UDC 004.89

O. Gladkova, A. Parkhomenko, Ya. Zalyubovskiy
Zaporizhzhia National Technical University

DEVELOPMENT AND APPLICATION OF THE RECOMMENDATION METHODS FOR EMBEDDED SYSTEMS COMPUTER AIDED DESIGN

The analysis of mathematical methods for elements similarity defining for realization in recommendation algorithms is presented in the paper. The usage of the recommendation methods for hardware-software platforms selecting during embedded systems computer aided design is proposed. The results of the practical application of the developed recommendation system during realization of embedded system for moving objects control are given.

Key words: embedded system, hardware-software platform, recommendation method, knowledge-based method, knowledge database, distance method.

РОЗРОБЛЕННЯ ТА ЗАСТОСУВАННЯ РЕКОМЕНДАЦІЙНИХ МЕТОДІВ ПІД ЧАС АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ

Проаналізовано математичні методи визначення подібності елементів для реалізації у рекомендаційних алгоритмах. Запропоновано використання рекомендаційних методів для вибору апаратно-програмних платформ під час автоматизованого проектування вбудованих систем. Наведено результати практичного застосування розробленої рекомендаційної системи під час створення вбудованої системи управління рухомими об'єктами.

Ключові слова: вбудована система, апаратно-програмна платформа, рекомендаційний метод, фільтрація на основі знань, база знань, метод відстаней.

Introduction

The application of prototyping technologies for complex objects and systems design can reduce the development time by decreasing the risk of final product reworking due to the shortcomings identified at the final development stages [1]. Therefore, the usage of ready hardware-software platforms for the rapid creation and investigation of the prototype is expedient in the field of embedded systems (ES) design. The process of hardware-software platforms selecting is rather complicated. It takes a lot of time because of the need to search and analyze a large number of options offered by manufacturing companies. Therefore, the task of automating the process of hardware-software platforms selecting on the basis of the requirements to the designed system through the development and implementation of the recommendation system is relevant. This will reduce the transition time between the system and the functional-logical levels during ES design, as well as increase the level of automation of design work.

Analysis of options for implementing the recommendation method

The investigation has shown that the recommendation system (RS) includes the following phases (Fig. 1): information collection phase, processing/learning phase, prediction/recommendation phase [2]. These phases are built around the principles of information retrievals and used recommendation methods: collaborative filtering, content-based filtering, knowledge-based filtering, hybrid filtering [2–4]. The analysis of the peculiarities of the implementation of these methods showed that, the most expedient is the use of knowledge-based method for recommendation system to solving the problem of the choice of hardware-software platforms. Such a system is not built around marks or ratings, and allows to take into account the current requirements of the developer for a specific project product. RS solves the specific tasks of each individual developer and usually does not preserve the behavior of developers and their marks. It works with knowledge bases, which collects data on possible options and, based on recommendation algorithms, provides quick solutions to search for a hardware-software platform that meets the requirements of the ES project. A diagram of the process of forming recommendations using the knowledge-based method can be presented as shown in Fig. 2. At the same time, for the implementation of the recommendation algorithm, various mathematical methods [5, 6] are used, which allow to calculate similarity for the recommendation elements.

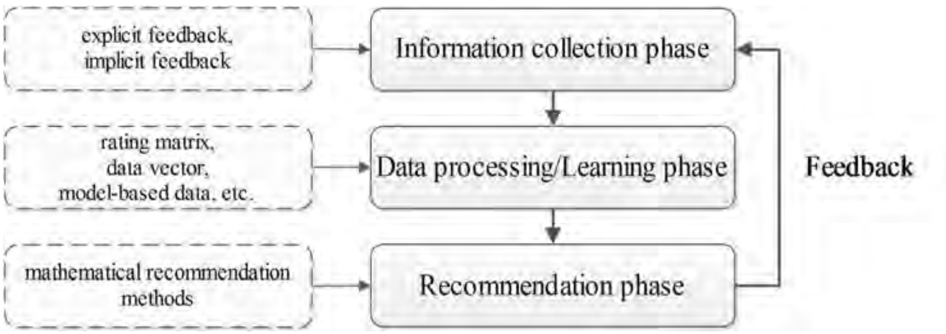

Fig. 1. Recommendation phases

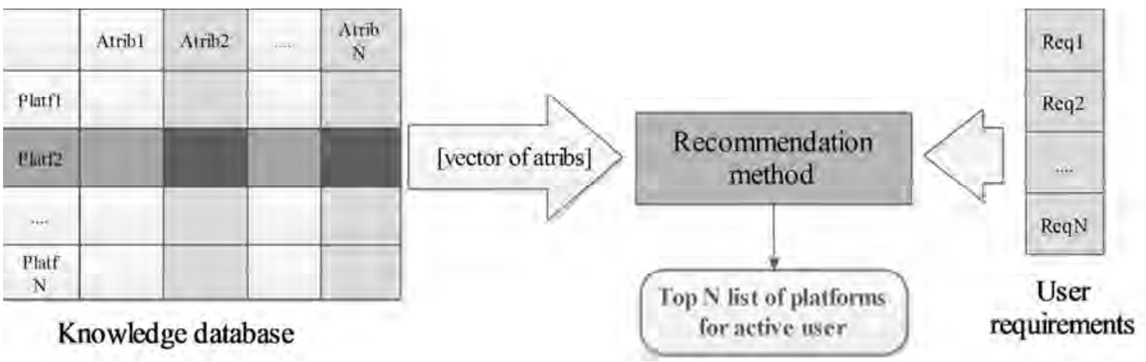

Fig. 2. Process of forming the recommendation

Based on the analysis of existing methods of similarity measurement, an easy-to-implement method of distances was chosen. Its essence is that with the help of the chosen metric (Euclidean distance, cosine of similarity, etc.), the distance between the two objects is calculated.

Euclidean distance is the most common measure of similarity between two objects [3]:

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2},$$

where n – number of attributes; x_k, y_k – the *k-th* attribute of the object vector.

The cosine of similarity is the most commonly used measure in collaborative filtering [7] to find similarity between two objects represented as the vectors [5]:

$$\cos(x,y) = \frac{\sum_{k=1}^{n} x_k y_k}{\sqrt{\sum_{k=1}^{n} x_k^2} + \sqrt{\sum_{k=1}^{n} y_k^2}},$$

where n – number of attributes; x_k, y_k – the *k-th* attribute of the object vector.

The result is sorted in ascending order and is issued in the form of a recommendation: the closest one – the most acceptable.

In [4], for a knowledge-based filtering method, a measure of similarity between objects and user requirements is considered:

$$similarity(X,Y) = \frac{\sum_{y_n \in Y} w_{y_n} \cdot sim(x_n, y_n)}{\sum_{y_n \in Y} w_{y_n}},$$

where X – the set of object attribute; Y – the set of requirements; y_n – requirement; w_{yn} – n-th requirement weight; $sim(x_n, y_n)$ – measure of the object x_n to the requirement y_n similarity.

There are several options for finding a measure of similarity to $sim(x_n, y_n)$ numerical attributes [4]:

· "More is better" (for example, the more analog and digital inputs on the platform, the better),

$$sim(x, y) = \frac{\varphi_y(x) - min(y)}{max(y) - min(y)}$$

· "Less is better" (the classic example is the price, the lower the better),

$$sim(x, y) = \frac{max(y) - \varphi_y(x)}{max(y) - min(y)}$$

· "Closer is better" (a classic approach in the form of Euclidean space, when the user has no difference, more or less, a kind of modification of strict rules of fuzzy form),

$$sim(x, y) = 1 - \frac{\left|\varphi_y(x) - y\right|}{max(y) - min(y)}$$

where, $sim(x, y)$ – measure of the similarity of object x to the requirement y; $min(y)$ and $max(y)$ – minimum and maximum attribute values; y – the exact value of the user requirement; $\varphi_y(x)$ – the corresponding value of the object attributes.

On the basis of the analysis of the effectiveness of existing methods for the similarity determination [8–10], we can conclude that the results of their work vary considerably and depend on the features of the solved task. According to studies [11], the accuracy of RS prediction does not depend on the chosen metric of similarity. However, in the case when of insufficient information at the RS input or its contradictions, unforeseen problems may arise during the above similarity detection methods using. Therefore, they need to be modified by processing semantically exceptional cases and/or by using elements of other methods (for example, the method of multicriteria analysis [12]).

Thus, for the solution of the task of the determination of the elements similarity for the implementation of the recommendation method based on knowledge, it is expedient to use the distance method and a combination of options for finding a measure of similarity for numerical attributes. However, in order to take into account the features of the task being solved, these measures of similarity have been modified.

Method of recommendations forming for hardware-software platforms

The developed method for the hardware-software platforms recommendations forming [13] uses a knowledge-based approach, which includes strict restriction methods and selection of similar objects, in order to create a more flexible recommendation system. The idea of both methods is as follows: the developer formulates his requirements to the object, the system tries to find the desired object, based on these requirements. In the first case, only those objects that exactly meet all the requirements of the developer are recommended. In the second case, objects with characteristics similar to requirements can be recommended (using mathematical methods of similarity search). Therefore, we have decided about application of these methods in a shared manner by separating the criteria into two groups. The first group is strict criteria and the second group is criteria for which the search of the similar objects should be performed, thus it will allow more detailed consideration of each criteria.

Consideration of only strict criteria does not make sense, because they obey the rule "IF – THEN". For example, if the user wants to use a simple platform for the novice level, we should not offer him platforms of professional level.

Methods of the group based on close selection, have the closest contact with user requirements. In this case, it is possible to recommend objects that partially meet the requirements.

The input data for the method of forming recommendations implementation is the vector of the developers' requirements. The first stage of the method is the formation of a plurality of hardware-software platforms in accordance with a given level of the developers' knowledge, derived from the vector of requirements. The system performs filtering of hardware-software platforms (farther platforms) according to the level of the developers' knowledge (strict criteria – see Table 1) and it forms a list of possible platforms.

Then, at the next stage, the similarity calculating is performed for the flexible criteria, namely, the measure of similarity is determined between the inputted vector of the developer's requirements and the attributes of the platform by the measure of similarity between the objects and the requirements of the user.

For a plurality of platforms formed at the first stage, a measure of similarity is calculated for "flexible" criteria (Table 1):

· if these criteria are not defined, the plurality of platforms formed on stage 1 comes back;

· if the criteria "number of analog inputs" or "number of digital inputs" are entered, then the platforms are determined from the plurality formed at stage 1 by the measure "more or better";

· taking into account that the developer enters the necessary number of the inputs, the recommendation of a less number is not correct, thus taking into account the specifics of criteria, the formula was modified by condition:

$$sim(x, y) \geq y,$$

· if the criteria "power supply" is entered, the platforms are determined from the plurality defined during the analysis of digital inputs, by the measure "closer is better";

· if the criteria "price" is entered, the platform is determined from the plurality defined during power supply analysis by the "less is better" measure.

It has been determined that the best option would be to assign a weight coefficient w = 0.2 for the number of analogue, digital outputs and power supply, and weight w = 0.7 for the price (Fig. 3). Next, the value of the similarity of the hardware-software platform is calculated, taking into account the weight of each flexible criteria. Platforms are sorted by weight and returned to the main flow of the algorithm.

Table 1

Metrics used by criteria

Criteria title	Criteria type	Measure	Weight coefficient
User level	Strict	Strict restrictions	1
Number of analog inputs	Flexible	Closer is better, but not less	0,2
Number of digital inputs	Flexible	Closer is better, but not less	0,2
Power supply	Flexible	Closer isbetter	0,2
Family of processors	Strict	Strict restrictions	1
A programming language	Strict	Strict restrictions	1
Price	Flexible	Less is better	0,7
Form factor	Strict	Strict restrictions	1

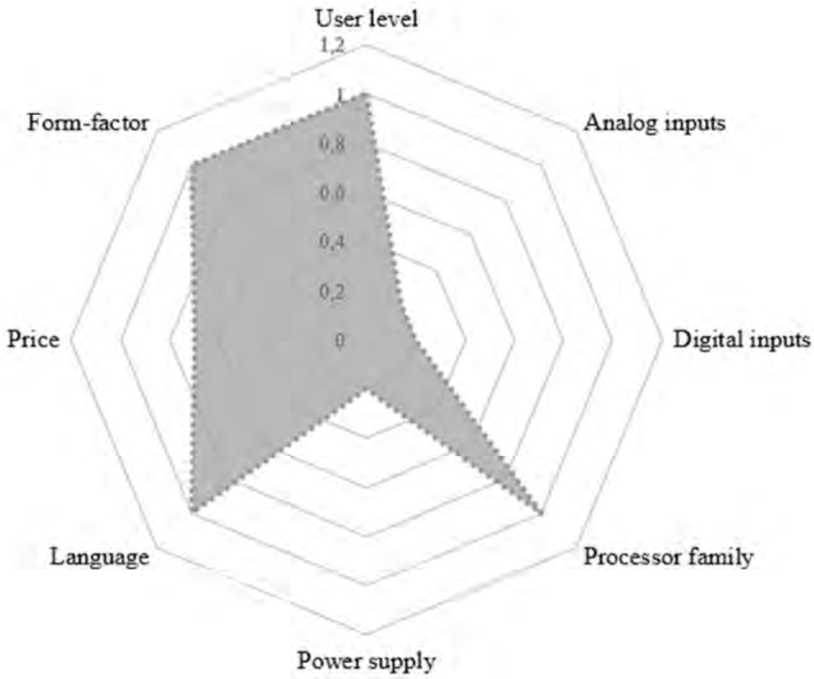

Fig. 3. Radar chart of correlation of criteria weighting coefficients

After the flexible criteria formation, the filtering is carried out according to strict criteria. The platforms from the modified at the previous stage plurality are sorted according to strict criteria (if they exist, otherwise the algorithm passes the stage of strict sorting).

Next, it is performed the formation of the resulting list of coefficients of similarity of the platforms to the requirements of the developer. The list of received recommendation elements is formed according increasing the indicator of the similarity coefficient.

After that, the number of recommended platforms is checked. If the plurality of platforms formed at the previous stage has one or more platforms, the four closest recommendations are displayed (in general, the best on demand and the three closest to it). Thus, at the output we get up to four hardware-software platforms with a minimum distance:

$$result = \min d,$$

where d is the value of the similarity coefficient.

In the case when no recommendation is made, we proceed to the formation of the result for each separate strict criteria. At this stage, the method of multicriteria analysis of the best alternative for each separate strict criteria is used to solve the contradictory of entered requirements. If the plurality of platforms formed at the previous stage is zero, then one of the following possible options is provided:

· the stages are performed repeatedly of the plurality formation by the level of developer knowledge and similarity by flexible criteria; after that a separate and joint pairwise strict sorting is performed according to the following criteria: family of processors, programming language, form factor of the platform; the best is chosen from these sorts (the largest number of matches according to these strict criteria) and one platform is displayed for each;

· otherwise, if the user's requirements are rather contradictory, a message is displayed that the platform is not found, it is suggested to return and edit entered by the user requirements.

Based on the developed recommendation method, the recommendation system (RS) was developed and implemented in the practice of ES engineering design [14]. This RS was integrated with a remote laboratory for embedded systems design. It was practically applied in the process of computer aided design of the mobile objects control embedded system [15]. The results of the work of the RS allowed us to select the hardware-software platforms for prototyping the developed ES (Figs. 4, 5). The verification of the of the prototype's functions showed that the set of requirements for the designed system was fully implemented, and therefore the provided recommendations were reasonable.

Fig. 4. The results of the formation of recommendations for all platforms

Fig. 5. The results of the formation of recommendations for minicomputers

Conclusions

The task of automation of the selecting of hardware-software platforms on the basis of requirements to the designed ES is solved by the development and implementation of the recommendation system.

The method of recommendations forming based on the knowledge has been developed. On the basis of this method, the toolkit for providing recommendations regarding hardware-software platforms to the designer during the process of ES computer aided design was created.

The list of the requirements for the searched hardware-software platform was formed, with the usage of the recommendation system, the hardware-software platforms were selected in accordance with the requirements for the designed system and the possible variants of the project development were determined. This allowed to increase the level of design works automation and to reduce the time of the corresponding platform searching for the creation and study of the prototype of designed ES.

1. Rapid Prototyping and Engineering Applications: A Toolbox for Prototype Development, FRANK W. LIOU CRC Press Taylor & Francis Group Boca Raton, FL 33487-2742, 2007, 592 p. 2. Isinkaye F. O., Folajimi Y. O., Ojokoh B. A. Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal Vol. 16, Issue 3, November 2015, Pages 261-273 3. Recomender Systems: Handbook / Ricci F., Rokach L., Shapira B., Kantor P.B. Springer Science+Business Media, LLC. 2011. 845 p. 4. Гомзин А. Г., Коршунов А. В. Системы рекомендаций: обзор современных подходов. Труды Института системного программирования РАН. 2012. Т. 22, С. 401–417. 5. Tarus K. J., Niu1 Z., Mustafa G. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. Artificial Intelligence Review. 2017. P. 1–28. 6. Белоцкий Е. А., Суети А. В. Построение рекомендательной системы по подбору высших учебных заведений для абитуриентов. Вестник СПбГУ. Сер.10. Прикладная математика. Информатика. 2016. Вып. 1. С. 66–77. 7. O'Mahony M. P., Hurley N. J., Silvestre G. C. M. An Evaluation of the Performance of Collaborative Filtering. Artificial Intelligence & Cognitive Science: proceedings of 14th Irish Conference, 19 December, 2003. P. 171–175. 8. Huang A. Similarity Measures for Text Document Clustering. NZCSRSC: procedings of 6th New Zealand Computer Science Research Student Conference, April, 2008. New Zeland. 2008. P. 49–56. 9. Rafi M., Shaikh M.Sh. An improved semantic similarity measure for document clustering based on topic maps. Cornell University Library. URL: https://arxiv.org/ftp/arxiv/papers/1303/1303.4087.pdf. (Last accessed: 15.03.2018). 10. Антонова А. Ю., Клышинский Э. С. Об использовании мер сходства при анализе документации. Электронные библиотеки: перспективные методы и технологии, электронные коллекции: труды 13 Всероссийской научной конференции, 2011. Воронеж, Россия, 2011. С. 134–138. 11. Lathia N., Hailes S., Capra L. The effect of correlation coefficients on communities of recommenders. SAC'08: proceedings of 23rd Annual ACM Symposium on Applied computing. Fortaleza, Ceara (Brazil), 16–20 March, 2008. P. 2000–2005. 12. Кузнецов В. І., Євтушенко Г. Л. Системне моделювання складних об'єктів на базі методів багатокритеріального аналізу. Системные технологии моделирования сложных систем: монография / под общей ред. проф. А. И. Михалёва. Днепр: НМетАУ-ИВК "Системные технологии", 2016. С. 349–374. 13. Гладкова О. М., Пархоменко А. В. Дослідження та практична реалізація рекомендаційної системи для вибору апаратно-програмних платформ при автоматизованому проектуванні вбудованих систем. Наукові праці ДонНТУ. Серія "Інформатика, кібернетика та обчислювальна техніка". 2017. № 2(25). С. 22–31. 14. Subbotin S., Gladkova O., Parkhomenko A. Knowledge-based recommendation system for embedded systems platform-oriented design. Computer science and information technologies: proceedings of XIII International Scientific and Technical Conference, 11–14 September, 2018. Lviv. P. 368–373. 15. Дослідження та розробка автоматизованої системи віддаленого керування групою рухомих об'єктів / А. В. Пархоменко, О. М. Гладкова, О. П. Кравченко, Д. П. Кравченко. Вісник СНУ ім. В. Даля. 2017. № 8(238). С. 67–74.