ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# HYBRID CASE RETRIEVAL USING FEATURE-VECTOR CASE REPRESENTATION IN A CBR E-LEARNING SYSTEM

**ABROUN SOUNDOUSS[1], GHAILANI MOHAMED[2], FENNAN ABDELHADI[3]**

[1, 3] Department of Computer Science, Laboratory LIST, Faculty of Science and Technology, Tangier,90000, Morocco

[2]Department of Computer Science. Laboratory LABTIC, National School of Applied Sciences, Tangier, 90000, Morocco

E-mail : [1]soundoussabroun@gmail.com, [2]ghalamed@gmail.com, [3]afennan@gmail.com

**ABSTRACT**

The COVID-19 pandemic has spurred the rapid adoption of e-learning systems, necessitating effective personalization strategies to cater to diverse learner needs. However, existing e-learning platforms often face challenges in providing tailored learning experiences. While existing Case-Based Reasoning (CBR) approaches in e-learning hold promise for personalization, their effectiveness hinges on robust case representation and retrieval methods. This paper addresses these limitations by proposing a hybrid case retrieval approach using feature-vector case representation for an Adaptive Intelligent Educational Distributed Case-Based Reasoning (AIED-CBR) system to capture comprehensive learner profiles This approach combines the strengths of rule-based and K-Nearest Neighbors (KNN) techniques within a Multi-Agent System architecture to enhance the efficiency and accuracy of case retrieval in personalized learning path generation. We leverage ontologies for knowledge description, facilitating efficient reasoning and knowledge sharing within the system. We present the system's architecture, detailing the hybrid retrieval mechanism and its integration with multi-agent collaboration, and the role of ontologies. This hybrid approach addresses limitations of existing CBR-based e-learning systems, offering the potential to create more effective and adaptable personalized learning experiences.

**Keywords:** *Case-Based Reasoning (CBR), Adaptive E-learning systems (AES), Ontologies, Multi-Agent System, K-Nearest Neighbors (KNN), Rule-based reasoning, Case representation.*

## 1. INTRODUCTION

The unprecedented global shift towards online learning due to the COVID-19 pandemic has underscored the critical need for e-learning systems that can deliver personalized and effective learning experiences tailored to individual learner needs and preferences. While traditional e-learning platforms often employ generic instructional methods, personalized learning can significantly enhance engagement, retention, and overall educational outcomes. Case-Based Reasoning (CBR) emerges as a promising paradigm in this context, leveraging past experiences to inform adaptive learning pathways and recommendations.

However, the effectiveness of a CBR system heavily relies on its case representation, which influences the similarity assessment between new learners (target cases) and past learning experiences (stored cases). This, in turn, affects the system's ability to suggest appropriate learning paths. Current CBR research in e-learning acknowledges this, with efforts focused on developing more robust case representations.

Moreover, the efficiency of case retrieval approach is essential for accurately matching new learner profiles with relevant past cases, enabling the system to recommend personalized learning paths effectively. Different case retrieval techniques exist for CBR systems, aiming to enhance the adaptability and responsiveness. Notably, [1] reviewed widely used methodologies in this area. A well-designed case retrieval method not only improves recommendation accuracy but also contributes to the overall efficacy and user satisfaction of personalized learning experiences.

Furthermore, the incorporation of learner feedback for continuous case evaluation is essential for refining and improving the adaptability of CBR systems in e-learning. By actively soliciting and integrating user input, such systems can iteratively optimize recommendations, ensuring they align with evolving learner needs and preferences. This iterative feedback loop fosters a dynamic learning environment where recommendations become increasingly personalized and effective over time

.

In this paper, we propose a novel approach to address these challenges by introducing a robust feature-vector case representation tailored for e-learning CBR systems. Our method aims to capture nuanced learner attributes comprehensively, enabling precise similarity assessments and personalized recommendations. Additionally, we advocate for a hybrid case retrieval strategy that combines rule-based reasoning with the K-nearest neighbor algorithm to enhance retrieval efficiency and accuracy. By emphasizing the importance of relevant case retrieval methods and the integration of learner feedback, our research aims to advance the state-of-the-art in personalized e-learning systems, ultimately enhancing the educational experience for learners worldwide. Additionally, we introduce the use of ontologies within the system. Ontologies provide a structured representation of domain knowledge, facilitating efficient reasoning, knowledge sharing, and interoperability between different agents in a multi-agent CBR system.

The following sections of this paper are organized as follows: in Section II, we present the case-based reasoning approach along with a review of some e-learning CBR systems using hybrid approaches for case retrieval. In Section III, we discuss the rule-based approach and its efficiency in CBR systems. In Section IV, we provide a comparative analysis of some supervised machine learning techniques that could be used for case retrieval in CBR systems. In Section V, we present the system architecture and technical architecture of the proposed CBR e-learning system using a hybrid approach for case retrieval. Finally, we draw some conclusions and perspectives in Section VI.

## 2. CASE BASED REASONING FOR E-LEARNING SYSTEMS

Case-Based Reasoning (CBR) stands as a prominent Artificial Intelligence methodology widely embraced across diverse domains ([2], [3], [4]) to tackle adaptation challenges. This paradigm holds significant promise for enhancing the effectiveness of complex and unstructured decision-making processes. The fundamental aim of the CBR approach is to address a current target problem, denoted as the "target case," by leveraging solutions from past cases known as the "source solution." These pairs, comprising (source_problem, source_solution), are commonly referred to as "source cases," collectively forming the "Case Base." This resolution process can be achieved following these steps [5]:

- **Elaborate:** Reformulate a clear description of the target case from the submitted request
- **Retrieval:** Finding similar cases from the case base based on the current problem.
- **Reuse:** Adapting the solution from a retrieved case to the new problem.
- **Revision:** Evaluating and potentially refining the adapted solution.
- **Retention:** Integrating the new experience (including the adapted solution) as a new case into the case base for future use.

Various works in the field of e-learning leverage CBR to personalize learning experiences for individual learners. These works include Case Based Reasoning Adaptive E-Learning System Based On Visual-Auditory-Kinesthetic Learning Styles [6]: An Adaptive E-learning System that tailors learning materials to learner styles (visual, auditory, kinesthetic), utilizing a feature-based approach for case representation and employing the CBR approach with the Nearest-Neighbor algorithm for case retrieval to provide suitable pedagogical content. MOOC-Rec [7] Is a CBR-based recommender system for MOOCs that suggests suitable courses based on learner profiles and past learning experiences. Cases in Mooc-rec are created by extracting learning source features from XML pages of MOOCs indexed and listed in online directories dedicated to MOOCs, these cases are represented as a simple (problem, solution) pair where Problem (user's query), Solution (MOOCs' URL)). For relevant cases retrieval, MOOC-Rec employs a k-d binary tree that organizes cases based on their similarity. The Levenshtein distance is used as a similarity measure to assess the local similarity between case attributes. Additionally, the system calculates similarity bounds to prioritize case clusters during the search, ensuring efficient retrieval of the most relevant MOOC recommendations for each learner. IHCBR [8] introduces a novel approach to adaptive learning systems by integrating multi-agent architecture with incremental hybrid case-based reasoning (IHCBR

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

which combines rule-based reasoning and case-based reasoning. Rule-based reasoning provides initial guidance based on domain knowledge, while case-based reasoning leverages past learning experiences to adapt learning paths further. The multi-agent architecture enables collaborative problem-solving and decision-making among various agents responsible for different aspects of the learning process, including case retrieval, adaptation, and evaluation. [9] A Hybrid Recommender System Using Rule-Based and Case-Based Reasoning this research explores a hybrid recommender system that combines rule-based and case-based reasoning for personalized recommendations. It highlights the complementary nature of these techniques, where rules provide initial filtering and CBR refines recommendations based on specific user profiles.

These studies highlight the advantages of the use of multi-agent systems and hybrid approaches in personalized learning, demonstrating the potential for integrating different reasoning techniques to overcome limitations and achieve more effective adaptation. However, some limitations remain, such as the insufficient knowledge diversity in cases as they don't cover all the information needed for an optimal adaptability as for [5] and [6]. Additionally, many of the referenced approaches may encounter the cold start problem especially with new cases as for [5], |6] and [8]. Furthermore, the systems discussed typically overlook user feedback on the recommended learning paths, impacting the improvement of the efficiency of the proposed solution. All These identified limitations have the potential to adversely impact the adaptability and the effectiveness of the learning system.

In response to these challenges, our proposed solution involves a comprehensive case representation that encompasses detailed learner information. We advocate for a hybrid case retrieval approach that enables precision through two-phase case extraction. Furthermore, we emphasize the importance of incorporating user feedback to assess content clarity, adaptability, and efficiency, thereby enhancing the overall effectiveness of the learning system.

## 3. FEATURE-VECTOR CASE REPRESENTATION

In the Adaptive CBR E-learning system (ACES), case representation is a key problem that affects information gathering and processing. ACES collects information about learners' preferences, learning styles, prerequisite and their behavior while using the system, for helping the system to adapt to each individual learner. This information is typically defined as a feature vector where each variable represents a type of information about the learner and the concrete value of it.

In a vector representation each feature corresponds to a particular characteristic or attribute of the represented data.[10] It is a measurable property aiding in the description or the distinction of data in the vector. Feature vectors play an important role in machine learning algorithm, serving to characterize the individuals or data instances and enabling the execution of various tasks such as classification, regression, clustering, and more.[11] The choice of appropriate feature vectors significantly influences the success of machine learning models. Therefore, it is crucial for features to be meaningful, discriminating, and independent attributes, ensuring the effectiveness of the machine learning algorithms employed.

Several studies have been conducted to establish a well-defined learner model using feature-vector representation [12] [13]. However, these systems focus primarily on external factors such as learner behavior and performance, neglecting internal factors describing learner's learning style, preferences, proficiency level and personality. On the other hand, [14] proposes a novel method for learner model description using feature-vector representation that incorporates both external and internal information aiming to satisfy learners' personal preferences. It provides a clear description of learner information, including behavioral features, performance features and personal features. Our research draws inspiration from this work [14], adapting the proposed solution to our CBR e-learning system.

## 4. RULE-BASED REASONING

Rule-based reasoning (RBR) is an artificial intelligence (AI) approach that makes decisions or draws conclusions based on a set of explicitly defined rules. These rules typically follow an "if-then" structure, where the "if" part specifies conditions and the "then" part defines corresponding actions. RBR offers a structured and efficient way to encode domain knowledge and guide decision-making. in e-learning systems [15].

Rule-Based Reasoning (RBR) in CBR e-learning systems can utilize predefined rules to make decision and provide personalized recommendations for learning materials, courses, or activities based on learners' profiles, these rules are created based on expert knowledge or domain-specific principles.

Several CBR systems leverage RBR in conjunction with case-based reasoning [8],[9] as it provides several advantages in term of Explainability (Rules are often easier to understand and interpret compared to other reasoning techniques) and Efficiency (Predefined rules can facilitate quick decision-making in situations requiring immediate responses).

## 5. K-NEARSET NEIGHBOR ALGORITHM

K-nearest neighbors (KNN) is a commonly used case retrieval algorithm in case-based reasoning Recommender System (CBR-RS) systems. It is used to find similar cases from a case base based on their characteristics, due to its effectiveness and potential for improving the accuracy of CBR systems. Several papers discuss the use of KNN in case retrieval, such as [16], [17]and [4]. It is considered as one of the oldest and simplest approaches for pattern classification. Its principle comes from geometric measurement, where each unlabeled example is assigned the majority label of its k-nearest neighbors in the training set. Even though it is a simple approach, the KNN rule frequently produces competitive outcomes [18].

For e-learning Case-Based Reasoning (CBR) recommender systems, K-Nearest Neighbors (KNN) can be a valuable algorithm to achieve the retrieval process, in order to provide personalized recommendations to learners. Each case is typically represented as feature-vector, containing information about the learner, the learning content, and any contextual factors. The KNN algorithm identifies the K most similar cases (nearest neighbors) to the new case based on the calculated similarities. These nearest neighbors represent past learning experiences or recommendations that are most similar to the current learner's situation.

In addition, we need to carefully consider the choice of k, and the recognition rate of the classifier obtained by choosing different will vary to some extent. When the value of k is too small, the approximation error will decrease, but the estimation error will increase; when the value of k is too large, the estimation error will decrease, but the approximation error will increase.

Overall, K-nearest neighbor (k-nn) classifiers are appealing because of their simplicity, ability to model a wide range of parametric and non-parametric distributions, and theoretical optimality as the training size goes to infinity [18]. They can serve as well as a powerful technique in e-learning CBR recommender systems. what motivates us to use it as method for case retrieval in our Adaptive CBR e-learning system.

## 6. SYSTEM ARCHITECTURE:

In our proposed system, we represent cases using a four feature-vector to facilitate case utilization and extensibility. We adopt a hybrid approach combining rule-based and KNN algorithms for case retrieval. Additionally, we incorporate user feedback in case evaluation to enhance the effectiveness of learning experiences. Furthermore, we leverage several ontologies (competencies, domain knowledge, learning object, learner model, pedagogical training) for knowledge description. This combination of techniques aims to provide a robust and adaptable personalized learning experience within our e-learning system.

Inspired by [8],[14]and[9], we adopt the feature-vector method for case representation. Adapting the feature-vector representation to our CBR e-learning system, we divided cases (problems to solve) in our system into four subcases (subproblems), each described using the feature vector containing all the learner's features considered significant for the system's performance.

PI(La)= {L, P, FN, LN, A, M, C}
TR(La)= {G, C, PT, FSLM}
LT(La)= {CC, LOi, ti, NVLOi, RT, RN}
FB(La)= {QScore, PTScore}

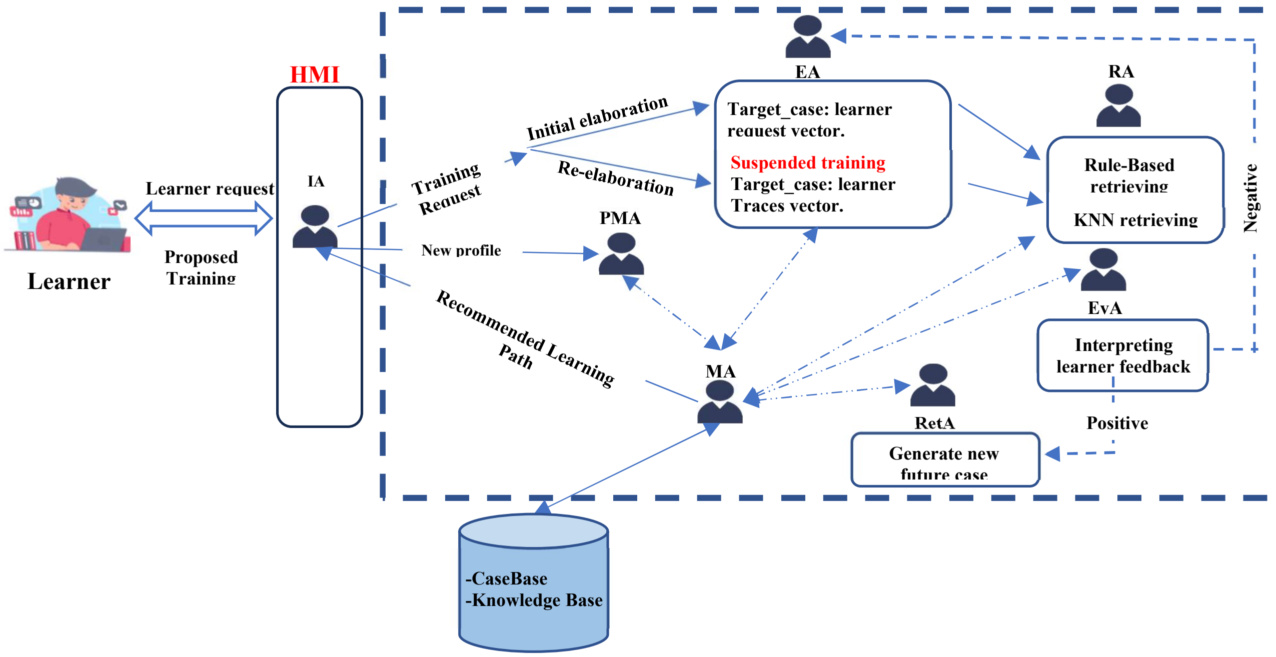


*Figure 1: The proposed Architecture of the Multi-Agent Hybrid CBR E-learning System*

Where PI(La) is the personal information vector of an active learner describing demographic information of the learner (Login (L), password (P), first_name (FN), last_name (LN), age (A), mail (M), country (C)). TR(La) is the Training Request vector defining the learner's goal for the pedagogical training (G), targeted competency (Comp), Result of the prerequisite test (PT) and the result of the Felder and Silverman Learning style test (FSLM). LT(La) is the learner traces vector, storing learner interactions considering Chosen Course (CC), sequence of the learning object training (LOi), Time (Ti) spent in each learning object (LOi), number of visits (NVLOi) of the learning object (Si), Type of resources (Definition, Notion, Example, Algorithm, Exercise, Quiz) (RT), Nature of resources (Text, Audio, Picture, Video) (RN). Finally, FB(La) is the learner feedback vector collecting opinions about the consumed pedagogical training considering: the score of a choice-based questionnaire gathering information about resource type adaptability, resource nature adaptability and provided learning object clarity; and the score of a post-training assessment measuring knowledge acquisition and retention.

This comprehensive feature-vector method facilitates a detailed representation of each subcase, providing essential data for the efficient functioning of the CBR cycle within our system.

### 6.1 Multi-agent CBR system:

The architectural design of the Adaptive Educational Distributed Case-Based Reasoning System (AEDCBRS), as illustrated in *Figure 1*, is the improved version of the architecture proposed in [19]. It intricately incorporates 9 collaborating agents to collectively address various functionalities crucial to the system. For personalized pedagogical training generation, a dynamic Case-Based Reasoning (CBR) cycle is employed and meticulously managed by six key agents to ensure the effective functioning of the reasoning cycle:

**Interface Agent (IA),** functioning as the ReactJS frontend, plays a pivotal role in receiving learning requests from users, encompassing their goals and preferences. Beyond this, it gathers pertinent information about the learner's existing knowledge and skills by utilizing the results of the prerequisite test and Felder-Silverman Learning Style Test Forms. Subsequently, it seamlessly communicates with the Manager Agent to initiate the CBR cycle. Its responsibilities extend to displaying learning content, thereby ensuring a tailored and user-friendly experience for learners.

**Manager Agent (MA),** implemented in Python FastAPI, acts as the orchestrator of both agent interactions and the Case-Based Reasoning (CBR) process flow. It plays a crucial role in managing the entire system by:

- **Assigning tasks**: Based on expertise and roles, the MA efficiently distributes tasks among other agents, ensuring optimal utilization of their capabilities.
- **Managing communication and data flow**: It acts as a central hub, coordinating seamless communication and data exchange between agents, fostering smooth collaboration.
- **Overseeing the problem-solving process**: The MA oversees the entire CBR process, ensuring its effective execution and progress.
- **Data management**: The MA also acts as a central repository, saving and updating all information sent by other agents. This ensures everyone has access to the latest data for informed decision-making.

**Profile Manager Agent (PMA)** operates as a Python FastAPI microservice interacting with MongoDB. This agent retrieves and provides comprehensive learner profiles, encompassing competencies, learning styles, and preferences. It is responsible for creating, updating, deleting, searching, and storing learner profiles in response to Interface Agent requests. Additionally, the PMA collaborates with the Manager Agent to retrieve relevant information about learners, learning objects, competencies, and pedagogical training sequences, offering support to the Elaboration Agent in its tasks.

**The Elaboration Agent (EA),** a Python FastAPI microservice, plays a pivotal role in constructing problem descriptions from learner profiles and requests. By analyzing the learner's profile and learning goals, the EA creates a detailed representation of the learning problem (the target case). Providing concise descriptions of learner information based on input from the Manager Agent, the EA significantly contributes to the search for similar source cases, laying the groundwork for subsequent agents in the system.

In our architecture, two distinct types of elaborations are implemented:

**Initial Elaboration:** This process is initiated after the learner's initial request, using features stored in the Training-Request TR($L_a$) vector to reformulate a basic target case. It provides an initial proposition of a personalized learning process.

**Re-Elaboration:** This is Triggered when a learner encounters a problem or expresses dissatisfaction with the delivered learning sequence or abandons the current training. In this phase we consider learner's behavior and activities during the first recommendation to provide a more suitable learning path using the Learner-Traces LT($L_a$) vector.

**Retrieving Agent (RA),** The Retrieving Agent (RA), implemented as a Python FastAPI microservice, facilitates the retrieval of relevant past learning cases from a comprehensive case base. Leveraging ontologies, rule-based reasoning, and K-Nearest Neighbors (KNN) algorithms, the RA efficiently identifies and filters the most similar cases within the system, guiding the subsequent adaptation process for new learners.

**Case Retrieval for Initial Elaboration: Rule-Based Approach:**

During initial elaboration, the RA utilizes the rule-based approach to search for similar cases based on the Training-Request vector features. These rules follow an "IF-THEN" structure, where specific conditions based on the TR($L_a$) vector trigger actions that dictate the retrieval of appropriate cases from the existing pool.

The proposed algorithm given below in (*Algorithm 1*) differentiates between two **types** of features for similarity measurement: numerical and categorical features. Numerical features like prerequisite scores are compared using a normalized difference-based similarity *(Formula 1)*, emphasizing exact value matching. Categorical features like goals and competencies are evaluated using Jaccard similarity [20] *(Formula 2)*, which is robust to minor variations in wording.

$$\mathbf{Similarity}(\boldsymbol{L_a}, \boldsymbol{L_i}) = \frac{\mathbf{1}}{\mathbf{1} + \left|\boldsymbol{PT_{ScoreL_a}} - \boldsymbol{PT_{ScoreL_i}}\right|} \tag{1}$$

$$Fuzzy\_Similarity(L_a, L_i) = \frac{sizeof\ feature_a \cap feature_i}{sizeof\ feature_a \cup feature_i} \tag{2}$$

Journal of Theoretical and Applied Information Technology
15th June 2024. Vol.102. No. 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

**Dynamic Thresholding for Personalized Retrieval**

To ensure flexible and context-aware matching, we adopt a feature-specific method for threshold definition:

- G & C: Use Jaccard *similarity* with a threshold of 0.75, considering the potential for minor variations in wording for these string features.
- PT: Normalized difference-based similarity with a dynamically calculated threshold based on feature distribution characteristics (mean *(Formula 3)* and standard deviation *(Formula 4)*). This ensures stricter matching for higher scores and more flexibility for lower scores, tailoring retrieval to the specific value.

$$mean\_value = \sum_{i=0}^{n} \frac{PT\_Score_i}{n} \qquad (3)$$

$$std\_deviation = \sqrt{\frac{\sum_{i=0}^{n}\left(PT_{Score_i} - mean\_value\right)^2}{n}} \qquad (4)$$

With:

- i: Index of retrieved case.
- n: Number of retrieved cases.
- Threshold = mean_value ± std_deviation (adjusted based on target_value < mean_value)

- FSLM: Strict similarity (threshold of 1.0) is used since learning styles are typically more specific and less prone to variations.

```
Rule-Based Algorithm
Input
Target case: defined by TR(La) vector: (LP,G, C, PT, FSLM)
similarity_threshold ={"PT": t} /*define similarirty threshold for numerical features*/
p_cases: collection /*list of previous cases defined by TR(Li) vector*/

Begin
retrieved_cases: collection /*list to store retrieved cases*/

/*Fetch cases from the database*/
For (Case : p_cases) do
      match_score = 0;
      For feature in ["G", "C", "PT", "FSLM"] do
            if feature == similarity_threshold Then

                  /* Apply rules based on the TR(La) vector*/
                  if similarity (case[feature], target_case[feature]) >=
similarity_thresholds[feature] Then
                  match_score += 1;
                  Endif
            else
                  /*Apply fuzzy matching for text features*/
                  if feature in ["G", "C", "FSLM"]  Then
                        if fuzzy_matching(case[feature], target_case[feature]) >= 0.75:
                              match_score += 1;
                        Endif
                        else
                        if case[feature] == target_case[feature] Then
                              match_score += 1;
                        Endif
              Endif
            Endif
          EndFor
      if match_score >= 3 Then /*minimum score for retrieval*/
        retrieved_cases.append(case)
EndFor
/* Extract learning paths from the retrieved_cases.*/
Output
 return Learning_Path /*learning paths completed by successful learners*/
End
```

*Algorithm 1: Rule-Based Case Retrieval Algorithm*

**Case Retrieval for Re-Elaboration: KNN Approach**

In this phase we consider the behaviour of the learner considering his learning paths during the first proposition, This comparison is based on information that is collected from the observation and analysis of learning traces LT(La) vector to provide a tailored pedagogical training.

In our learning environment, each learner's educational journey consists of a sequence of learning objects, indexed by i (ranging from 0 to LOn, indicating the number of learning objects in each course). For each learning object, we record the time spent by the learner to achieve a learning object in the chosen course in addition to the number of visit of each one. This information is captured in the LT(Li) vector, defined as:

$$LT(Li)= \{CC, LO_1, LO_2, LO_3,\dots, LO_n, t_1, t_2, t_3,\dots, t_n, NVLO_1, NVLO_2, NVLO_3,\dots, NVLS_n, RT, RN\}$$

To retrieve similar cases to the current query case we employ K-Nearest Neighbors (k-NN) algorithm with the Euclidean similarity measure (*Algorithm 2*). This retrieves the k most similar cases from the stored data based on their learning traces. The optimal K value usually found is the square root of N [21], where N is the total number of successful learners.

To further personalize the recommendation, we create a frequency table that counts the occurrences of each unique learning path among the retrieved cases. Each row of the frequency table represents a unique learning path, and each column represents a learning object. By analyzing the frequency table, particularly focusing on paths with the highest counts, we identify common learning patterns exhibited by successful learners.

```
KNN Algorithm
Input
Target_case: LT(L,) vector defining the traces of the active learner
K: number of nearest neighbors.
p_cases: collection /*list of previous cases defined by TR(Li) vector*/

Begin
retrieved_cases: collection /*list to store retrieved cases*/
similarities /*list to store calculated similarities*/
sorted_cases /*list to store sorted_cases */

For (Case : p_cases) do
    /*Fetch cases from the database*/
        similarity= calculate_distance (Case, Target_case); /* calculate
similarity between the LT vector of the active learner and successful learners*/
        similarities.append(similarity);
EndFor
 /* Sort cases by distance (ascending) */
sorted_cases = sorted(similarities);

/*Select K nearest neighbors*/
  neighbors = sorted_cases [:K]

/* Extract learning paths from the selected nearest neighbors.*/
/* Create a frequency table to count the occurrences of each unique learning
path.*/
/* Identify common learning patterns by finding the most frequent paths in the
frequency table.*/
  common_path = find_most_frequent_path(neighbors)

Output
 return common_path /* List of common learning paths identified among the
retrieved cases*/

End
```

*Algorithm 2: KNN Algorithm for Case Retrieval*

**The Evaluation Agent (EvA),** The Evaluation Agent (EvA) in the CBR e-learning system is responsible for assessing learner performance and understanding, collecting learner feedback regarding the effectiveness of the learning experience, and triggering case retention or re-elaboration based on this feedback. Implemented as a Python FastAPI microservice, the EvA plays a crucial role in providing insights for case base updates and system improvement based on feedback vector features.

- **Collecting Learner Feedback:**
  The EvA collects feedback from learners regarding their learning experience using a choice-based questionnaire provided after the completion or suspension of a recommended course. This questionnaire assesses learner satisfaction and provides valuable insights into their experience.
- **Assessing Learner Performance:**
  After the completion of each learning object, learners are invited to take a post-training assessment to measure their knowledge or skills. This assessment helps evaluate learning gains and informs the adaptation of future learning recommendations.

The EvA interprets learner feedback to understand their judgment regarding the effectiveness of the learning experience. It categorizes feedback into positive or negative sentiments to decide whether to retain the learning process in the case base or initiate a new elaboration.

**The Retaining Agent (RetA)**, a Python FastAPI microservice, is responsible for storing new cases in MongoDB. This agent ensures the preservation of new learning cases and successful solutions in the case base for future use. Simultaneously, it updates learner profiles with acquired competencies and progress, thereby enriching the repository for future reference and enhancing the system's adaptability.

### 6.2 Knowledge Models:

To simplify the adaptation process and ensure the reuse and sharing of information across different e-learning systems, we propose to organize the system into various interoperable models adhering to diverse standards. Below, we present different models modeled in described in details in [22]:

**Learner Model,** also referred to as Learner Profile, furnishes a comprehensive description of the learner, encompassing goals, preferences, prerequisites, learning styles, competencies, and more. This model is crafted based on the LMS-LIP standard, ensuring a robust foundation for accurate representation and utilization of learner information. [23].

**Learning Object Model:** This model offers a comprehensive depiction of the pedagogical content delivered to the learner, encompassing tags, difficulty levels, prerequisites, and more. It plays a crucial role in enhancing the alignment between learning objects and the specific needs and preferences of learners. The modelling of Learning Resources was executed utilizing Learning Object Metadata (LOM) for precision and efficiency. [24].

**Competency Model:** This model defines the skills and relationships that learners can acquire

Journal of Theoretical and Applied Information Technology
15th June 2024. Vol.102. No. 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

through specific pedagogical training. It provides a comprehensive description of target competencies, enabling precise mapping of learning opportunities to desired outcomes. The model adheres to the IMS-RDCEO standard [25], ensuring interoperability and integration with other competency-based systems.

**Pedagogical Training Model:** This model offers a comprehensive overview of various learning object sequences. Its purpose is to generate a tailored pedagogical training program designed for specific goals or competencies in need of improvement. The model was structured utilizing IMS-Learning-Design [26] for enhanced effectiveness and adaptability.

## 7 TECHNICAL ARCHITECTURE:

In the pursuit of furnishing learners with effective pedagogical training for competency development within an adaptive e-learning system, we propose a technical architecture for an Adaptive Intelligent Educational Distributed Case-Based Reasoning (AIED-CBR) system presented in *Figure4*.
The technological foundation of our system is organized into Frontend, Backend, Agent Infrastructure, Database, and Ontologies

**Frontend**:

The student interface is implemented using ReactJS, to ensure dynamism and responsiveness, managing all learners' activities on the system: registration, login, profile management, course exploration and enrollment, personalized learning paths, content recommendations, interactive exercises, assessments, and adaptive feedback and progress tracking.

**Backend:**

A robust and scalable backend API, implemented with Python and FastAPI, efficiently oversees user data, course content, Case-Based Reasoning (CBR) processes, and coordination among agents for effective communication.

**Agent Infrastructure:**

Asynchronous communication and coordination among agents are facilitated by ACL language [27] allowing for parallel processing and enhancing system efficiency in responding to user interactions.

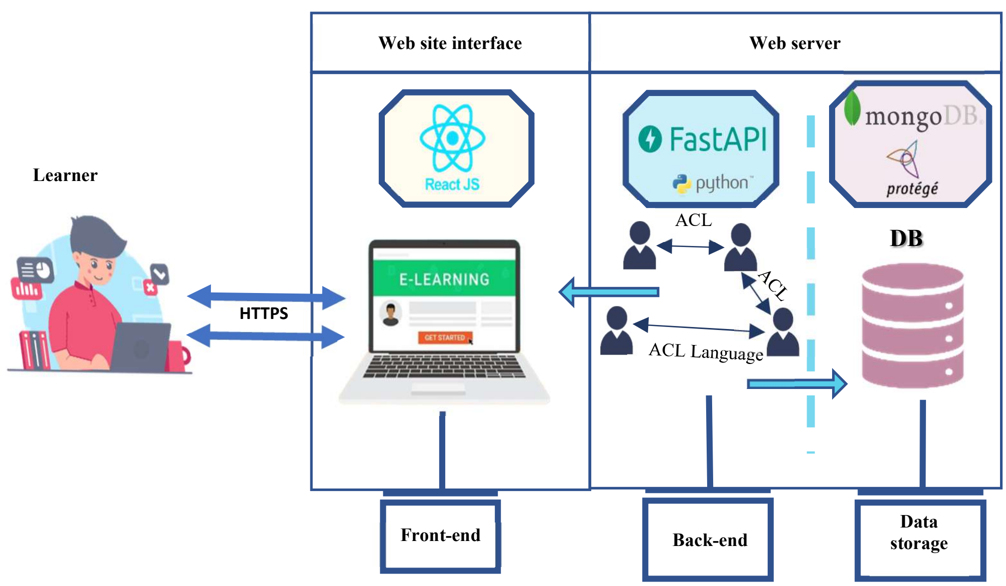


*Figure 2: The proposed Technical Architecture for the AIED-CBR system*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

**Database:**

MongoDB serves as the storage solution for diverse system data, including learner profiles containing competencies, preferences, and progress, course content with metadata and learning objects, a case base housing problem-solution pairs, and ontologies providing domain knowledge and competency descriptions.

**Ontologies:**

Formal representation of domain knowledge and competency structures is achieved through the use of OWL. This choice enables semantic reasoning and ensures interoperability with other systems. This comprehensive technology stack collectively contributes to the adaptability, scalability, and efficiency of our e-learning system, aligning with the demands of contemporary educational landscapes.

## 8. WORK REPORT

In this paper, we propose an enhancement of the foundational architecture of an adaptive e-learning distributed CBR system introduced in [19], transitioning towards a design for an adaptive e-learning Multi-Agent Hybrid CBR system. This enhancement is clearly illustrated in Table 1.

*Table 1: Ameliorations in the New Architecture*

| Adaptive E-Learning Distributed CBR System | Adaptive E-Learning Multi-Agents Hybrid CBR System |
|---|---|
| - Using CBR dynamic cycle.<br>- Integrate MAS with CBR to achieve the CBR cycle.<br>- Employ ontologies for knowledge description. | - Improve the multi-Agent CBR dynamic cycle.<br>- Propose a clear and detailed case representation using feature-vector representation to simplify the case retrieval.<br>- Utilizing a hybrid approach for case retrieval combining the strengths of Rule-based and KNN methods.<br>- Incorporate User's feedback for case evaluation<br>- Use of new technologies for architecture implementation. |

## 9. CONCLUSION:

This paper presents the architecture of an Adaptive Multi-Agent CBR E-learning system, employing feature-vector case representation to describe the learner profile. This representation covers all pertinent information necessary for relevant learning path recommendations, including personal information, learner requests, behavioral data, and learner feedback. The information is structured into four feature vectors, each tailored for specific phases of the CBR cycle. Additionally, we propose a hybrid approach to case retrieval within the multi-agent CBR cycle, integrating rule-based reasoning and the K-NN algorithm to enhance case retrieval efficiency and accuracy compared to traditional CBR methods. We also incorporate learner feedback into the evaluation phase of the CBR cycle to ensure ongoing improvement in system efficiency over time. The proposed system architecture and knowledge models offer a flexible and scalable framework for personalized learning experiences.

Looking ahead, future research directions may involve exploring advanced machine learning algorithms, such as deep learning and reinforcement learning, for further enhancing the intelligence and adaptability of hybrid CBR systems in e-learning environments. Additionally, investigating the impact of user feedback mechanisms and collaborative filtering techniques on recommendation quality and learner engagement could provide valuable insights for refining our approach. Furthermore, we look forward to testing the proposed technical architecture in order to evaluate its efficiency for the CBR cycle managing.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

**REFERENCES:**

[1] P. R. Varshavskii and A. P. Eremeev, 'Modeling of case-based reasoning in intelligent decision support systems', *Sci. Tech. Inf. Process.*, vol. 37, no. 5, pp. 336–345, Dec. 2010, doi: 10.3103/S0147688210050096.

[2] Z.-P. Fan, Y.-H. Li, X. Wang, and Y. Liu, 'Hybrid similarity measure for case retrieval in CBR and its application to emergency response towards gas explosion', *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2526–2534, 2014.

[3] P. Berka, 'Sentiment analysis using rule-based and case-based reasoning', *J. Intell. Inf. Syst.*, vol. 55, no. 1, pp. 51–66, Aug. 2020, doi: 10.1007/s10844-019-00591-8.

[4] 'A case-based reasoning recommender system for sustainable smart city development | AI & SOCIETY'. Accessed: Feb. 24, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s00146-020-00984-2

[5] J. L. Kolodner, 'An introduction to case-based reasoning', *Artif. Intell. Rev.*, vol. 6, no. 1, pp. 3–34, 1992, doi: 10.1007/BF00155578.

[6] A. Rahman and U. Budivanto, 'Case based reasoning adaptive e-learning system based on visual-auditory-kinesthetic learning styles', in *2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, IEEE, 2019, pp. 177–182. Accessed: Feb. 24, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8976921/

[7] F. Bousbahi and H. Chorfi, 'MOOC-Rec: a case based recommender system for MOOCs', *Procedia-Soc. Behav. Sci.*, vol. 195, pp. 1813–1822, 2015.

[8] K. M. El Ghouch Nihad and E.-N. El Mokhtar, 'Designing and modeling of a multi-agent adaptive learning system (MAALS) using incremental hybrid case-based reasoning (IHCBR)', *Int. J. Electr. Comput. Eng. IJECE*, vol. 10, no. 2, pp. 1980–1992, 2020.

[9] S. Tyagi and K. K. Bharadwaj, 'A hybrid recommender system using rule-based and case-based reasoning', *Int J Inf Electron Eng*, vol. 2, no. 4, pp. 586–590, 2012.

[10] S. S. Thygesen, D. Witschard, A. Kerren, T. B. Masood, and I. Hotz, 'Designing Feature Vector Representations: A case study from Chemistry', *ArXiv Prepr. ArXiv221203731*, 2022, Accessed: Mar. 02, 2024. [Online]. Available: https://arxiv.org/abs/2212.03731

[11] I. Jang, S. Kaur, and A. Yethiraj, 'Importance of feature construction in machine learning for phase transitions', *J. Chem. Phys.*, vol. 157, no. 9, 2022, Accessed: Mar. 02, 2024. [Online]. Available: https://pubs.aip.org/aip/jcp/article/157/9/094904/2841886

[12] C. J. Villagrá-Arnedo, F. J. Gallego-Durán, F. Llorens-Largo, P. Compañ-Rosique, R. Satorre-Cuerda, and R. Molina-Carmona, 'Improving the expressiveness of black-box models for predicting student performance', *Comput. Hum. Behav.*, vol. 72, pp. 621–631, 2017.

[13] C.-J. Villagrá-Arnedo, F. J. Gallego-Durán, P. Compañ, F. Llorens Largo, and R. Molina-Carmona, 'Predicting academic performance from behavioural and learning data', 2016, Accessed: Mar. 02, 2024. [Online]. Available: https://rua.ua.es/dspace/handle/10045/57216

[14] A. Real-Fernández, R. Molina-Carmona, M. L. Pertegal-Felices, and F. Llorens-Largo, 'Definition of a Feature Vector to Characterise Learners in Adaptive Learning Systems', in *Research & Innovation Forum 2019*, A. Visvizi and M. D. Lytras, Eds., in Springer Proceedings in Complexity. , Cham: Springer International Publishing, 2019, pp. 75–89. doi: 10.1007/978-3-030-30809-4_8.

[15] N. Masri *et al.*, 'Survey of Rule-Based Systems', vol. 3, no. 7, 2019.

[16] T. Pinto, R. Faia, M. Navarro-Caceres, G. Santos, J. M. Corchado, and Z. Vale, 'Multi-agent-based CBR recommender system for intelligent energy management in buildings', *IEEE Syst. J.*, vol. 13, no. 1, pp. 1084–1095, 2018.

[17] I. G. T. Mahardikaa and I. W. Suprianaa, 'A Case Based Reasoning System For Recommendation Of Restaurant In Jimbaran Using K-Nearest Neighbor', *J. Elektron. Ilmu Komput. Udayana P-ISSN*, vol. 2301, p. 5373, 2020.

[18] V. Athitsos and S. Sclaroff, 'Boosting nearest neighbor classifiers for multiclass recognition', in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, IEEE, 2005, pp. 45–45. Accessed: Mar. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1565346/

[19] S. Abroun, M. Ghailani, and A. Fennan, 'Modeling and design of an architecture for Adaptive Intelligent Educational Distributed CBR system.', *Int. J. Inf. Sci. Technol.*, vol. 3, no. 2, pp. 29–38, 2019.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195


[20] L. Amgoud and V. David, 'Similarity Measures Based on Compiled Arguments', in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, vol. 12897, J. Vejnarová and N. Wilson, Eds., in Lecture Notes in Computer Science, vol. 12897. , Cham: Springer International Publishing, 2021, pp. 32–44. doi: 10.1007/978-3-030-86772-0_3.

[21] M. Maleki, N. Manshouri, and T. Kayıkçıoğlu, 'A novel simple method to select optimal k in k-nearest neighbor classifier', *Int. J. Comput. Sci. Inf. Secur. IJCSIS*, vol. 15, no. 2, 2017, Accessed: Feb. 24, 2024. [Online]. Available: https://www.researchgate.net/profile/Negin-Melek/publication/337474083_A_Novel_Simple_Method_to_Select_Optimal_k_in_k-Nearest_Neighbor_Classifier/links/5dda2ca092851c1fedacab33/A-Novel-Simple-Method-to-Select-Optimal-k-in-k-Nearest-Neighbor-Classifier.pdf

[22] S. Abroun and M. Ghailalni, 'Modeling an Adaptive E-learning System skills oriented based on Multi-agents reasoning approach for dynamic cases', Accessed: Feb. 24, 2024. [Online]. Available: https://www.academia.edu/download/56516845/Modeling_an_Adaptive_E-learning_System_skills_oriented.pdf

[23] L. Oubahssi and M. Grandbastien, 'Un modèle pour améliorer l'interopérabilité des données pédagogiques apprenant', in *3ième conférence en Environnement Informatique pour l'Apprentissage Humain-EIAH 2007*, INRP, 2007, pp. 437–448.

[24] J.-P. Pernin, 'LOM, SCORM et IMS-Learning Design: ressources, activités et scénarios', in *actes du colloque «L'indexation des ressources pédagogiques numériques», Lyon*, 2004.

[25] M. El Asame and M. Wakrim, 'Towards a competency model: A review of the literature and the competency standards', *Educ. Inf. Technol.*, vol. 23, no. 1, pp. 225–236, Jan. 2018, doi: 10.1007/s10639-017-9596-z.

[26] J.-P. Pernin, 'LOM, SCORM et IMS-Learning Design: ressources, activités et scénarios', in *actes du colloque «L'indexation des ressources pédagogiques numériques», Lyon*, 2004. Accessed: Feb. 24, 2024. [Online]. Available: https://core.ac.uk/download/pdf/12465572.pdf

[27] M. T. Kone, A. Shimazu, and T. Nakajima, 'The State of the Art in Agent Communication Languages', *Knowl. Inf. Syst.*, vol. 2, no. 3, pp. 259–284, Aug. 2000, doi: 10.1007/PL00013712.