An Object-Oriented Model of Case-Based Reasoning System Using Situations Tree

Ruslan K. Akhmadulin
Department of Cybernetic Systems
Tyumen Industrial University
Tyumen city, Russia
ark@tsogu.ru

Igor N. Gluhih, Ivan Y. Karyakin
Information System Department
Tyumen State University
Tyumen city, Russia
igluhih@utmn.ru, ivan_nyc@mail.ru

***Abstract*— The article is dedicated to development of an object-oriented model of an intelligence system based on case-based reasoning (CBR). An upgrade of the decision tree has been introduced to represent the situations hierarchy and related decisions in the knowledge base. The construction of the situations tree has been described. The problem of representing the situations tree in terms of classes and objects has been solved and the method of storage of the situations tree in the database has been defined. As a result, on the basis of the object-oriented paradigm the proposed approach allows creating domain-independent CBR systems that are shells for filling with case-oriented knowledge to solve specific problems.**

Index Terms - case-based reasoning, object-oriented model, decision tree.

I. Introduction

In modern intelligence systems of decision support – decision support system (DSS) – one of the most popular concepts of reasoning is case-based reasoning (CBR). The basic idea of CBR is to use knowledge about similar problems and their solutions, already familiar from previous experiences, to search for solutions to new problems [1-2]. The universality and efficiency of this approach allows using it in the development of activity and decision-making support systems in various subject areas: medicine [3, 4], construction and industry [5-7], software engineering [8], financial sector [9, 10], development of recommender systems [11], corporate knowledge bases [12], complex objects management automation [13-15], etc.

Cases in CBR represent recorded in the knowledge base descriptions of known situations and the decisions which were made in these situations and yielded positive results or were deemed rational by the experts forming the knowledge base [2]. To formalize the representation a couple P = <situation (Sit), resolution (R)>, is introduced which can be supplemented with other auxiliary elements, for example, the results of the decision [2]. The very knowledge base (KB) of such a system is called cases base.

In reasoning a current situation (problem) is identified and case P* is searched for which meets some criteria of closeness to this situation. Next, the solution from the P* couple is applied. This may be accompanied by adaptation of the solutions [1, 2, 16], which takes into account possible differences between the current situation and that, contained in KB. If in the course of solution adaptation and application new results for a new problem have been obtained, a new case is included into KB.

Among the most important research tasks in the development of CBR systems there are the tasks of searching for and selection of cases for KB, for which two main approaches are used [1, 2]. The first one involves search and selection of cases based on calculation of situation attributes proximity in space using some metrics; the second one is sequential analysis of the attributes using a decision tree. And although the second approach is less universal and flexible, more time consuming to edit or augment the knowledge base, it has certain advantages. Thus, the occurrence of the decision tree already specifies the arrangement of the cases in KB and the very algorithm of output.

In this work, we introduce an upgrade of the decision tree to represent the situations hierarchy and related decisions in KB. On this basis an object-oriented model of CBR-systems, invariant to the subject area, has been developed; it is applicable to creation of reasoning problem-oriented systems in various subject areas.

II. Preliminaries

For an object-oriented model of a CBR system knowledge representation and reasoning are used on the basis of the situations tree (ST) [17]. When building the ST a ratio ρ = «supertype-subtype» (is-a) on a variety of situations is introduced and a formal description of a situation through a variety of its attributes is specified:

$$Sit = \{A_i \mid i \in M^k\}, \tag{1}$$

where Mk is a set of indices of attributes, the capacity of which depends on the level k in the situations tree.

A_i is an i-th situation attribute: $A_i = \langle ID_i, D_i \rangle$, ID_i is an attribute identifying code (name); D_i is a set of attribute values.

In general case it is assumed that the attribute value can be of any type, i.e. any known quantitative or qualitative scales are used.

When building a ST, a number of conditions are adopted which regulate the creation of cases base and allow arranging reasoning [18]:

- the situations of each lower level are a refinement (extension of the definition) of the situations of the upper level, i.e. $M_k \subseteq M_{k+1}$. In other words, a situation of the lower level inherits the set of situation attributes of the upper level.
- every situation in the ST is a subtype of only one situation-prototype; it can serve as a prototype for an unlimited number of situations of the lower level;
- the situations of the same level are described by the same set of attributes. The difference between them is provided by different values of these attributes;
- when comparing the same attributes of situations X and Y the concept of distance in general case is not introduced, but a conclusion about the equality-inequality of their values can be made due to belonging to the same subset (or subrange) of values $d \in D^k_i$: $x \in d,\ y \in d \Rightarrow x = y$, $x \in d,\ y \notin d \Rightarrow x \neq y$.
- each situation of the ST has its decision. The decision of situation-prototype is applicable for all subtypes; but it differs from them in less detail and a higher level of abstraction. Thus, each point of the ST corresponds not only to its own situation, but in general, to couple P = <Sit, R>.

Graphically the ST is built up as a graph of dependence ρ on the set of possible situations (Figure 1):

$$G_{Sit}(\{S^k_j\}, V), \tag{2}$$

where $\{S^k_i\} \Leftrightarrow \{Sit^k_i\}$ is a set of points one-to-one corresponding to a set of situations; j = 1, 2, ..., N^k is the index of the number of a situation in the k-th level, N^k is the number of selected situations on the k-th level.

V is a set of arcs with $v(S^k_j, S^{k+1}_j) \in V$, if dependence $\rho(Sit^k_j, Sit^{k+1}_j)$ is realized, and there is no arc from the lower level to the upper one, as well as between the elements of the same level.

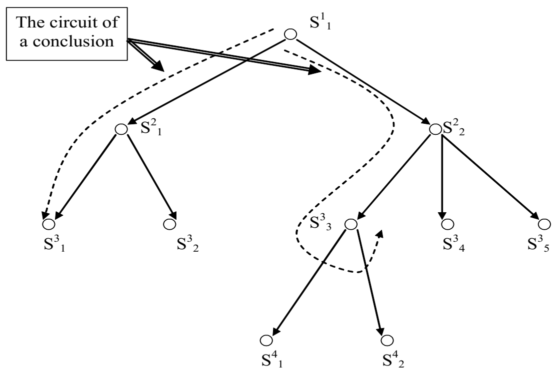

Figure 1. Situations tree

During the operation of the CBR-system the movement is from the root of the tree downwards. Since each level contains its set of attributes, this determines what questions should be asked to identify the current situation Sit_{cur} at the given k-th level. The closest $Sit^{k*} \in \{Sit^k_i\}$ is selected according to the determined attribute values Sit_{cur}, which determines the further path of movement along the tree. Reasoning ends with the end point (example in Figure 1 – point S31). Since a case with its decision corresponds to this point, it is applied as desired.

The arrangement of a ST allows reasoning in conditions of lack of information about the values of certain attributes taking into account the above conditions, when each point corresponds to its decision. The example in Figure 1 is reasoning S^3_3, which ends the algorithm of searching for a decision with unknown values of attributes present at the 4th level of the description of the situations, which does not allow recognizing Sit_{cur} as S^4_1 or S^4_2.

III. Object-Oriented Model

For further development of an object-oriented model of CBR-system and software implementation the following problems have been solved in the article:

- representation of a situations tree in terms of classes and objects;
- definition of the method of storage of situations trees in the database.

In solution of these tasks the results of the works devoted to representation of the systems based on knowledge in the form of object models [19-22] were used

For software implementation of the situations base in terms of object-oriented approach a common class ("layout") was created which allows describing any case, represented in the system, as a class instance. In this class not only its attributes are stored, but the reference to a "parent" case that allows creating a situations tree.

To store the case, class Case is created. Such attributes as the Name of the case, the Reference to the parent case, the Reference to the solution, and the List of attributes with values are implemented in this class. Among the methods of the class a method of Equivalency checking is implemented, which allows determining whether the situation corresponds to that under consideration.

The list of attributes is an arbitrary number of couples Attribute-Value. Each such couple is represented as a class instance of the Attribute case. This class implements not only the fields and properties that allow storing an Attribute and its Value, but also the method that correctly compares the current attribute value with the compared value for compliance.

Another class used to describe the situations tree is the Solution. The Instances of this class store possible variants of the solutions to the situations that are referenced by the instances of Case class. The main characteristics of a class instance are the Name, the result Type, and the result Text.

The implementation of a situations tree using object approach has simplified the process of searching for a similar case and an appropriate solution. Now for this purpose the methods implemented in class Case are used, including Equivalency checking. As a result, search for a case in the tree corresponding to the desired situation is reduced to alternate method call starting from the tree root and further on.

To store cases and solutions a knowledge base is implemented, which contains all the required information about the situations tree in terms of object-oriented approach. The physical basis for the situational knowledge base is a database that stores cases with the list of values of their attributes, as well as the solutions reached in these situations.

The database contains four main entities, shown in figure 2.

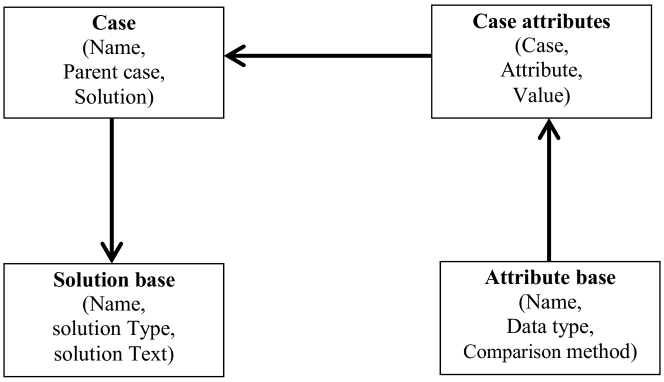

Figure 2. Database layout

First, it is Cases. Each case contains both its own name, and a reference to the parent case. This approach allows representing all the cases in a tree form where each case of the next level adds new attributes to its "parent" cases. Case-root of the tree has no reference to its parent.

Each case can have a reference to a Solution. All solutions are stored in the entity of Solution base. Each solution is represented in one of possible forms, for example in the form of text or program code.

To store case attributes and their values, another entity - Case Attributes - is designed. Each record of this entity stores a Reference to the case, a Reference to an attribute, Field to store the value of an arbitrary data type and an additional field to store the Second value in case the attribute value is specified by the range.

The very attributes are stored in a centralized entity of Attributes Base. It contains a description of each attribute: The Type of its values, and the Equivalency checking method: strict equality, larger or less than a predetermined value, entering the interval, etc.

IV. Results And Conclusions

Based on the object-oriented paradigm the proposed approach allows creating domain-independent CBR systems, which are shells of these systems, filled with knowledge about the cases for solving specific problems.

One of the important advantages of this approach is the possibility of developing basic case libraries for a wide class of problems in a certain subject area. The mechanisms of inheritance and overriding, typical for object-oriented programming, with the use of these libraries allows developers without knowledge implementing further configuration of their own system which takes into account specific features of certain problems.

Thus, there is a new class of systems-shells, based on knowledge, where knowledge bases are not empty, but contain knowledge generally valid for this or that subject area. They can be used by means of refining and addition when creating their advising or recommender systems, in which the situations are redefined due to new attributes, take into account local characteristics and contexts for decision-making in specific circumstances.

V. References

[1] J.L. Kolodner "An introduction to Case-Based Reasoning. Artificial Intelligence Review", num. 6, 1992, pp.3-34.

[2] I.D.Watson, F.Marir "Case-based reasoning: A review. The Knowledge Engineering Review", vol. 9, No. 4, 1994, pp. 355-381

[3] H.J. Gómez-Vallejo, B. Uriel-Latorre, M. Sande-Meijide, B Villamarín-Bello., R. Pavón, F. Fdez-Riverola, D. Glez-Peña "A case-based reasoning system for aiding detection and classification of nosocomial infections. Decision Support Systems", vol. 84, April 2016, pp. 104-116.

[4] P.R. Varshavskiy "Mekhanizmy pravdopodobnykh rassuzhdeniy na osnove pretsedentov (nakoplennogo opyta) dlya sistem ekspertnoy diagnostiki", 11-ya natsionalnaya konferentsiya KII-2008: Trudy konferentsii. V 3-kh t., T. 2, M: Lenand, 2008, s. 321 — 329.

[5] N. Naderpajouh, A. Afshar A case-based reasoning approach to application of value engineering methodology in the construction industry. Construction Management and Economics, № 26(4), 2008, pp.363–372.

[6] M.M. Mohamed, M.A. Rasha "A case-based reasoning approach for estimating the costs of pump station projects", Journal of Advanced Research, vol. 2, 2011, pp. 289–295.

[7] Ji Xia, Gang Chen, Peng Tan, and Cheng Zhang. "An online case-based reasoning system for coal blends combustion optimization of thermal power plant", International Journal of Electrical Power & Energy Systems, vol. 62, 2014, pp. 299-311.

[8] Rodrigo G.C. Rocha, Ryan R. Azevedo, Ygor Cesar Sousa, Eduardo de A. Tavares, and Silvio Meira. A case-based reasoning system to support the global software development. 18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES2014. Procedia Computer Science, №35, 2014, pp. 194 – 202.

[9] M.T. Elhadi Bankruptcy support system: taking advantage of information retrieval and case-based reasoning. Expert Systems with Applications, vol.18, issue 3, 2000, pp. 215-219.

[10] Kyong Joo Oh, and Tae Yoon Kim. Financial market monitoring by case-based reasoning. Expert Systems with Applications, vol.18, issue 3, 2007, pp. 789-800.

[11] I. N. Glukhikh Adaptation in the recommendation systems with case based reasoning. In Advanced Studies in Science: Theory and Practice (Vol. 5). London, UK: The Global partnership LLC. doi: 10.17809/0557468427-43

[12] I.N. Gluhih, V.N. Piankov, A.R. Zabolotnov Situational models in corporative knowledge base on geological-technical measures' know-how. Neftyanoe khozyaystvo - Oil Industry, №6, 2002, pp. 45.

[13] V.I.Vasiliev, V.A. Pestrikov, A.S. Krasko Intellektualnaya podderzhka prinyatiya resheniya v ekstremalnykh situatsiyakh na osnove vyvoda po pretsedentam. Izvestiya UFU. Tekhnichesliye nauki, 2008. Vyp.8, s.5-13.

[14] P.R. Varshavskiy, A.P. Yeremeyev Modelirovaniye rassuzhdeniy na osnove pretsedentov v intellektualnykh sistemalh prinyatiya resheniy. Iskusstvenny intellect I prinyatiye resheniy, №2, 2009, s. 45-57.

[15] D.A. Pospelov Situatsionnoye upravleniye. Teoriya I praktika. M.: Nauka, 1986, 288 s.

[16] Susan Craw, Nirmalie Wiratunga, Ray C. Rowe. Learning adaptation knowledge to improve case-based reasoning. Artificial Intelligence, vol. 170, 2006, pp. 1175–1192.

[17] I. Gluhih Predstavlenie znanij i vyvod reshenij v situacionnyh bazah znanij//Vestnik Tjumenskogo gosudarstvennogo universiteta. – 2006. - № 5. – P. 265-270.

[18] I. Gluhih Intellectualnye informacionnye systemy. – Moscow, Academy, 2010, 112p.

[19] R. Bergmann, A. Stahl Similarity measures for object-oriented case representations. In Proceedings of the Fourth European Workshop on Case-Based Reasoning. - Berlin: Springer, 1998, pp.25-36.

[20] Bin He, Zhongqiang Deng and Haifeng Lv. Object-oriented Knowledge Modelling for Conceptual Design of Mechanisms. International Journal of Database Theory and Application, vol.6, No.5, 2013, pp.67-84.

[21] Dattatraya Vishnu Kodavade, and Shaila Dinakar Apte. A Universal Object Oriented Expert System Frame Work for Fault Diagnosis. International Journal of Intelligence Science, № 2, 2012, pp. 63-70.

[22] Nabil Mohammed Ali Munassar, A. Govardhan. Comparison between Traditional Approach and Object-Oriented Approach in Software Engineering Development. International Journal of Advanced Computer Science and Applications, vol. 2, No. 6, 2011, pp. 70-76.