A Case-Based Reasoning (CBR) approach for Engineer-To-Order systems performance evaluation

Abdourahim Sylla * **Thierry Coudert** ** **Laurent Geneste** ***

* *University Grenoble Alpes, CNRS, G-SCOP, Grenoble 38 000, France (e-mail: abdourahim.sylla@grenoble-inp.fr).*
** *University Toulouse, INP ENI Tarbes, LGP, Tarbes 65 000, France (e-mail: thierry.coudert@enit.fr)*
*** *University Toulouse, INP ENI Tarbes, LGP, Tarbes 65 000, France (e-mail: laurent.geneste@enit.fr)*

Abstract: In Engineer-To-Order (ETO) industrial contexts, decision-making about which systems to offer for sales or which subsystems to integrate into the systems for sales is based on various performance indicators. However, the lack of relevant information to evaluate these indicators challenges the decision-making process. In this article, to face this issue, a CBR approach is proposed for the evaluation of the performances of ETO systems. The main contributions are : (i) an object-oriented case representation model which allows to store previous evaluated systems for an effective and time efficient similar systems retrieval, and (ii) a method that allows to compute similarity between two systems that can have different structures.

Keywords: Complex systems design, Performance evaluation, Decision making, Engineer-To-Order, Case-based Reasoning

1. INTRODUCTION

In Engineer-To-Order (ETO) industrial situations, companies must totally or partially define novel or adapted systems in order to cover the customer requirements Sylla et al. (2018). When defining such systems, called non-standard systems, decision-making about which systems to offer for sales or which subsystems to integrate into the systems for sales is based on various performance indicators. They characterize the system and its development process. Some examples of performance indicators are cost, feasibility, complexity and risks. A reliable evaluation of these indicators is crucial in order to foster right decisions.

However, most of the time, in ETO contexts, relevant information to the evaluation of a non-standard system is not fully available Sylla et al. (2020). In some companies, to overcome this issue, domain experts, based on their experiences, extrapolate the evaluation of standard (existing) systems in order to evaluate the non-standard (new) one. This may work when dealing with simple system with few subsystems and few integrations between subsystems. When dealing with complex systems, composed of numerous subsystems and integrations, the cognitive workload associated to the evaluation task is high and cannot be supported by a human brain. Moreover, a highly human-dependent approach can lead to inaccurate or inexact evaluation. Therefore, a computerized approach capable of reproducing human reasoning is needed in order to support the evaluation of complex systems in Engineer-To-Order industrial situations.

To face this issue, a Case-Based Reasoning (CBR) approach is proposed in this article. As explained in several works as Kolodner (1992) and Aamodt and Plaza (1994), CBR approach is an artificial intelligence based approach which allows to solve a new problem by finding similar past problems and reusing knowledge and information from those problems. In a CBR terminology, a problem situation is referred as a “case” or an “experience”. A past problem situation that has been studied so that its related knowledge and information can be reused to solve new similar problems is called “previous case” or “source case”. Similarly, a new problem situation is called “new case” or “target case”. CBR approach has been used for several evaluation problems, especially effort estimation in software development Wu et al. (2018), cost estimation in new product development Relich and Pawlewski (2018) and prediction of the roughness and residual stress of machined surface Xu et al. (2020). In these works, all systems are considered to have the same structure, which means that all systems are defined by the same set of features. Consequently, the proposed CBR approaches do not consider situations where the system to be evaluated has different structure from the past existing systems. However, in Engineer-To-Order industrial situations, a non-standard system maybe defined by adding non-standard values for standard features or adding non-standard features or subsystems to standard systems.

In this article, we consider Engineer-To-Order industrial situations and propose a CBR approach for the evaluation of complex systems in domains where systems may have different structures. The main contributions are: (i) a “complex system performance evaluation” case representation model which allows to model and store system evaluation situations in a way that fosters an effective and time efficient case retrieval, and (ii) a simple method that exploits the case base structure to retrieve the most similar cases from the case base, taking into account common and non-common features of the target case and the previous cases. The rest of the paper is structured as follows. Next section provides the fundamentals of CBR approach for problem solving. Section 3, 4 and 5 describe the proposed CBR approach. Section 6 presents an illustrative application dealing with the evaluation of the cost of a tower crane system. Finally, Section 7 presents conclusion and future research.

2. FUNDAMENTALS OF CBR SYSTEMS

A CBR system is generally described by a cycle composed of five main phases. They are described in the following Aamodt and Plaza (1994).

- Define: a new problem is described in order to compare it to past problems stored in the case base.
- Retrieve: using a similarity measure, most similar problems in the case base are searched and selected.
- Reuse: knowledge or information related to the selected cases are reused to propose an initial solution.
- Revise: if the initial solution is not suitable, it is revised to make it more convenient.
- Retain: the new problem and its related information and knowledge are stored in the case base.

Two major challenges related to the development of an effective CBR system are to design a suitable case representation model and to define a relevant similarity measure. A case representation model is used to describe the problems. As explained in Bergmann et al. (2005), it generally consists of a problem and its solution description. It makes it possible to compare two cases and to compute their similarity. It also allows to store already studied cases in a case base. An appropriate case representation model combined with a well-organized case base allow for an effective and time efficient case retrieval and reuse.

Various case representations have been proposed in the literature, namely feature vector, hierarchical, ontology, and object oriented Bergmann et al. (2005); Shaker and Elmogy (2015). With feature vector, each case is represented as a set of features (attributes-values) which describe the problem and its associated solution. In this setting, all cases must have the same structure. The similarity measure is based on the values of the attributes and the relative importance of each attribute. In an hierarchical representation framework, a case is represented at multiple levels of detail Bergmann and Wilke (1996). This allows, when finding similar cases to a new one, to retrieve appropriate cases at the same levels of detail Bergmann et al. (2005). With Ontologies, cases are represented with formal, explicit and sharable concepts and properties. Properties are used to describe the concepts and the relationships (hierarchical or non-hierarchical) between concepts Foguem et al. (2008). The similarity measure is generally based on the position of each case related concept in the hierarchy of concepts. With object-oriented representations, each case is represented as an object which is described by a set of attribute-value pairs Bergmann et al. (2005).

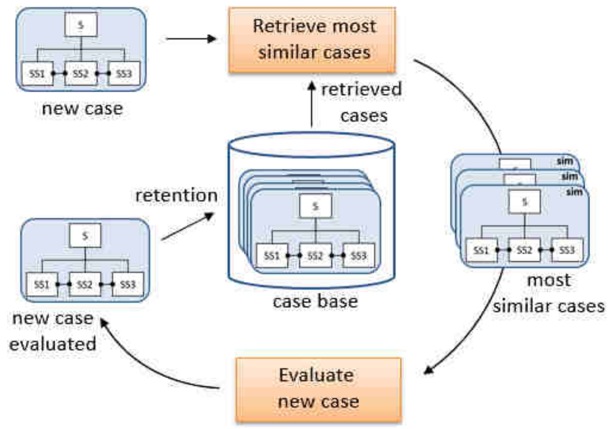

Fig. 1. CBR inspired cycle for ETO system evaluation

The object-oriented framework contains powerful relations which allow to describe complex cases in domains where cases may have different structures. Bergmann and Stahl (1998) proposed two measures to compute the similarity of two cases (Intra-Class and Inter-Class) which are based on the class attributes.

In this work, as we consider ETO industrial contexts where systems may have different structures, we have chosen the object oriented representation to develop the CBR approach. It is based on the CBR inspired cycle shown in Figure 1. When a new system must be evaluated, it is represented using the case representation model proposed in Section 3. Then, it is mapped to the case base and the most similar cases are retrieved and selected using the method proposed in Section 4. Finally, knowledge and information related to the selected cases are used to evaluate the performance of the new system using the method proposed in Section 5. The new system, when evaluated, is added to the case base along with its related knowledge and information.

3. CASE REPRESENTATION FOR COMPLEX SYSTEMS EVALUATION

A complex system can be defined as a set of subsystems which are integrated through adequate interfaces Henderson and Clark (1990). Each system or subsystem is, on one hand, defined by a set of properties (example: length and stiffness of a crane jib) and, on the other hand, characterized by a set of performance indicators such as cost, feasibility, complexity and risks. Based on that and using object-oriented framework, a "complex system performance evaluation" case representation is defined as follows:

- a case is a specific "system evaluation situation" and it is represented by an object which is an instance of a class;
- a class may be composed of one or several other classes and classes are organized in a class hierarchy;
- an object's class determines its attributes which may be a property, a performance indicator or an object;
- the collection of all objects represents the case base.

The upper part of Figure 2 depicts an UML class diagram which is an object-oriented case representation of a tower

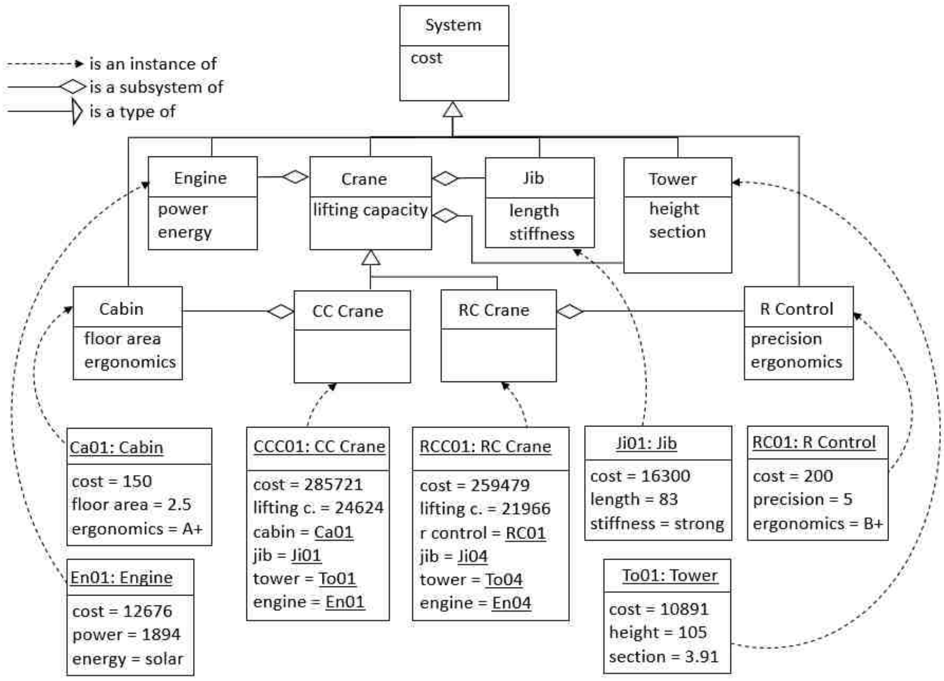

Fig. 2. An example of object-oriented case representation

crane system. In this diagram, a crane is a kind of system defined by the class *Crane*. It is composed of three subsystems (a tower, a jib and an engine) which are also kinds of systems described by the classes *Tower*, *Jib* and *Engine*, respectively. There are two kinds of crane system: Cabin Controlled Crane (*CC Crane*) and Remote controlled Crane (*RC Crane*). They are specializations of the class *Crane*. In addition to attributes (*cost, lifting capacity, engine, jib and tower*) inherited from the class *Crane*, they have specific attributes: {*Cabin*} for *CC Crane* and {*R Control*} for *RC Crane*. At the lower part of Figure 2 are shown examples of previous system performance evaluation cases. For instance, *CCC01* is a case of a cabin controlled crane system performance evaluation and *Ca01* is a case of a cabin system performance evaluation. In the next section, the method that allows to retrieve similar previous system evaluations for the evaluation of a new system is presented.

4. CASES RETRIEVAL AND SIMILARITY MEASURE FOR COMPLEX SYSTEMS

We assume that one may use our CBR system to evaluate a system for sale (crane system in the example of Figure 2) or a system which is a part of the system for sale (Jib system in Figure 2). In the latter situation, the CBR system provides an evaluation of a part of the system for sale, which can be used through another evaluation method to evaluate the system for sale. This is particularly appropriate in ETO situations where few parts of an existing system must be modified to develop a new system. In this article, we propose a generic approach which allows to retrieve the most similar cases in both situations (see Figure 3). The main steps are described in the following.

1.Relevant case retrieval. At the first step, the structure of the case base is exploited to directly retrieve relevant previous cases to the target case. In fact, as each

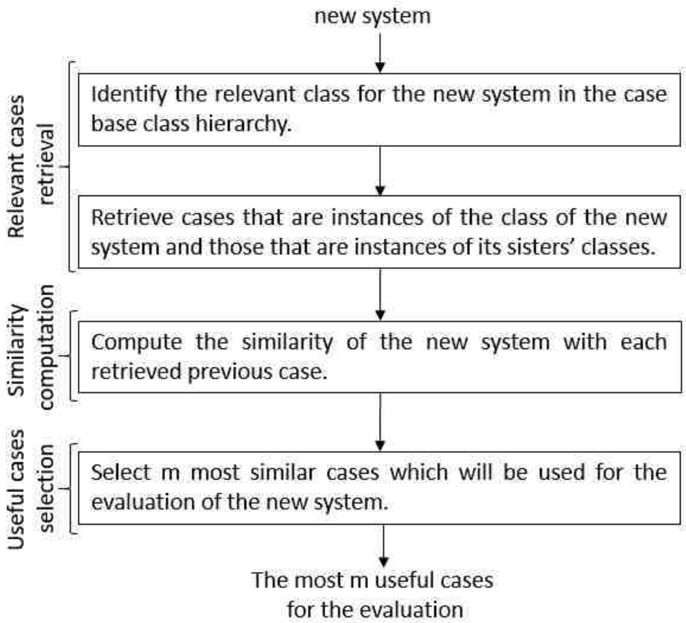

Fig. 3. The selection process of the most similar cases

class of the case base is built based on a set of common properties of systems, the case base structure contains general knowledge about which systems are of the same type and which systems are not of the same type. For instance, *CCC01* and *RCC01* are of the same type, they are all cranes. In the contrary, *CCC01* and *CaO1* are not of the same type, *CaO1* is a cabin and not a crane. Exploiting the case base structure allows to reduce the computational time related to search and similarity computation. Practically, once a new system must be evaluated, its properties are extracted in order to identify the relevant class in the case base class hierarchy. Only cases that are instances of the class of the target case and those that are instances of its "sisters' classes" are retrieved. We define "sisters' classes" as classes that have a Direct Common Superclass (DCS). For example, in Figure 2, *CC Crane* is a sister class of *RC Crane*. The DCS of *CC Crane* and *RC Crane* is the class *Crane*. Therefore, if the target case is an instance of the class *CC Crane*, all systems that are instances of the classes *CC Crane* and *RC Crane* are retrieved for further similarity computation. The other cases that are not instances of *CC Crane* and *RC Crane*, *CaO1* for example, are not compared to the target case.

2.Similarity computation. After that, the similarity of the target case C_t with each retrieved previous case C_p is computed. The two situations that can be encountered are considered. In the first situation, the target case and the previous case are instances of the same class. The two cases have the same set of attributes $\Omega(C_t) = \Omega(C_p)$, see the upper part of Figure 4. In the second situation, the target case and the previous case are instances of different classes. The two cases have a set of common attributes $\Omega(C_t) \cap \Omega(C_p)$ inherited from their Direct Common Superclass (DCS). In addition, at least, one case has one or more attributes that the other case does not have, see lower part of Figure 4. Based on the similarity computation method in feature-vector approach, a simple, effective and easy to implement method is proposed to compute similarity of two cases in any of the above mentioned two situations. The global similarity of the two cases, noted $\mathrm{Sim}(C_t, C_p)$, is a real number between 0 and 1. It is computed as a weighted average of the local similarities of attributes, noted $\mathrm{Sim}(att_i^t, att_i^p)$, see Equation 1. The weight w_i of an attribute indicates its relative importance to the performance indicator being evaluated. In order to take into account the two situations mentioned above, the following hypothesis are considered for the computation of the local similarities:

- For an attribute that characterizes both target and previous cases, the local similarity is a real number between 0 and 1. In fact, such an attribute tends to increase the global similarity of the two systems. If the values for the attributes for the two systems are the same, the total weight w_i of the attribute is added to the global similarity. If not, a percentage (which corresponds to the local similarity) of w_i is added to the global similarity.
- For an attribute that characterizes only one of both cases (target or previous) the local similarity is equal to 0. In fact, such an attribute tends to decrease the global similarity of the two systems. By setting the local similarity to 0, the weight w_i of the attribute is not added to the global similarity. It is removed from the maximum global similarity which becomes 1 - w_i. So that, if the attribute in consideration has a strong impact on the performance indicator to evaluate, the global similarity of the two cases will be low.

In the context of complex systems evaluation, the attributes may be simple or relational. The values of simple attributes can take multiple forms, namely crisp symbols and crisp numbers. In example of Figure 2, for the "jib system", the value of its attribute "stiffness" may be "strong" and the value of its attribute "length" may be "100 meters". The value of a relational attribute is an object. Depending on the form of an attribute, a different

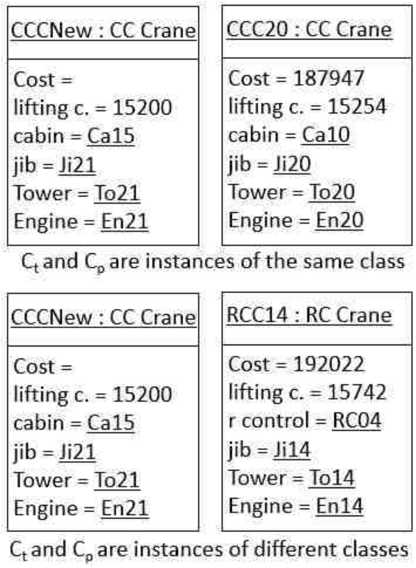

Fig. 4. Two situations of similarity computation

	A+	A	B+	B	C
A+	1.0	0.75	0.5	0.25	0
A	0.75	1.0	0.75	0.5	0.25
B+	0.5	0.75	1.0	0.75	0.5
B	0.25	0.5	0.75	1.0	0.75
C	0	0.25	0.5	0.75	1

Fig. 5. A similarity matrix for the ergonomics attribute

method is used to compute the local similarity. Let q_{pi} and q_{ti} denote the values of an attribute att_i for a previous case C_p and a target case C_t. If q_{pi} and q_{ti} are in the form of crisp numbers, the local similarity is computed using equation 2. If q_{pi} and q_{ti} are in the form of crisp symbols, the local similarity is computed using a similarity matrix as shown in Figure 5. If q_{pi} and q_{ti} are objects (i.e. subsystems), the local similarity is recursively computed using equation 1.

$$\begin{aligned} &Sim(C_t, C_p) = \sum_{i=1}^{k} w_i * Sim(att_i^t, att_i^p) \\ &0 < w_i \leq 1 \, and \sum_{i=1}^{k} w_i = 1 \\ &\forall i \in \Omega(C_t) \cap \Omega(C_p),\, 0 \leq Sim(att_i^t, att_i^p) \leq 1 \\ &\forall i \in \Omega(C_t) \backslash \Omega(C_p), Sim(att_i^t, \{\}) = 0 \\ &\forall i \in \Omega(C_p) \backslash \Omega(C_t), Sim(att_i^p, \{\}) = 0 \end{aligned} \tag{1}$$

$$sim(att_i^t, att_i^p) = 1 - \frac{|q_{pi} - q_{ti}|}{\max_{p \in \phi} \{|q_{pi} - q_{ti}|\}} \tag{2}$$

ϕ *is the set of the retrieved previous cases.*

One can notice that the weights of the attributes take a great place in this method. The efficiency of the similarity assessment strongly relies on what weights reflects the relative importance of the attributes with regards to the performance indicator being evaluated Doan et al. (2006). Several methods have been proposed to determine the weights of attributes in CBR systems, namely Analytic Hierarchy Process (AHP), Artificial Neural Network (ANN) and optimization techniques. In this article, we use AHP method Saaty (1980). AHP is a systematic method which uses knowledge of domain experts to determine the weight (relative importance) of each attribute with respect to the performance indicator being evaluated. First, a pairwise comparison of the attributes is performed in order to have the input data. Second, the obtained pairwise comparison matrix is used to compute the weight of the attribute. Finally, a consistency ratio is computed to examine the consistency of the weights determination process. As AHP is a well-established method that has been described in many researches, we do not describe it in detail. More details can be found in numerous articles, especially Saaty (1980).

3. Useful cases selection. After computing the similarity of the target case with each previous case, based on a "similarity threshold" and a "maximum number of useful cases", the most useful cases are selected for the evaluation of the target case. The similarity threshold allows to define a minimum similarity value under which a previous case is not considered similar enough to the target case in order to be considered for the evaluation. The maximum number of useful cases is used to limit the number of cases in the evaluation process. It allows to reduce the computational time, especially in situations where numerous previous cases are retrieved. In the following, we describe how to evaluate a new system using knowledge and information related to past similar systems.

5. EVALUATION OF THE TARGET SYSTEM

In this phase, the performance *perf* of the target case C_t is evaluated using the evaluations of the m most similar previous cases retrieved and selected from the case base (C_j^p, j = 1 to m). We assume that one may want to revise the evaluations of the selected previous cases in order to make them conform to the evaluation context of the target case. This is particularly appropriate in situations where the time difference between the evaluations of previous and target cases is long. For example, when evaluating the cost a new system, the inflation factor due to the time difference must be considered. Let l be a factor that characterizes the evaluation context of the target case and K_l its revision coefficient. The revised evaluation of the performance of a selected previous case C_j^p, noted $E_r^{perf}(C_j^p)$, is computed using Equation 3. In this equation, $E^{perf}(C_j^p)$ is the original evaluation of the previous case C_j^p. Then, the evaluation of the target case is performed as the weighted sum of the revised evaluations of the previous cases (see Equation 4). The weight W_j of each case being the normalized value of its similarity to the target case. It is computed using Equation 5.

$$E_r^{perf}(C_j^p) = K_l * E^{perf}(C_j^p) \tag{3}$$

$$E^{perf}(C_t) = \sum_{j=1}^{m} W_j * E_r^{perf}(C_j^p) \tag{4}$$

$$W_j = \frac{Sim(C_t, C_j^p)}{\sum_{j=1}^{m} Sim(C_t, C_j^p)} \tag{5}$$

In the next section, an illustrative application is presented to show how the proposed approach can be used.

6. ILLUSTRATIVE APPLICATION

The aim of this section is to show how to use the proposed CBR system to evaluate a new system during an engineering design process. The example concerns the evaluation of the cost of a new crane. Following the proposed case representation model presented in Section 3, the new crane, called C_t, is defined as follows.

C_t = {*lifting capacity = 15200 kg, cabin = Ca15 =* {*floor area = 2.5* m^2*, ergonomics = A+*}*, jib = Ji21 =* {*stiffness = strong, length = 96 m*}*, tower = To21 =* {*height = 108, section = 3.77* m^2}*, engine = En21 =* {*power = 1890, energy = solar*}}.

The attributes of the crane C_t allow to identify the corresponding class in the case base class hierarchy. It is the class *CC Crane*, and C_t has one "sisters' class" which is *RC Crane* (see Figure 2). Therefore, as explained in Section 4, only cases which are instances of the classes *CC Crane* and *RC Crane* are retrieved for further similarity computation. The other cases are not retrieved as they are not cranes. This enables to directly retrieve the relevant cases to the target case. Thus, it allows to reduce the computational time related to the retrieval and similarity computation process. In total, fifty previous cases is retrieved. A sample is shown in Table 1. After that, the similarity computation method presented in Section 4 is used to compute the similarity of each previous case with the target case. For the seek of clarity, all details about the similarity computation are not shown. Instead, we choose to present a sample of the final result in Table 2. In this table, *weight 1* is the weights of the attributes when the target and previous cases are instances of the same class, whereas *weight 2* is the weights when they are not instances of the same class. It is important to notice that the proposed method supports not only the similarity computation between two systems of the same structure (e.g. C_t and *case 1* in Table 2), but also the similarity computation between two systems of different structures (e.g. C_t and *case 4* in Table 2). For the evaluation of the cost of the new system, in order to select cases that have a great similarity to the target case, the similarity threshold was fixed to *0.90.* Only three cases (case 12, case 14 and case 20) correspond to this requirement (see Table 2). As the maximum number of useful cases is set to *5*, the costs of the three cases are used to compute the cost of the new system using the method presented in Section 5. The revision coefficient is supposed *1*, which means that a revision is not necessary in this context. The final result of the evaluation of the cost of the target case is *192016.45.*

7. CONCLUSION

In this article, a CBR approach has been proposed for the evaluation of the performance of engineer-to-order systems during engineering design process. It is based on an object-oriented case representation which allows to store previous cases for effective and time efficient similar cases retrieval. The proposed approach is applicable in domains where the system to evaluate can have different structure with the

Table 1. A sample of retrieved previous cases

	lifting c.	stiffness	length	height	section	power	energy	floor area	ergo.	precision	ergo.	cost
case 1	24624	strong	83	105	3.91	1894	solar	2.5	A+	-	-	285721
case 2	23542	strong	88	98	3.68	1811	gasoline	2	A	-	-	273378
case 4	21966	strong	100	110	3.59	1690	solar	-	-	5	B+	259479
...	...	...	...	...	...	...	...	...	...	...	...	...
case 12	16075	strong	104	114	3.94	1237	electric	2.2	C	-	-	196304
case 14	15792	strong	98	108	3.72	1195	gasoline	-	-	3	A	192022
case 20	15254	strong	106	118	3.77	1173	electric	2.5	A+	-	-	187947
case 25	7380	medium	50	60	2.84	568	gasoline	-	-	4	B+	91574
case 43	897	low	32	42	2	69	gasoline	2	B	-	-	17700

Table 2. A sample of local and global similarities

	lifting c.	stiffness	length	height	section	power	energy	floor area	ergo.	precision	ergo.	global sim.
weight 1	0.179	0.141	0.163	0.165	0.142	0.179	0.002	0.020	0.010	-	-	-
weight 2	0.171	0.134	0.155	0.157	0.136	0.171	0.001	0.019	0.009	0.016	0.029	-
case 1	0.35	1.0	0.83	0.96	0.94	0.34	0.5	1.0	1.0	-	-	0.72
case 2	0.42	1.0	0.89	0.87	0.96	0.42	0.3	0.0	0.75	-	-	0.73
case 4	0.53	1.0	0.95	0.97	0.92	0.53	0.5	0.0	0.0	0.0	0.0	0.73
...	...	...	...	...	...	...	...	...	...	...	...	...
case 12	**0.94**	**1.0**	**0.89**	**0.92**	**0.92**	**0.94**	**1.0**	**0.4**	**0.0**	**-**	**-**	**0.91**
case 14	**0.96**	**1.0**	**0.97**	**1.0**	**0.98**	**0.97**	**0.3**	**0.0**	**0.0**	**0.0**	**0.0**	**0.9**
case 20	**1.0**	**1.0**	**0.87**	**0.89**	**1.0**	**0.99**	**1.0**	**1.0**	**1.0**	**-**	**-**	**0.96**
case 25	0.46	0.5	0.39	0.37	0.58	0.46	0.3	0.0	0.0	0.0	0.0	0.43
case 43	0.01	0.25	0.16	0.13	0.2	0.01	0.3	0.0	0.25	-	-	0.12

previous evaluated systems. An illustrative application is presented to show how to use the proposed approach. As a future research, we intend to apply the proposed CBR system to use cases from industries in order to show its applicability and effectiveness.

REFERENCES

Aamodt, A. and Plaza, E. (1994). Case Based Reasoning : foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), 39–59.

Bergmann, R., Kolodner, J., and Plaza, E. (2005). Representation in case-based reasoning. *The Knowledge Engineering Review*, 00(0), 1–4. doi: 10.1017/S0000000000000000.

Bergmann, R. and Stahl, A. (1998). Similarity Measures for Object-Oriented Case Representations. In *European Workshop on Advances in Case-Based Reasoning*, 25–36. Springer Berlin Heidelberg.

Bergmann, R. and Wilke, W. (1996). On the Role of Abstraction in Case-Based Reasoning. In *European Workshop on Advances in Case-Based Reasoning*, 28–43. Springer, Berlin, Heidelberg.

Doan, S.Z., Arditi, D., and Günaydn, H.M. (2006). Determining Attribute Weights in a CBR Model for Early Cost Prediction of Structural Systems. *Journal of Construction Engineering and Management*, 132(10), 1092–1098. doi:10.1061/(asce)0733-9364(2006)132:10(1092).

Foguem, B.K., Coudert, T., Bler, C., and Geneste, L. (2008). Knowledge formalization in experience feedback processes : An ontology-based approach. *Computer in Industry*, 59, 694–710. doi: 10.1016/j.compind.2007.12.014.

Henderson, R.M. and Clark, K.B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*, 35(1), 9–30.

Kolodner, J.L. (1992). An Introduction to Case-Based Reasoning. *Artificial Intelligence Review*, 6, 3–34. doi: 10.1136/bmj.4.5576.398.

Relich, M. and Pawlewski, P. (2018). A case-based reasoning approach to cost estimation of new product development. *Neurocomputing*, 272, 40–45. doi: 10.1016/j.neucom.2017.05.092.

Saaty, L.T. (1980). The analytic hierarchy process.

Shaker, H. and Elmogy, M. (2015). Case Based Reasoning: Case Representation Methodologies. *International Journal of Advanced Computer Science and Applications*, 6(11). doi:10.14569/ijacsa.2015.061126.

Sylla, A., Coudert, T., Vareilles, E., Geneste, L., and Aldanondo, M. (2020). Possibilistic Pareto-dominance approach to support technical bid selection under imprecision and uncertainty in engineer-to-order bidding process. *International Journal of Production Research*, (in press). doi:10.1080/00207543.2020.1812754.

Sylla, A., Guillon, D., Vareilles, E., Aldanondo, M., Coudert, T., and Geneste, L. (2018). Configuration knowledge modeling : How to extend configuration from assemble / make to order towards engineer to order for the bidding process. *Computers in Industry*, 99, 29–41. doi:10.1016/j.compind.2018.03.019.

Wu, D., Li, J., and Bao, C. (2018). Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation. *Soft Computing*, 22(16), 5299–5310. doi:10.1007/s00500-017-2985-9.

Xu, L., Huang, C., Li, C., Wang, J., Liu, H., and Wang, X. (2020). An improved case based reasoning method and its application in estimation of surface quality toward intelligent machining. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-020-01573-2.