the related technology covering issues related to "infrastructure", "methods", "tools" as well as "applications & services" can be taken into account. Considering this point a part of the domain-specific ontology used for this example is shown in Figure 2. A domain specific ontology can generally be made based on the consensus obtained among the specialists of that domain on the one side and the basic knowledge of that domain on the other side. In our research we followed this procedure.

Coming back to the above example, and supposing that the request keywords ("mobile platform", "consulting" and "map location") have not been directly used in the "technical description" of any stored case, the ontology of Figure 2 would tell us that there exist the words"android", "chatting" and "location", which respectively correspond to these keywords and are able to justify retrieval of the cases of Figures 3(a), 3(b), 3(c), since these terms ("android", "chatting", and "location") have been fairly used in their descriptions. Having retrieved these cases, it would now be important to check which parts in the corresponding solutions can be considered as the alternative parts for the final solution on the related request. With regard to the case of Figure 3(a), learning resources on "Java Language", "Java

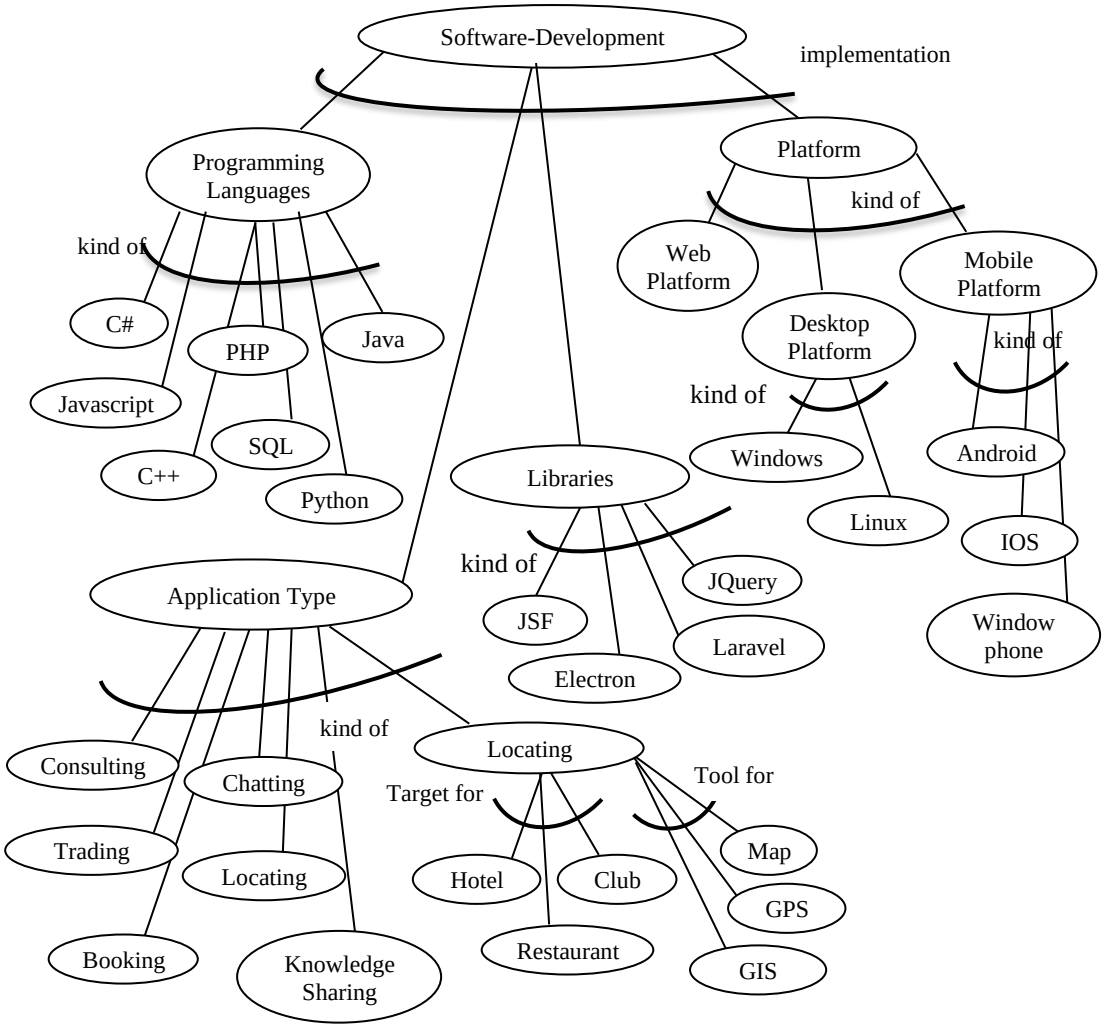


**Figure 2.** A part of domain –specific ontology used in the example

Language", "SQLite Database Manager", "XML Markup Language" and again "Java Language" can take such a responsibility, since they serve "Implementing Business Logic for Android", " Implementing Database for Android" and " Implementing User Interface for Android" which comply with the user's request. Our basis for reasoning is to take into account the very similarity which does exist between the keywords in the "user's request" on the one side, and the keywords existing in the situation of the corresponding case on the other side, which is actually realized by means of "domain specific ontology" discussed here.

However there exists no need for learning resources on "Soap Protocol", "Magneto Framework" and "MySQL Database Manager", since they have got nothing to do with the requirements in the request of Figure 3(a). In our approach, the so-called requirements are with regard to the main keyword in the user's query; "Mobile platform" in this case which comprises entities such as "Java Language", "Java Language", "SQLite Database Manager", "XML Markup Language" and again "Java Language". These entities can be used as active resources for learning "Implementing Business Logic for Android","Implementing Database for Android" and "Implementing User Interface for Android" as illustrated in Figure 3(a). In the same manner, some parts in the cases of Figures 3(b) and 3(c), as highlighted, are considered as the parts for the final solution. As the result, set of "learning resources" for the related request would be "Java language" , "SQLite

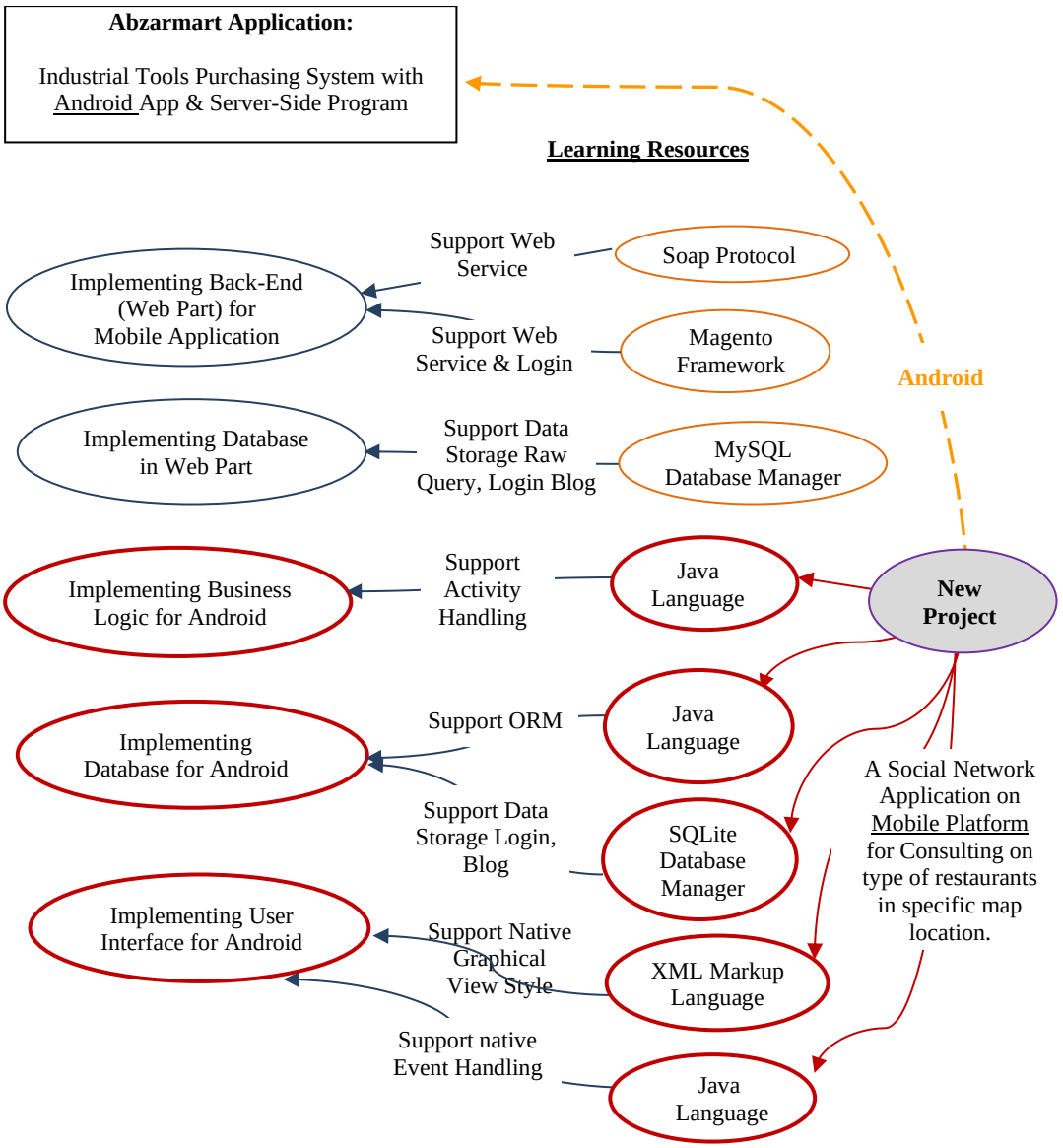


**Figure 3(a).** An example on compositional adaptation about implementing Android Application

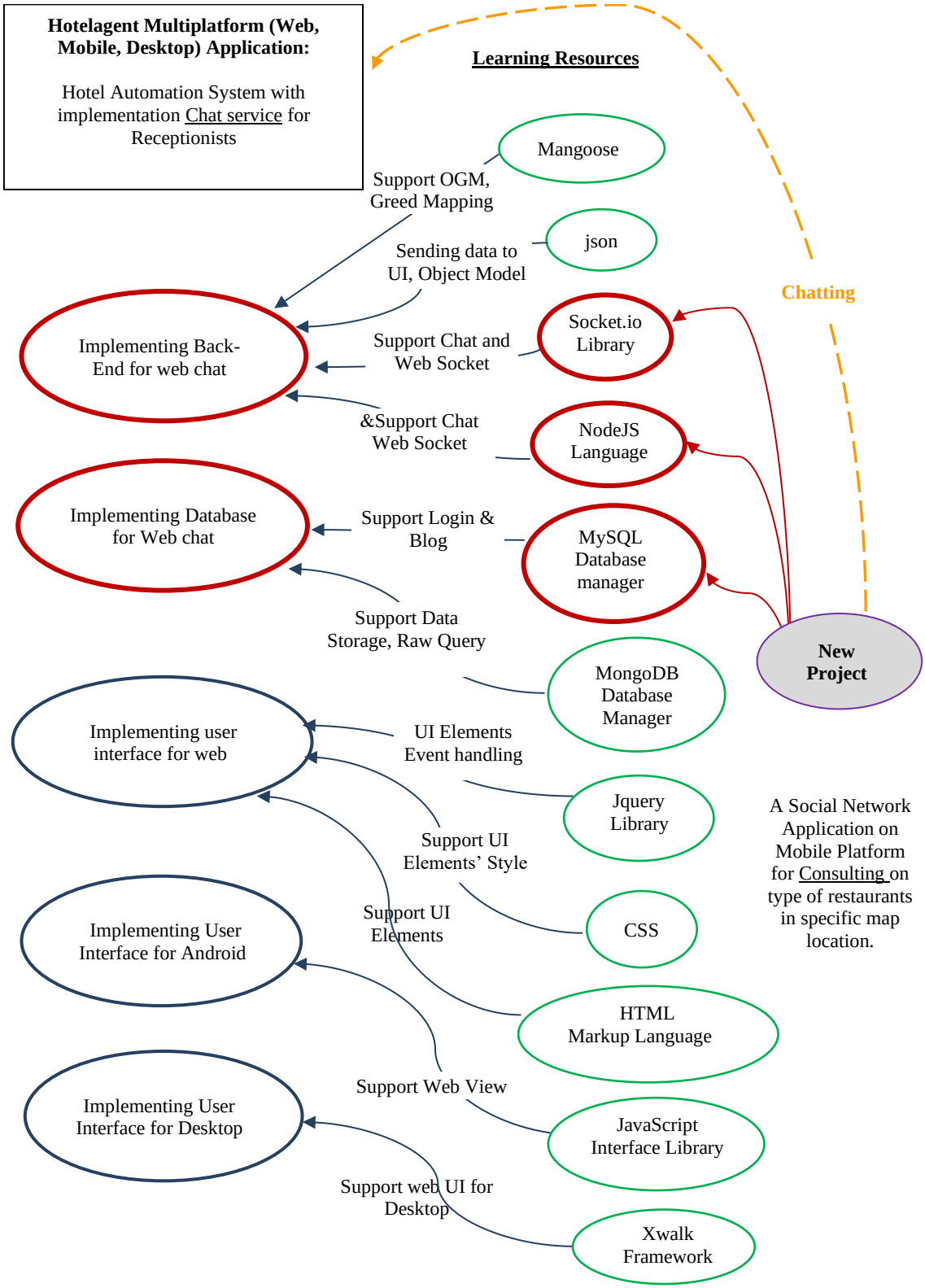


**Figure 3(b).** An example on compositional adaptation about implementing Communicating by Chat Services

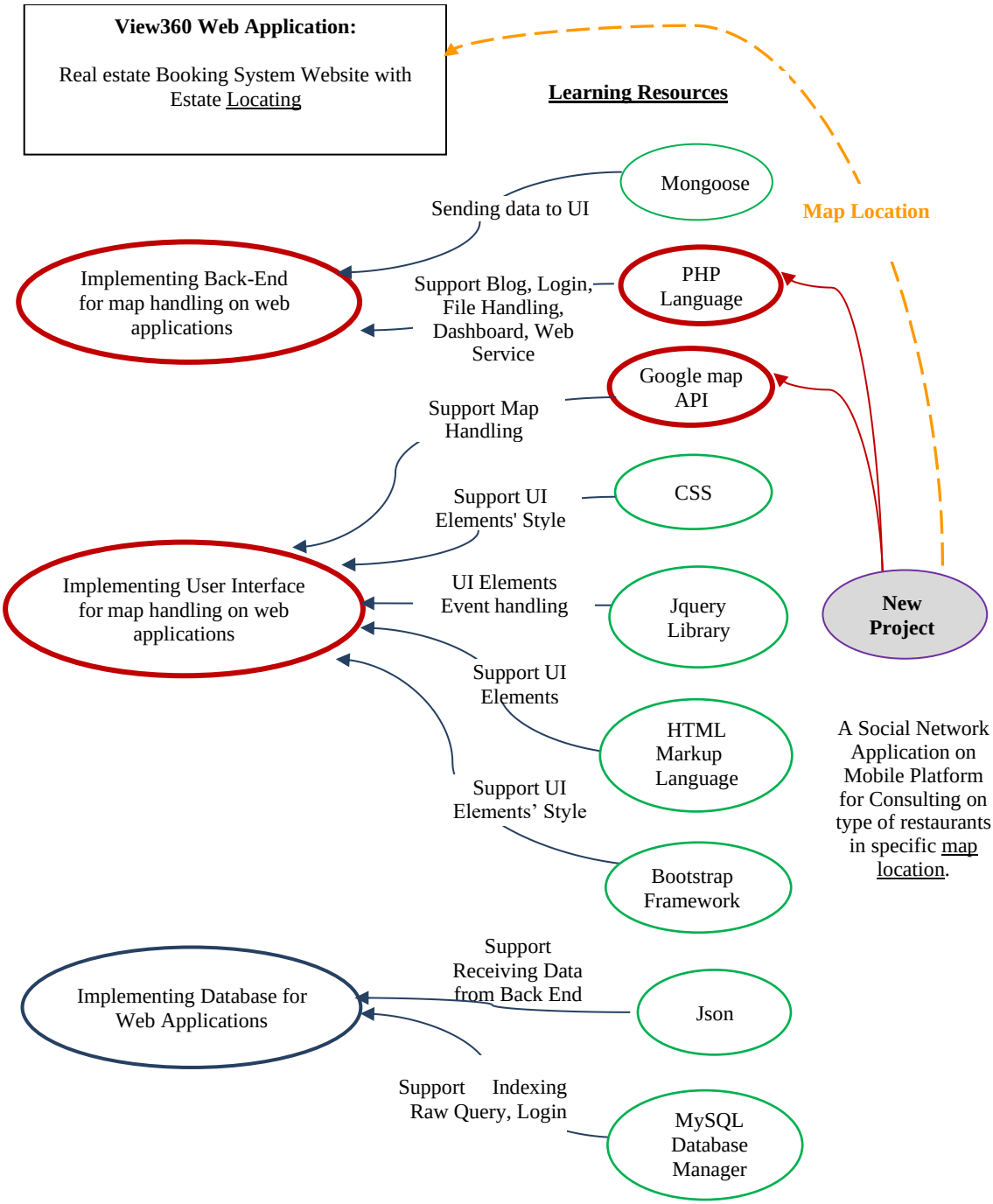


**Figure 3(c).** An example on compositional adaptation about implementing Map Location

Database Manager", "XML Markup Languge", "NodeJS Language", "My SQL Database Manager", "PHP Language" and "Google Map API". For instance, “Java Language” can be applied for supporting activity handling towards implementing business logic for android or for supporting ORM towards implementing database for android or for supporting native event handling towards implementing user interface for android.

## 4. EXPERIMENTAL EVALUATION AND RESULTS

To evaluate the impact of the system on the recommendation of different learning sources, the system has experimented on real users with academic education in the field of computer science. In particular, more than 100 computer science graduated students, junior developers in the field of software programming, took part in this experiment. After a short training, each

participant worked with the system at least 15 minutes, and then filled in the UEQ questionnaire which was designed by Laugwitz, et al. [28] and has been known as a standard tool for assessing various scales of user experience in different research works.

As Manouselis et al. [7] mentioned the qualitative comparison of e-learning recommender systems is very difficult due to their strong dependence on context. Therefore, the main purpose of our experiment is to determine the accuracy of the diagnosis and prediction of resources required by the user. Users' satisfaction measured by the corresponding section designed in the system is also evaluated. Each set of the experiments is calculated by grouping the users, ten by ten, based on their time order. The average value taken for each of these groups is presented in Figures 4 and 5.

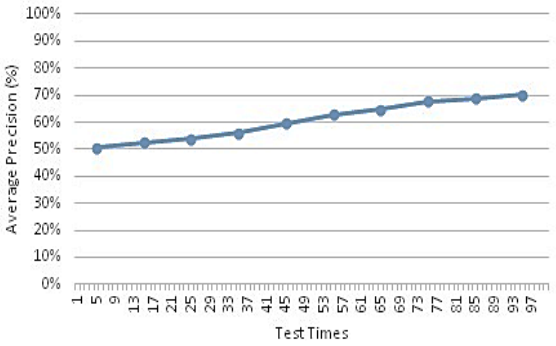


**Figure 4.** Average precision change over test times

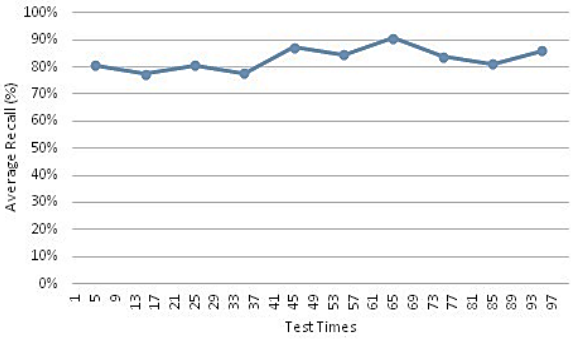


**Figure 5.** Average precision change over test times

**4. 1. Performance Measure** To evaluate the performance of the proposed recommender system, precision, recall, and f-measure were considered. The reason for using such factors goes back to the retrieval nature of the proposed CBR-based recommender system which calls for selecting those factors which can be responsible for validating a retrieval process. In this regrad, two ways usually exist for identifying the relevancy of the items in a recommender system: 1) applying pre-tagged datasets, 2) receiving the feedback of the user, which in our case, the second way was examined.

Precision is, in fact, is a measure to evaluate the accuracy of data retrieval, which in this experiment results in a ratio of the number of learning materials resources regarded as relevant (by the user) to the total number of learning materials recommended by the system. Figure 4 reveals the slight slope of the precision by increasing the number of tests. As the numbers of system users and appropriate stored cases increase, the precision is also increased. When there are more stored cases in the system, more possibility would exist to have similar cases with the problem situation, and as a result, more comprehensive solutions for the problem's constraints. So, due to the existence of more similar cases, components causing errors would hold lower priority, and would therefore have less effect on the formation of the final solution. In this way, the accuracy of the system will ultimately be increased. On the other hand, the recall is a measure to evaluate the number of related items that are properly retrieved, and the result would therefore be equal to the ratio of the number of correct learning materials resources retrieved to all the related learning materials resources. Regarding Figure 5, due to the fact that the number of recommended references highly depends on the problem posed by the user, unlike the usual situations wherein the recall index is used to evaluate the comprehensiveness of the retrieved results, the total number of related resources in the system is expected to be limited. Therefore, the recall measure would have no significant changes, and would always be in a range of 0.75 to 0.90.

In such a case where these two criteria are close to each other, in order to better assess the system, another criterion called F-measure, is considered to take care of the harmonic mean of precision and recall (Figure 6). According to Figures 4-6, through increasing the precision and stability of the recall criterion, F-measure increased as the combination of these two.

Another measurable and essential criterion in such systems is the time, or in other words, the accessibility to various resources and presenting them as the final solution. There are several factors in determining the retrieval time in this system: the number of resources received per user or per problem situation's constraint, the number of keywords in the user's request or words that are semantically related to the resources in the system, and total number of stored cases in the database which are generally reflected during the retrieval and representing of the final solution (Figures 7-9).

Figure 7 shows the increase of retrieval time when the number of retrieved sources for each keyword or each constraint in the problem situation is increased. For example, if the system recommends two different learning resources for one semantically meaningful element in the problem situation, it takes 151 milliseconds to find and retrieve these resources. It is obvious that the number of semantically connected

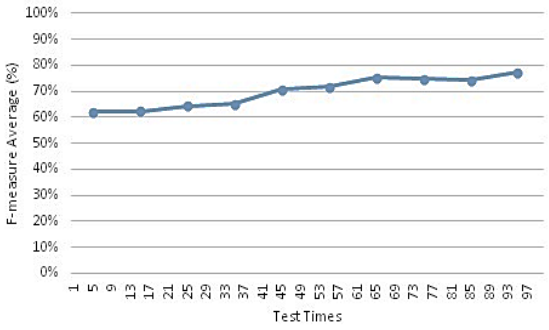


**Figure 6.** F-measure average change over test times

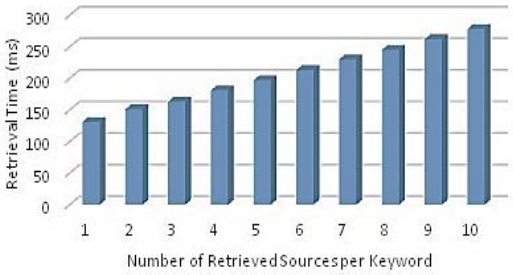


**Figure 7.** Average precision change over test times

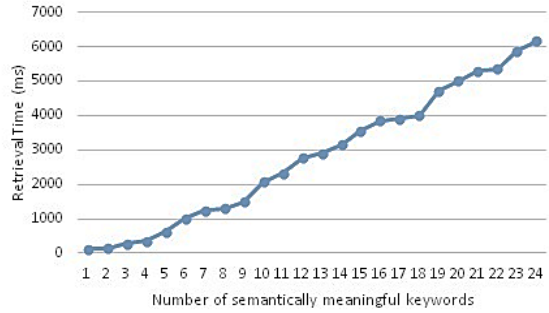


**Figure 8.** Retrieval time by keywords count

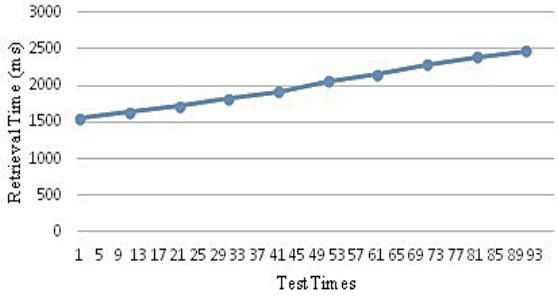


**Figure 9.** Total retrieval time over test times

keywords in the problem situation would affect directly the retrieval time.

According to Figure 8, when the number of semantically meaningful keywords in the problem situation is increased, it would take much more time in order to represent connected resources to these keywords as a solution. For example, when the problem situation contains 13 keywords, which are semantically connected to different sources, it takes about 3000 milliseconds for retrieving the connected resources and representing them to the user as a final solution.

Regarding Figure 9, and assuming that the other factors in this title are normalized, we notice that the total retrieval time has been increased over the test times mainly because of an increase in the stored cases as the basis for search. It is seen that this increase occurs linearly, because of using an ontology for finding similar cases. Let say, within-ontology search for this purpose causes the system to check each pre-stored case in order to find the most similar cases to the problem situation.

**4. 2. User Satisfaction Measure** There is a specific element in the user interface of the system that announces user's satisfaction degree towards the presented solutions. In this regard, users are grouped ten by ten based on the corresponding time orders. The average of satisfaction in each group is shown in Figure 10. According to Figure 10, user satisfaction seems to have increased due to an increase in the number accurate recommended resources.

In the meantime, it has to be noticed that when the number of resources tremendously increases, the user confronts a sort of conflict in finding the most appropriate resources for his purpose. Taking this point into consideration, and based on the results shown in Figure 11, the suitable number of recommended resources to be presented to the user is 2 to 3. It is obvious that, a threshold bigger than this may cause confusion and resultantly dissatisfaction, while the one less than this may similarly be not satisfactory either.

**4. 3. User Experience Measure** User experience is considered as one of the most important issues in the field of human-computer interaction, which in fact examines the users' perceptions and sentiments toward the system or application they interact with. Developers usually count on their achievements on users'

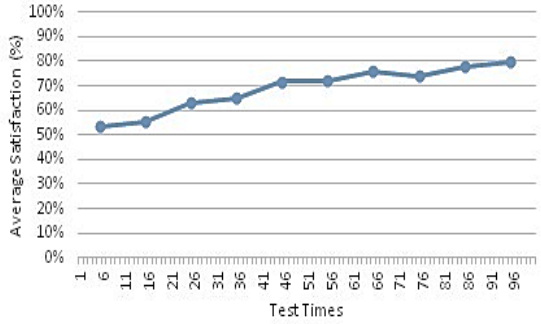


**Figure 10.** Average satisfaction change over test times

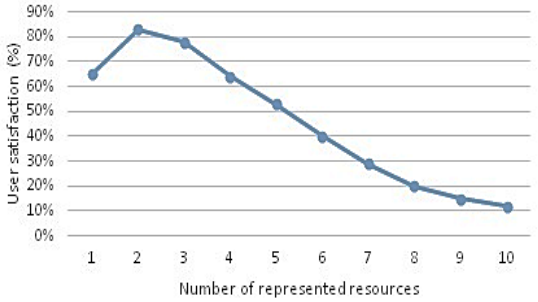


**Figure 11.** User satisfaction percentage by the number of represented resources for each constraint

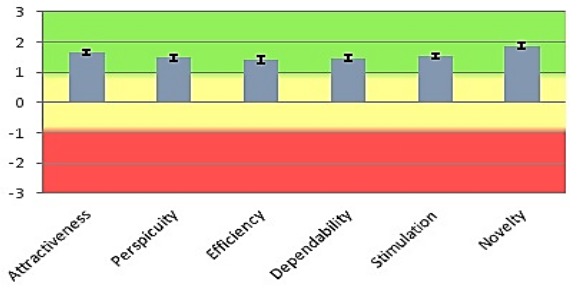


**Figure 12.** UEQ test result on six scales with confidence intervals

experiences, which result in improving the usability and friendliness of their system application [29]. Moreover, in complex projects where the number of requirements is high, for choosing customers' priorities during the product development, an intelligent algorithm such as Binary Artificial Algae Algorithm seems to be useful [30].

Considering the above points into account, in this paper, we decided to evaluate our recommender system from the user experience perspective, too. In this respect, a kind of user experience questionnaire (UEQ) designed by [30] was applied. The related test includes six scales of "attractiveness", "perspicuity", "efficiency", "dependability", "stimulation" and "novelty". It is to be noted that these scales, were measured according to some specific items which are considered in this questionnaire exactly for this purpose. The user's general feeling is calculated by the "attractiveness" scale, and ought to be affected by the other five scales. Thus, these scales are not independent of each other. The reason for using such a questionnaire is that it has been known as a standard tool for assessing various scales of user experience, and has functioned successfully in a variety of research works [31-33].

To evaluate the proposed system based on user experience perspective, in this experiment, 100 B.Sc. graduated engineers in computer science and computer-engineering fields voluntary accepted to have cooperation. After a short training, each participant worked with the system at least 15 minutes, and then filled in the user experience questionnaire (UEQ questionnaire). As illustrated in Figure 12, all UEQ's quality scales show a positive evaluation. Among them, "novelty" seems to be the highest. That is because no compositional adaptation with the specifications of the suggested approach has functioned for proposing appropriate resources for software development projects such as the one we proposed in this paper. Next to that "attraction" scale has turned to be highly acceptable, as the user interface of the system has been designed well and user-friendly. The other task-related scales (such as "assortment", "perspicuity", "efficiency", and "dependability"), are a little bit lower, because the system is in the pilot state and for the moment, the case library may not cover all resources for software development projects; a fact which in turn may cause weak outcome in facing complex or obscure requests.

More investigation on the mean value of UEQ reveals that "novelty" and "efficiency" are the two extremes of the spectrum. With respect to the standard deviation, it is noticeable that the highest value belongs to "perspicuity". That means there has been no consensus among participants on "perspicuity", while the most common belief has been on "attractiveness" among the participants. That is why the second highest average is allocated to "attractiveness". Table 1 shows the mean values, standard deviation and confidence interval of each UEQ's scale by 5% confidence intervals.

To evaluate the UEQ scales of our proposed system (compared to the other existing systems), the UEQ test dataset containing data from 9905 users in 246 studies concerning different products (like social networks, business software, web shops and etc.) has been considered. Figure 13 illustrates the status of each scale compared to the others.

As it is seen from the figure, "novelty" exists in the excellent area that means the proposed recommender system is in the range of 10% of the most successful systems. This outcome is achieved because of its semantic potential in finding and offering several appropriate solutions to the users. Respectively, the second place is allocated by "attractiveness" which reflects the good impression of users through working with this system. Next to "attraction", "stimulation" holds the third mean value. Results of test demonstrate that the proposed system has the ability to motivate users for further usage. The reason is the very accuracy and completeness of the previous suggestions already provided by the system. In this scale, the proposed system is placed in a position better than 75% of the other systems in the dataset. This comes true whereas just 10% of other products provide a better "stimulation" a fact that is quite remarkable for such a prototype system.

**TABLE 1.** UEQ's scales mean, standard deviation and confidence intervals (p=0.05)

| Scale | Mean | Std. Dev. | N | Confidence | Confidence interval | |
|---|---|---|---|---|---|---|
| Attractiveness | 1.643 | 0.425 | 100 | 0.083 | 1.560 | 1.727 |
| Perspicuity | 1.475 | 0.609 | 100 | 0.119 | 1.356 | 1.594 |
| Efficiency | 1.418 | 0.546 | 100 | 0.107 | 1.310 | 1.525 |
| Dependability | 1.463 | 0.515 | 100 | 0.101 | 1.361 | 1.564 |
| Stimulation | 1.535 | 0.437 | 100 | 0.086 | 1.449 | 1.621 |
| Novelty | 1.875 | 0.556 | 100 | 0.109 | 1.766 | 1.984 |

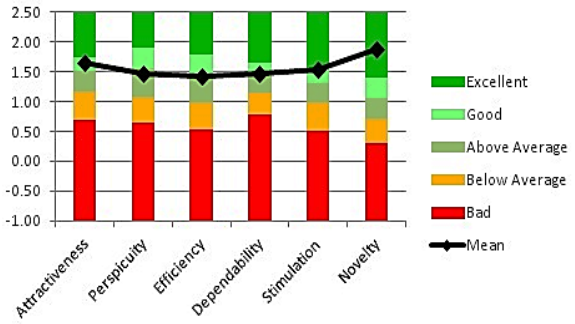


**Figure 13.** UEQ test result in comparison with the other systems/ products

On the other side, "Perspicuity" and "dependability" behave almost in a similar manner, and both are superior to 50% of the similar systems in the dataset. The very convenience in getting familiar with the system from the perspective of "perspicuity", and the safe control on interactions and complying the users' expectations from the "dependability" viewpoint, make the proposed system as successful as possible. However, by increasing the number of cases in the case library, and characterizing more significant semantic similarity (either between the cases or between the problem situation and the cases), more reliable and comprehensive solutions are expected to be provided, and "efficiency" of the system will thus increase. Moreover, by upgrading the hardware of the system, higher performance in general, and less retrieval time in particular, will be resulted.

## 5. CONCLUSION AND FUTURE PROSPECTS

In this paper, a compositional adaptation approach was presented for tailoring different types of learning resources in the area of software development with the purpose of recommending them to the users. The main point in this approach is the ability to produce a solution in the situations where the keywords in the user's request might not have necessarily been used in the technical descriptions of the stored cases. A domain-specific ontology, which comprises significant aspects of a technological area such as "infrastructure", "methods & tools", "applications & services", has been shown to be helpful in this regard. A salient point in our approach to compositional adaptation is that, the components in a case solution may hold some characteristics, which would justify the way the essential operations in the case solution are supported. This, as shown in the paper, gives us the chance to consider only those components that comply with the requirements of the user's request. In this way, the components gathered from the corresponding retrieved cases can be joined together to shape the final solution for the user's request. Although the approach emphasized in this paper has been proposed for recommending learning resources in the area of software development, it however can be equally applied to other technological areas as well through developing domain-specific ontology for that area. Therefore, the proposed approach can be regarded as a helpful means for recommending learning resources in any technological area in general. As a future research work, developing highly efficient domain-specific ontology as well as considering fuzzy logic for assessing semantic similarity, is suggested.

## 6. REFERENCES

1. Holstein, K., Yu, Z., Sewall, J., Popescu, O., McLaren, B.M. and Aleven, V., "Opening up an intelligent tutoring system development environment for extensible student modeling", in International Conference on Artificial Intelligence in Education, Springer. (2018), 169-183.
2. Dwivedi, A., Dwivedi, P., Bobek, S. and Zabukovšek, S.S., "Factors affecting students' engagement with online content in blended learning", ***Kybernetes***, (2019), https://doi.org/10.1108/K-10-2018-0559
3. Jaderyan, M. and Khotanlou, H., "Automatic hashtag recommendation in social networking and microblogging platforms using a knowledge-intensive content-based approach", ***International Journal of Engineering, Transactions B: Applications,*** Vol. 32, No. 8, (2019), 1101-1116, doi: 10.5829/ije.2019.32.08b.06.
4. Lu, J., Wu, D., Mao, M., Wang, W. and Zhang, G., "Recommender system application developments: A survey", ***Decision Support Systems***, Vol. 74, (2015), 12-32, https://doi.org/10.1016/j.dss.2015.03.008
5. Ai, F., Chen, Y., Guo, Y., Zhao, Y., Wang, Z., Fu, G. and Wang, G., "Concept-aware deep knowledge tracing and exercise recommendation in an online learning system", ***International Educational Data Mining Society***, (2019).
6. Ahn, J.-w., Tejwani, R., Sundararajan, S., Sipolins, A., O'Hara, S., Paul, A., Kokku, R., Kjallstrom, J., Dang, N.H. and Huang, Y., "Intelligent virtual reality tutoring system supporting open educational resource access", in International Conference on Intelligent Tutoring Systems, Springer. (2018), 280-286.
7. Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H. and Koper, R., Recommender systems in technology enhanced learning, in Recommender systems handbook. 2011, Springer.387-415.

8. Huang, M.J., Chiang, H.K., Wu, P.F. and Hsieh, Y.J., "A multi-strategy machine learning student modeling for intelligent tutoring systems: Based on blackboard approach", ***Library Hi Tech***, (2013), https://doi.org/10.1108/07378831311329059
9. Barzegar Nozari, R., Koohi, H. and Mahmodi, E., "A novel trust computation method based on user ratings to improve the recommendation", ***International Journal of Engineering***, Vol. 33, No. 3, (2020), 377-386, doi: 10.5829/IJE.2020.33.03C.02.
10. Lops, P., Jannach, D., Musto, C., Bogers, T. and Koolen, M., "Trends in content-based recommendation", ***User Modeling and User-Adapted Interaction***, Vol. 29, No. 2, (2019), 239-249, https://doi.org/10.1007/s11257-019-09231-w
11. Cheng, L.C. and Lin, M.-C., "A hybrid recommender system for the mining of consumer preferences from their reviews", ***Journal of Information Science***, Vol. 46, No. 5, (2020), 664-682, https://doi.org/10.1177/0165551519849510
12. Tarus, J.K., Niu, Z. and Yousif, A., "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining", ***Future Generation Computer Systems***, Vol. 72, (2017), 37-48, https://doi.org/10.1016/j.future.2017.02.049
13. Lin, F., Graf, S. and McGreal, R., "Adaptive and intelligent web-based educational systems", (2009).
14. Cuong, N.D.H., Arch-Int, N. and Arch-Int, S., "Fuse: A fuzzy-semantic framework for personalizing learning recommendations", ***International Journal of Information Technology & Decision Making***, Vol. 17, No. 04, (2018), 1173-1201, https://doi.org/10.1142/S0219622018500220
15. Azcona, D., Hsiao, I.-H. and Smeaton, A.F., "Detecting students-at-risk in computer programming classes with learning analytics from students' digital footprints", ***User Modeling and User-Adapted Interaction***, Vol. 29, No. 4, (2019), 759-788, doi: 10.1007/s11257-019-09234-7.
16. Kolodner, J.L., Cox, M.T. and González-Calero, P.A., "Case-based reasoning-inspired approaches to education", ***The Knowledge Engineering Review***, Vol. 20, No. 3, (2005), 299-303, doi. https://doi.org/10.1017/S0269888906000634
17. Dixit, P., Nagar, H. and Dixit, S., "Student performance prediction using case based reasoning knowledge base system (cbr-kbs) based data mining", ***International Journal of Information and Education Technology***, Vol. 12, No. 1, (2022), doi: 10.18178/ijiet.2022.12.1.1583.
18. Reyhani, N., Badie, K. and Kharrat, M., "A new approach to compositional adaptation based on optimizing the global distance function and its application in an intelligent tutoring system", in Proceedings Fifth IEEE Workshop on Mobile Computing Systems and Applications, IEEE. (2003), 285-290.
19. Arshadi, N. and Badie, K., "A compositional approach to solution adaptation in case-based reasoning and its application to tutoring library", in Proceedings of 8th German Workshop on Case-Based Reasoning. Lammerbuckel, Citeseer. Vol. 11, (2000).
20. Bousbahi, F. and Chorfi, H., "Mooc-rec: A case based recommender system for moocs", ***Procedia-Social and Behavioral Sciences***, Vol. 195, (2015), 1813-1822, doi. https://doi.org/10.1016/j.sbspro.2015.06.395
21. Garrido, A., Morales, L. and Serina, I., "On the use of case-based planning for e-learning personalization", ***Expert Systems with Applications***, Vol. 60, (2016), 1-15, doi. https://doi.org/10.1016/j.eswa.2016.04.030
22. Guessoum, D., Miraoui, M. and Tadj, C., "Contextual case-based reasoning applied to a mobile device", ***International Journal of Pervasive Computing and Communications***, (2017), doi. https://doi.org/10.1108/IJPCC-11-2016-0056
23. Nasiri, S., Zenkert, J. and Fathi, M., "Improving cbr adaptation for recommendation of associated references in a knowledge-based learning assistant system", ***Neurocomputing***, Vol. 250, (2017), 5-17, https://doi.org/10.1016/j.neucom.2016.10.078
24. Salam, A. and Fathurrahmad, F., "Student final project recommendation system model using case-based reasoning (cbr) method", ***Jurnal Mantik***, Vol. 5, No. 3, (2021), 1535-1542, https://iocscience.org/ejournal/index.php/mantik/article/view/1641
25. Dhiman, H., Wächter, C., Fellmann, M. and Röcker, C., "Intelligent assistants", ***Business & Information Systems Engineering***, (2022), 1-21, doi: 10.1007/s12599-022-00743-1.
26. Wilke, W. and Bergmann, R., "Techniques and knowledge used for adaptation during case-based problem solving", in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer. (1998), 497-506.
27. Badie, K. and Mahmoudi, M.T., "Compositional adaptation in case-based reasoning based on the semantic relations between the components in the cases", in 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), IEEE. (2017), 449-454.
28. Laugwitz, B., Held, T. and Schrepp, M., "Construction and evaluation of a user experience questionnaire", in Symposium of the Austrian HCI and usability engineering group, Springer. (2008), 63-76.
29. Laumer, S., Maier, C. and Weitzel, T., "Information quality, user satisfaction, and the manifestation of workarounds: A qualitative and quantitative study of enterprise content management system users", ***European Journal of Information Systems***, Vol. 26, No. 4, (2017), 333-360, https://doi.org/10.1057/s41303-016-0029-7
30. Pirozmand, P., Ebrahimnejad, A., Alrezaamiri, H. and Motameni, H., "A novel approach for the next software release using a binary artificial algae algorithm", ***Journal of Intelligent & Fuzzy Systems***, Vol. 40, No. 3, (2021), 5027-5041, doi: 10.3233/JIFS-201759.
31. Hinderks, A., Meiners, A.-L., Domínguez Mayo, F.J. and Thomaschewski, J., "Interpreting the results from the user experience questionnaire (UEQ) using importance-performance analysis (IPA)", in WEBIST 2019: 15th International Conference on Web Information Systems and Technologies (2019), pp. 388-395., ScitePress Digital Library. (2019).
32. Rauschenberger, M., Schrepp, M., Pérez Cota, M., Olschner, S. and Thomaschewski, J., "Efficient measurement of the user experience of interactive products. How to use the user experience questionnaire (UEQ). Example: Spanish language version", (2013), doi: 10.9781/ijimai.2013.215.
33. Schrepp, M., Hinderks, A., Thomaschewski, J. and Marcus, A., "Design, user experience, and usability. Theories, methods, and tools for designing the user experience", ***Lecture Notes in Computer Science***, Vol. 8517, (2014), 383-392, doi. https://doi.org/10.1007/978-3-319-07668-3_37

## Persian Abstract

**چکیده**

در این مقاله در خصوص کاربرد رویکرد سازگارسازی ترکیبی جهت پیشنهاد منابع یادگیری به کاربران درحوزه توسعه نرم افزار بحث می شود. بدین منظور از یک هستان نگار پیشنهادی وابسته به دامنه جهت جستجوی کلماتی که در توصیف فنی مورد های ذخیره شده بکار رفته اند استفاده می شود. آنچه در بازنمایی موردهای رویکرد پیشنهادی حائز اهمیت است، ارائه دقیق مشخصات منابع یادگیری در بخش راه حل های موردهاست. بدین ترتیب تنها راه حل هایی که با درخواست کاربر متناسب باشد، بازگردانده خواهد شد. دراین مقاله کارایی رویکرد پیشنهادی جهت پیشنهاد منابع یادگیری همراه با وضعیت تجربه کاربر در تعامل با سیستم پیسنهاد دهنده نیز مورد ارزیابی قرار گرفته است. نتایج حاصل نشان می دهد که منابه یادگیری پیشنهادی توسط این رویکرد برای کاربران با درجه رضایت بالایی مفید فایده بوده است. لازم به ذکر است که هرچند رویکرد پیشنهادی بکار گرفته شده جهت پیشنهاد منابع یادگیری در حوزه توسعه نرم افزاراستفاده شده است ولیکن قابل استفاده در حوزه های فنی دیگر به شرط طراحی هستان نگار دامنه آن موضوع نیز می باشد.