A Feature-wise Comparative Assessment of the CBR-based Methodologies FLEA and SEASALT

Viktor Eisenstadt[1,2,*], Jessica Bielski[3], Christoph Langenhan[3], Klaus-Dieter Althoff[1,2] and Andreas Dengel[1]

[1] *DFKI (German Research Center for Artificial Intelligence), Germany*
[2] *University of Hildesheim, Germany*
[3] *Technical University of Munich, Germany*

Abstract

This paper presents a feature-wise comparison of the capabilities of two methodologies developed for design and implementation of distributed artificial intelligence (AI) systems that utilize case-based reasoning (CBR) as the main computational method. The evaluated methodologies are SEASALT (Sharing Experience using an Agent-based System Architecture Layout), an established approach for organization of multi-agent CBR in several coherent system layers, and FLEA (Find, Learn, Explain, Adapt), a novel AI-based methodology for combining CBR and deep learning (DL) in flexible dependency graphs. SEASALT was conceptualized for use in different application domains at its outset, while FLEA was developed for the DL-based support of the architectural design process and then generalized for other domains.

The comparative assessment aims to investigate the structural, relational, and technological capabilities of the methodologies, seeking for shallow as well as deeply hidden differences. As an overall conclusion, the evaluation has revealed that SEASALT provides a more restrictive feature set that enables for more stability for the systems that utilize this methodology, while FLEA leaves more room for selection of applicable computational methods and offers an integrated support for explanation of decisions made by the system. Furthermore, it has been shown that a SEASALT-based approach can be turned into a FLEA-based process, without loss of functionality. The work presented in this paper is part of a PhD thesis written in the context of the research project Metis-II funded by the German Research Foundation (DFG).

Keywords

artificial intelligence, methodology, case-based reasoning, deep learning, multi-agent-systems

1. Introduction

In modern research and industry, many complex systems that utilize artificial intelligence (AI) as the main computational technique make use of a specific methodology to provide the underlying approach with a coherent and comprehensible structure. Such methodologies usually help to organize the system by utilizing specific components and interconnections that control the execution processes and data flow cycles within the system.

LWDA 2023 Workshops: BIA, DB, IR, KDML and WM. Marburg, Germany, 09.-11. October 2023

*Corresponding author.

EMAIL: viktor.eisenstadt@dfki.de (V. Eisenstadt); jessica.bielski@tum.de (J. Bielski); langenhan@tum.de (C. Langenhan); klaus-dieter.althoff@dfki.de (K. Althoff); andreas.dengel@dfki.de (A. Dengel)

In the previous research, different methodologies were proposed to accomplish these tasks. In the field of case-based reasoning (CBR) [1], AI-based research and development discipline for making informed decisions based on experiences recorded in the past, such methodologies are common and have been applied for different domains of engineering, science, or creativity in order to structurally organize the corresponding systems.

In this paper, an established and frequently used CBR-based methodology SEASALT (Sharing Experience using an Agent-based System Architecture Layout) [2, 3] will be evaluated against the new methodology FLEA (Find, Learn, Explain, Adapt) [4] in order to explore the capabilities of both approaches and investigate their corresponding advantages. The main goal of research presented in this paper is to find out if certain application cases exist in which one of the methodologies might be a superior choice, how to recognize these cases, and what are the boundaries of both approaches. First, both methodologies will be briefly described, after that their features will be compared using a list of criteria. In conclusion, the findings of the comparison will be presented.

A related research work [5] was published in 2023 by Schultheis et al., it contains a feature-wise comparison of the currently existing CBR programming frameworks, such as *myCBR*[1], *CloodCBR*[2], or *jColibri*[3]. We consider the methodology comparison presented in this paper a methodical supplementation of work [5]. Comparing CBR methodologies provides a more complete picture on the currently available CBR development tools.

2. Methodology SEASALT

The methodology SEASALT (see Fig. 1) is described by its authors as *'an application-independent architecture that features knowledge acquisition from a web-community, knowledge modularisation and agent-based knowledge maintenance'* [3]. It is based on the approach CoMES (Collaborative Multi-Expert-Systems) [6] and can be considered a universal distributed CBR methodology that can be utilized for any application domain, without further specification. The distributed structure of SEASALT allows for processing various application domain knowledge, consisting of five main components: *Knowledge Sources* (case bases or expert knowledge collected from the internet), *Knowledge Formalisation* (conversion of unstructured knowledge into the structured form), *Knowledge Representation* (structured knowledge formats for processing by the knowledge containers of CBR), *Knowledge Provision* (segmentation into discerning domain topics using, e.g., case factories [7] and knowledge lines [8]), and *Individualised Knowledge* (user interfaces).

SEASALT can be considered one of the most frequently used methodologies in the research field of distributed case-based reasoning. In the previously published research, it has been shown that SEASALT can be utilized for different application domains, such as travel medicine (research project *docQuery* [3, 9]), cyber security (network intrusion detection system [10]), big data (adds new main component *Knowledge Stream Management* [11]), or decision support for maintenance and diagnosis of aircraft [12].

[1] http://mycbr-project.org/ – Last access on 14th Sep. 2023
[2] https://cloodcbr.com/ – Last access on 14th Sep. 2023
[3] https://gaia.fdi.ucm.es/research/colibri/jcolibri/ – Last access on 14th Sep. 2023

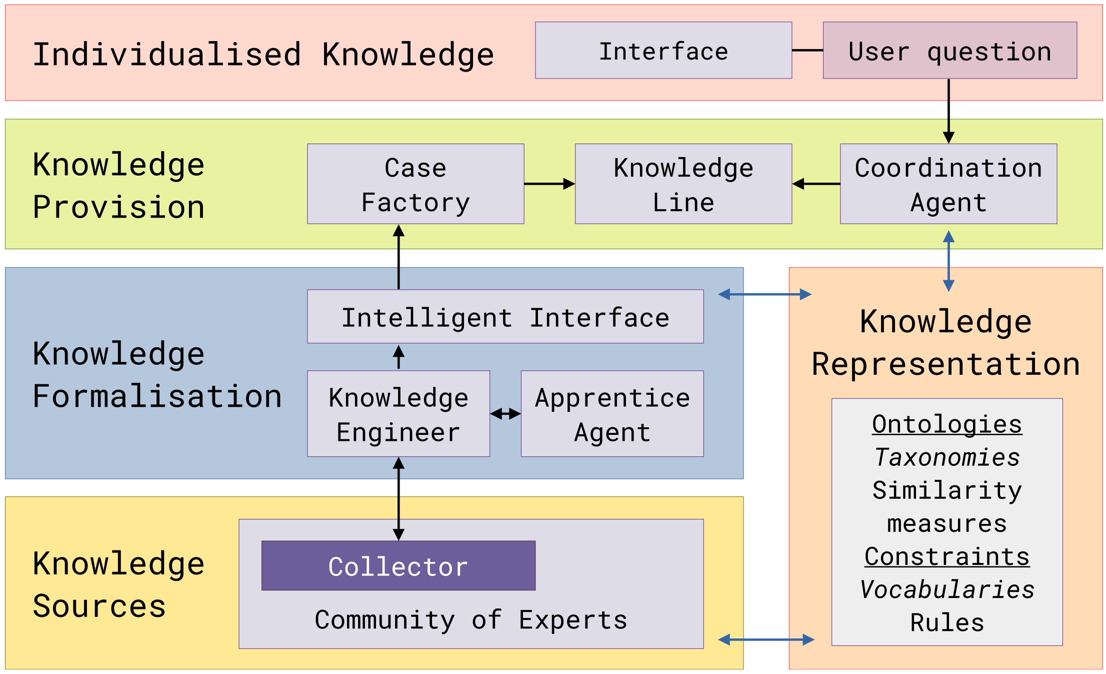

Figure 1: The components of the SEASALT methodology.

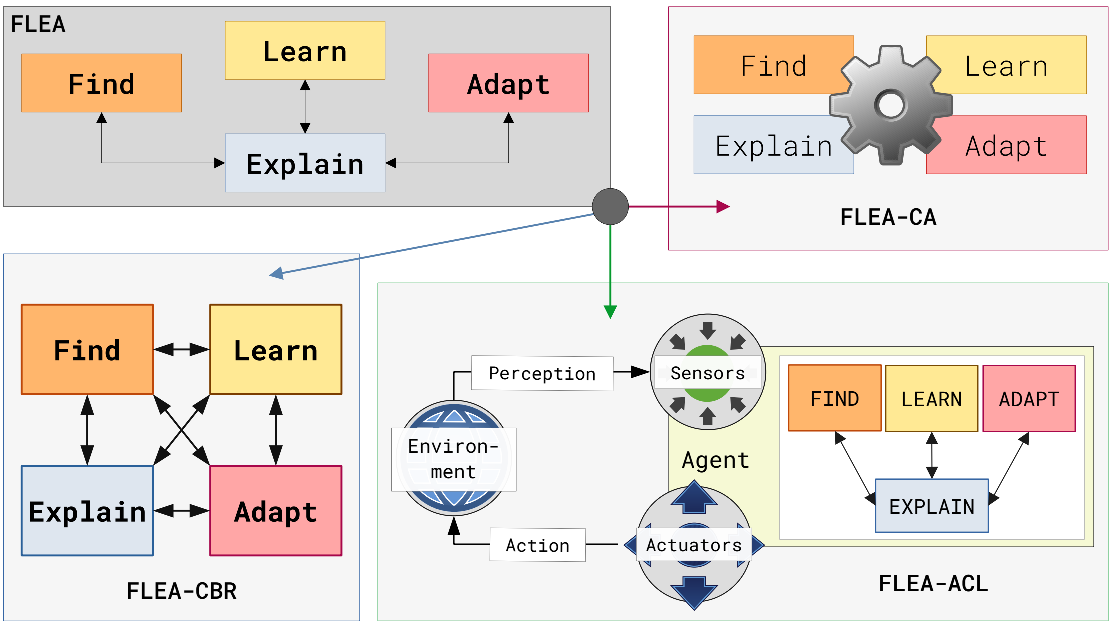

Figure 2: Methodology FLEA with its derivatives FLEA-CBR, FLEA-ACL, and FLEA-CA.

3. Methodology FLEA

The methodology FLEA (see Fig. 2) was originally developed to support the early phases of the architectural design process with modern AI methods, such as CBR or deep learning (DL). Using a dependency graph model that distributes design support tasks, such as retrieval of similar floor plans *(Find)* or autocompletion of design steps *(Learn)*, among the components of the execution process, different design scenarios can be supported (see published examples [4]). FLEA's combination of CBR and DL methods helps to accelerate architectural design process in its ideation phases, making it more efficient and sustainable. Additionally, FLEA was generalized for use in domains other than architecture via its derivatives FLEA-CBR, FLE-ACL, and FLEA-CA.

FLEA-CBR [13] can be considered a *do-it-yourself* CBR execution cycle and an alternative to the 4R cycle (Retrieve, Reuse, Revise, Retain) [14] for the domains that are not able to use 4R. FLEA-CBR allows for flexibility of the execution order using extended capabilities for mixing, sequencing, or repeating of CBR phases. FLEA-ACL [15] is an improved agents communication language (ACL) for *case-based agents* in multi-agent systems, replacing the established language FIPA-ACL for such agents by bringing more precision and sparsity to their communication processes. Finally, FLEA for *cognitive architectures* (CA) aims to provide an approach for an autonomous early design agent that elaborates the most optimal design solution by learning and competing with the user. FLEA-CA is under active development, it aims to provide a CBR-based CA for architectural design, being related to other CAs, such as *ACT-R* [16] or *SOAR* [17].

4. Assessment of Features

In this section, the feature-wise comparative assessment of the methodologies FLEA and SEASALT will be presented. A specific list of criteria was developed to evaluate the capabilities of methodologies. The list is based on the foundations and goals of the project Metis-II, such as CBR, *multi-agent systems* (MAS), or integration of explainability of AI methods. For each feature, it will be elaborated how it is supported by the corresponding methodology, if the support is sufficient and what can be improved to increase it.

4.1. General Type

To start the assessment, it should be investigated if both methodologies can be considered a general type, i.e., if they are *universal* for use in multiple domains. This was already mentioned in this paper for SEASALT (see Sect. 2 with references to examples) and the authors of SEASALT denote the methodology as independent of domain or application [2, 3]. The original FLEA was developed to support the design process in the domain of architecture, via FLEA-CBR, FLEA-ACL, and FLEA-CA (see Sect. 3) it was generalized for use in other domains and applications. For FLEA-CBR, a theoretical example for use for *library services optimization* exists [13]. It can be concluded that SEASALT can be currently considered a more established and superior methodology for this criterion. FLEA and its derivatives require several different application domains to catch up.

4.2. Integration of Explainable AI

In order to make the decisions of the AI-based systems transparent for the user, the paradigm of *XAI* (eXplainable AI) was initiated and gained a great grade of popularity among the AI researchers and developers, being present in the contributions of the major AI conferences and in specific monograph publications. Following this trend, FLEA and its derivatives integrate the XAI features 'out-of-the-box' in the component *Explain*, it is up to the developers of the implementation to decide which XAI solutions (for example, *explanation patterns* [18]), or frameworks (for example, the XAI framework by Wang et al. [19]) will be utilized (see also Sect. 4.5). SEASALT does not provide a direct integration of an XAI component, however, similar to the addition of the new *Knowledge Stream Management* [11] component, SEASALT can be extended with an XAI facility. FLEA can be seen as a more advanced methodology regarding XAI, i.e., it can be currently considered a superior choice if explainability of system's processes is required.

4.3. CBR Knowledge Containers

The *knowledge containers* [20] of CBR consist of *vocabulary* (knowledge representation type), *similarity measure* (a function to calculate similarity between cases), *case base* (collection of previously recorded experiences for reasoning), and *solution transformation* (a.k.a. *adaptation knowledge*, stands for methods for adaptation of the solutions from similar cases to the current problem, e.g., using rules). Knowledge containers play a crucial role for CBR, they provide essential foundations for its computational processes. Both FLEA and SEASALT provide a full support for CBR knowledge containers, being both based on CBR foundations. SEASALT covers the containers by implementing them in its main component *Knowledge Representation* (but also *Knowledge Sources*), while FLEA implements them in multiple facilities of its respective components, e.g., case bases and similarity measures in *Find* and *Learn* or transformation rules in *Adapt*. Both methodologies can be considered equal regarding the support of knowledge containers, their implementations should keep up with the current advances of the containers.

4.4. Special Features for Agents

Being conceptualized as distributed methodologies at the outset, both SEASALT and FLEA can be used to form a multi-agent system, offering the capability of task-sharing between and within their corresponding components, which communicate with each other in order to achieve their own local as well as the common global goals. While the features of communication and collaboration are typical for every multi-agent system, SEASALT and FLEA provide several specific additional MAS features. In order to enhance the coordination process within SEASALT-based systems, the methodology utilizes the previously mentioned concept of *case factories* [7], while FLEA, via FLEA-ACL, offers an improvement of the communication process between case-based agents. Both methodologies can be considered equal regarding special agent-related features.

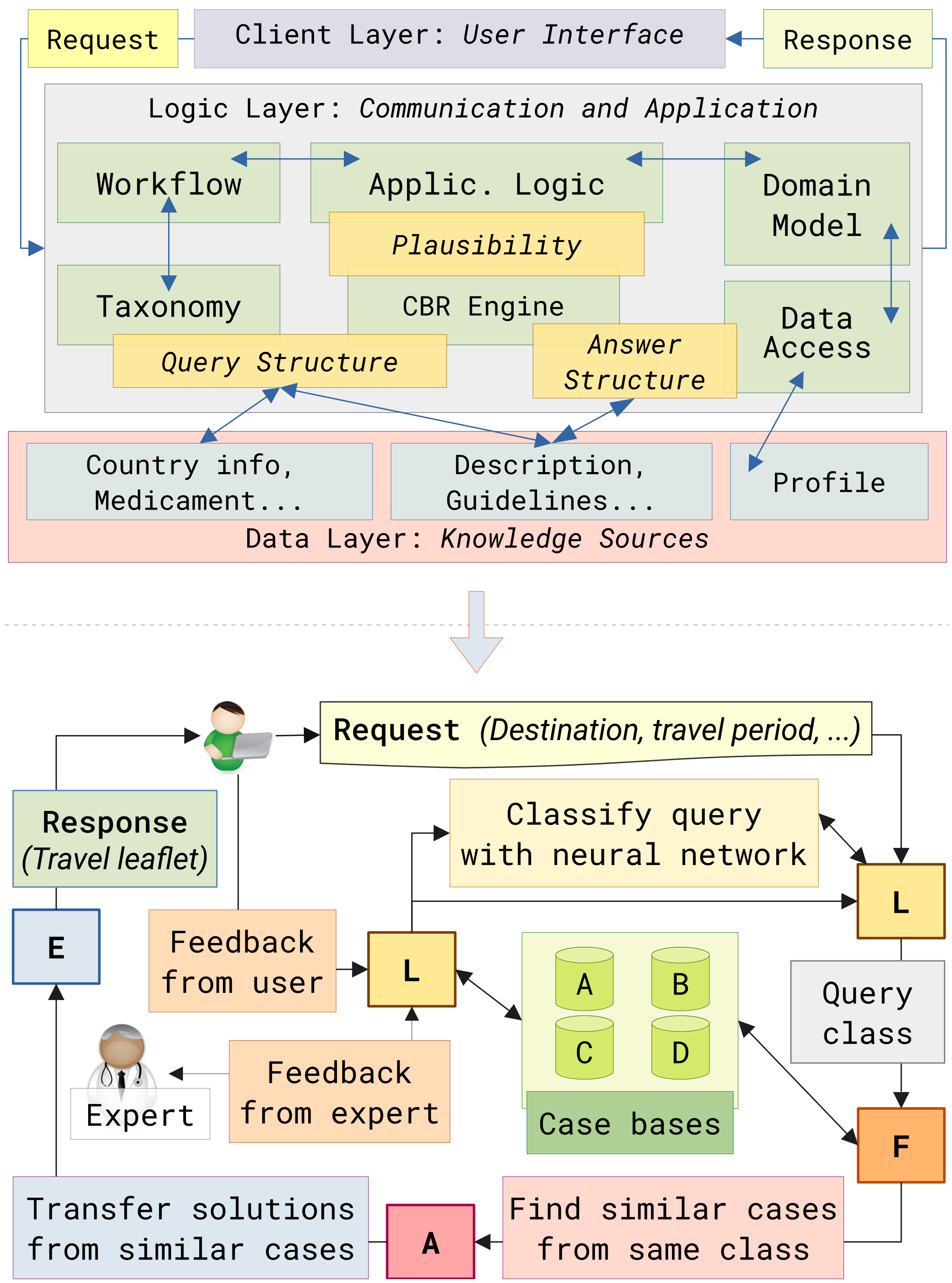

Figure 3: Implementation in SEASALT (top) and dependency graph of FLEA (bottom) for *docQuery*.

4.5. Restrictiveness of Methods

Finally, both methodologies can be compared by the freedom or limit of method selection. While FLEA and its derivatives provide no specifications on algorithms and data structures for their corresponding implementations, SEASALT goes more in detail, describing which methods and data representations can be applied. That is, FLEA leaves the choice of methods to the researchers and developers involved in the implementation process, defining only the overall abstract structure and general tasks of the components *Find*, *Learn*, *Explain*, and *Adapt*. In contrast, SEASALT specifies the agent types (e.g., Coordinator), approaches (e.g., knowledge lines or case factories), or data representations (taxonomies, ontologies, vocabularies) – the persons involved can make more concrete choices.

5. Conclusion and Outlook

To conclude the research presented in this paper and evaluate the methodology comparison in a simple manner, it can be first summarized that both investigated CBR methodologies are similar on the high level of abstraction: both FLEA and SEASALT make use of the same foundations and are generally comparable in terms of the overall structural relationships of their respective components. This can be further demonstrated by a simple example in Fig. 3, where the first existing implementation of SEASALT for the already mentioned project *docQuery* [3, 9] (distributed CBR-based support for travel medicine information) is converted into a FLEA dependency graph. The FLEA graph executes the same tasks as the original SEASALT implementation, adding new modern AI methods to the process, such as artificial neural networks (ANN) for query classification.

Overall, it can be concluded that the SEASALT methodology provides a more stable structure with established CBR and MAS methods, while FLEA and its derivatives offer integration of the latest AI advances (XAI or the current ANN models). With more FLEA domains, both approaches can be compared more comprehensively in the future.

References

[1] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann, 1993.

[2] K. Bach, M. Reichle, K.-D. Althoff, A Domain Independent System Architecture for Sharing Experience, in: Lernen, Wissen, Adaption. Workshop Proceedings, 2007, pp. 296–303.

[3] M. Reichle, K. Bach, K.-D. Althoff, Knowledge engineering within the application-independent architecture SEASALT, International Journal of Knowledge Engineering and Data Mining 1 (2011) 202–215. doi:10.1504/IJKEDM.2011.037643.

[4] V. Eisenstadt, C. Langenhan, K.-D. Althoff, Supporting Architectural Design Process with FLEA. A Distributed AI Methodology for Retrieval, Suggestion, Adaptation, and Explanation of Room Configurations, in: J.-H. Lee (Ed.), Computer-Aided Architectural Design. "Hello, Culture", 2019, pp. 58–73. doi:10.1007/978-981-13-8410-3_5.

[5] A. Schultheis, C. Zeyen, R. Bergmann, An Overview and Comparison of Case-Based Reasoning Frameworks, in: Conference Proceedings of ICCBR 2023, 2023.
[6] K.-D. Althoff, K. Bach, J.-O. Deutsch, A. Hanft, J. Mänz, T. Müller, R. Newo, M. Reichle, M. Schaaf, K.-H. Weis, Collaborative Multi-Expert-Systems – Realizing Knowlegde-Product-Lines with Case Factories and Distributed Learning Systems, in: 3rd KESE @ KI, 2007.
[7] K.-D. Althoff, A. Hanft, M. Schaaf, Case Factory – Maintaining Experience to Learn, in: T. R. Roth-Berghofer, M. H. Göker, H. A. Güvenir (Eds.), Advances in Case-Based Reasoning, 2006, pp. 429–442. doi:10.1007/11805816_32.
[8] K. Bach, M. Reichle, A. Reichle-Schmehl, K.-D. Althoff, Implementing a coordination agent for modularised case bases, in: 13th UKCBR Workshop Procs., 2008, pp. 1–12.
[9] M. Reichle, K. Bach, K.-D. Althoff, The SEASALT Architecture and Its Realization within the docQuery Project, in: B. Mertsching, M. Hund, Z. Aziz (Eds.), KI 2009: Advances in Artificial Intelligence, 2009, pp. 556–563. doi:10.1007/978-3-642-04617-9_70.
[10] J. M. Schoenborn, K.-D. Althoff, Multi-Agent Case-Based Reasoning: a Network Intrusion Detection System, in: P. Reuss, V. Eisenstadt, J. M. Schoenborn, J. Schäfer (Eds.), Proceedings of the LWDA 2022 Workshops: FGWM, FGKD, and FGDB, 2022, pp. 258–269.
[11] K. Amin, Building an Integrated CBR-Big Data Oriented Architecture for Case-Based Reasoning Systems, in: ICCBR (Workshops), 2017, pp. 189–193.
[12] P. Reuss, K.-D. Althoff, W. Henkel, Case-Based Decision Support on Diagnosis and Maintenance in the Aircraft Domain, in: LWDA, 2016, pp. 249–256.
[13] V. Eisenstadt, C. Langenhan, K.-D. Althoff, FLEA-CBR – A Flexible Alternative to the Classic 4R Cycle of Case-Based Reasoning, in: K. Bach, C. Marling (Eds.), Case-Based Reasoning Research and Development, 2019, pp. 49–63. doi:10.1007/978-3-030-29249-2_4.
[14] A. Aamodt, E. Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI comm. 7 (1994) 39–59. doi:10.3233/AIC-1994-7104.
[15] V. Eisenstadt, K.-D. Althoff, RALE-ACL—A Language for Information Exchange between Case-Based Agents as Alternative to the FIPA-ACL-Based Communication, in: R. Barták, E. Bell (Eds.), Proceedings of the 33rd International FLAIRS Conference, 2020, pp. 185–188.
[16] F. E. Ritter, F. Tehranchi, J. D. Oury, ACT-R: A cognitive architecture for modeling cognition, WIREs Cognitive Science 10 (2019) e1488. doi:10.1002/wcs.1488.
[17] J. E. Laird, The Soar Cognitive Architecture, reprint ed. ed., MIT Press, 2019.
[18] J. Cassens, A. Kofod-Petersen, Designing Explanation Aware Systems: The Quest for Explanation Patterns, in: Explanation-Aware Computing (ExaCt), 2007, pp. 20–27.
[19] D. Wang, Q. Yang, A. Abdul, B. Y. Lim, Designing Theory-Driven User-Centric Explainable AI, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, p. 1–15. doi:10.1145/3290605.3300831.
[20] M. Richter, Knowledge containers, Readings in case-based reasoning (2003).