An Overview and Comparison of Case-Based Reasoning Frameworks

Alexander Schultheis[1], Christian Zeyen[1], and Ralph Bergmann[1,2]

[1] German Research Center for Artificial Intelligence (DFKI)
Branch University of Trier, Behringstraße 21, 54296 Trier, Germany
[2] Business Information Systems II, University of Trier, 54286 Trier, Germany
http://www.wi2.uni-trier.de
{alexander.schultheis,christian.zeyen,ralph.bergmann}@dfki.de

Abstract *Case-Based Reasoning* (CBR) is a methodology with many applications in industrial and scientific domains. Over the past decades, various frameworks have been developed to facilitate the development of CBR applications. For practitioners and researchers, it is challenging to overview the landscape of existing frameworks with their specific scope and features. This makes it difficult to choose the most suitable framework for specific requirements. To address this issue, this work provides an overview and comparison of CBR frameworks, focusing on five recent, open-source CBR frameworks: CloodCBR, eXiT*CBR, jColibri, myCBR, and ProCAKE. They are compared by supported CBR types, knowledge containers, CBR phases, interfaces, and special features.

Keywords: Case-Based Reasoning · CBR Framework · CBR Applications · CloodCBR · eXiT*CBR · jColibri · myCBR · ProCAKE

1 Introduction

Case-Based Reasoning (CBR) applications have been developed for over 30 years. The first systems implementing parts of the CBR methodology date back to the 80s [26, 28, 60]. Up to now, numerous CBR frameworks have been developed to ease the creation of CBR systems. In this work, a framework is regarded as a generic, domain-independent, and extensible software component that enables the implementation of specific applications. In particular, some non-commercial frameworks that are publicly available under an open-source license became widely used within the CBR research community. In the past, some publications addressed the comparison of CBR frameworks [3, 16, 18, 57, 60]. However, no publication covers a broad range of the most recent frameworks.

This work aims to overview all open-source CBR frameworks still under active development or used for current research, to help find suitable frameworks for developing specific CBR applications. This paper mainly describes and compares five recent open-source CBR frameworks: CloodCBR, eXiT*CBR, jColibri, myCBR, and ProCAKE. In particular, the supported CBR types and case representations, the framework's capabilities for implementing the four knowledge containers [48], the supported CBR phases [1], interfaces, and special features will be considered.

The remainder of this paper is organized as follows: Section 2 gives some background on the results of previous CBR framework overviews. Then, Section 3 provides a general overview of CBR frameworks found in the literature, including commercial frameworks. A more in-depth description of the five selected frameworks is presented in Section 4. On this basis, a comparison is made in Section 5. Finally, Section 6 summarizes the main findings of this work.

2 Background

In 1994, Watson and Marier [60] present and compare the three CBR frameworks Isoft ReCALL, Cognitive Systems Inc. ReMind, and AcknoSoft KATE, among other CBR systems, including applications of these. They show the early evolution of CBR although they do not explicitly use the term framework.

Atanassov and Antonov [3] describe the two non-commercial frameworks, jColibri and myCBR. The authors highlight the GUI and modification of weights in myCBR and the supported database interfaces of both frameworks.

ElKafrawy and Mohamed [16] compare a selection of CBR frameworks (CBR-Shell, FreeCBR, jColibri, myCBR, and eXiT*CBR) based on several criteria. They introduce and compare the frameworks regarding case selection strategies, retrieval, revision, storage, speed, and handling of missing or noisy data. In addition, they compare the retrieval phase in their evaluation. As a result, they find that freeCBR and eXiT*CBR are easy to use, especially because of their GUI. In contrast, myCBR and jColibri are suitable for more complex applications, whereby the interface as well as the set of functions of myCBR is rated higher in a direct comparison.

Thakur et al. [57] investigate the context in which CBR systems should be created and the tools that can be used for this purpose. Therefore, the frameworks AIAI CBR Shell, myCBR, and jColibri are presented and compared in the context of comparative analysis. Factors such as case structures, CBR phases, GUIs, and handling of uncertain data are considered.

In 2016, He and Wang [18] present a review of CBR shells and frameworks. They define a shell as an application generator to build specific applications quickly using a GUI. A CBR framework is defined as software designed for extensibility. The authors do not explain whether the shells can also be used as frameworks. He and Wang consider Caspian CBR Shell, CBR-Express, Cognitive Systems Inc. ReMind, CBRWorks, AIAI CBR Shell as shells and jColibri, myCBR, IUCBRF, Empolis Orenge, and freeCBR as frameworks and provide an overview of these.

Peixoto, Martinazzo, and Weber [42] survey cyberinfrastructure requirements for researchers as a target audience to reduce the entry barriers for using such infrastructures. The CBR community is used as an example, where an external inventory of CBR tools is conducted. CAKE, AIAI CBR Shell, CBR Works, Colibri Studio, eXiT*CBR, freeCBR, and myCBR are comparatively described as examples based on operating systems, required programming experience, ease of use, tutorials, and free availability.

Several publications addressed an overview and comparison of CBR frameworks in the past. However, there is no recent and complete survey of the latest frameworks. This work tries to close this gap and focuses on providing decision support for practitioners and researchers to find suitable frameworks for developing CBR applications.

3 General Overview of CBR Frameworks

To identify CBR frameworks, an extensive literature search was conducted on the following citation databases and search engines: Google Scholar, Semantic Scholar, BASE (Bielefeld Academic Search Engine), IEEE Xplore, DBLP (Digital Bibliography & Library Project), and CiteSeerX. We used full-text searches except for DBLP, which only supports a search of title and abstract.

We collected and reviewed publications that address CBR frameworks, systems, applications, and the comparison. We examined the literature and associated online resources to distinguish frameworks for developing arbitrary CBR applications from specific CBR applications. Systems and specific applications such as CasePoint [60], CBR-Express [60], CLAVIER [20,35], Compaq SMART [7,38], Eclipse – The Easy-Reasoner [60], ESTEEM — Case-Based Reasoning Shell [60] and CasePower [60] are not further considered for that reason. As a result, eleven open-source frameworks and five closed-source/commercial frameworks are selected. Table 1 shows the numbers of related publications for the respective framework, the period of the publications, and the year the source code was updated the last time. Due to many irrelevant results, we do not show the number of publications on CiteSeerX.

Some open-source frameworks are no longer under development and are thus not further considered in this paper: AIAI CBR Shell[1], CASPIAN CBR Shell[2], CAT-CBR [2], freeCBR[3], INRIA CBR*Tools[4] [24, 25], and IUCBRF[5] [11, 12]. However, the source code is still available (publicly or upon request) for some of the frameworks.

Over the years, several commercial frameworks have been developed, some of which have emerged from the research. However, these frameworks are not further considered in this overview due to the proprietary code: Brightware ART*Enterprise [60], AcknoSoft KATE [59], CBR Works [53] (emerged from research at the University of Kaiserslautern and parts of it have been adopted in the open-source framework myCBR [55]), Empolis Orenge [54] (emerged from

[1] http://www.aiai.ed.ac.uk/project/cbr/CBRDistrib/
[2] https://www.aber.ac.uk/~dcswww/Research/mbsg/cbrprojects/getting_caspian.shtml
[3] http://freecbr.sourceforge.net/
[4] http://www-sop.inria.fr/aid/software.html
[5] https://homes.luddy.indiana.edu/leake/iucbrf/

Tab. 1. Search Results on Publications Related to the Respective CBR Frameworks, Ordered by Code Accessibility and Search Results (requested on 04/18/2023).

CBR Framework	Google Scholar	Semantic Scholar	BASE	IEEE Xplore	DBLP	Publication Period	Last Code Update
jColibri	1,030	89	63	21	11	Since 2004	Jan. 2019
myCBR	626	49	42	7	10	Since 2007	Dec. 2022 / May 2019[6]
eXiT*CBR	105	4	35	0	7	Since 2008	Jan. 2017
IUCBRF	101	4	0	0	1	2001 - 2005	Dec. 2013
freeCBR	78	6	4	1	0	2005 - 2018	Mar. 2013
CAT-CBR	68	8	1	0	0	Since 2002	-
ProCAKE	41	4	16	0	1	Since 2005[7]	Apr. 2023
AIAI CBR Shell	17	16	0	0	0	1990 - 2019	Sep. 2010
Caspian CBR Shell	6	3	0	0	0	1990 - 2019	Jan. 1997
INRIA CBR*Tools	2	1	1	0	0	Since 1991	Sep. 2001
CloodCBR	6	0	0	0	0	Since 2020	Mar. 2023
Brightware ART*Enterprise (Closed Source)	306	1	0	0	0	Since 1997	-
AcknoSoft KATE (Closed Source)	177	122	1	0	0	1993-2021	-
CBR Works (Closed Source)	168[8]	22	10	0	1	1998-2017	-
Cognitive Systems ReMind (Closed Source)	124	1	1	0	0	Since 1997	-
Isoft ReCALL (Closed Source)	104	103	0	0	0	1994-2020	-
Empolis Orenge / Empolis Information Access Suite (Closed Source)	42 / 31	2 / 1	1 / 0	0 / 0	1 / 2	2002-2017	-

CBR Works [53] and is now part of Empolis Information Access Suite [18]), Cognitive Systems ReMind [59, 60], and Isoft ReCALL [60].

Regarding the identified frameworks' actuality, popularity, and source code availability, we selected the following open-source frameworks to be further considered in this work: CloodCBR, eXiT*CBR, jColibri, myCBR, and ProCAKE.

4 Presentation of Selected CBR Frameworks

This section presents the five selected open-source frameworks in more detail. The authors collected the information from the literature, which was checked

[6] Two different branches of myCBR exist that are being developed separately.

[7] Search results for the CAKE framework led to several false positive results. However, the start of publications can be dated to 2005. In 2019, CAKE was renamed ProCAKE.

[8] Due to several false results (807 for `"CBR Works"`), `"CBR Works" Shell` was queried.

and completed by the framework's developers. We checked the information on the framework's profiles based on the available source code.

4.1 CloodCBR

CloodCBR supports textual and structural CBR ("Clood" is a Scottish word for cloud). A focus is put on support for distributed and highly scalable generic CBR systems based on a microservices architecture.

General Information

Developer	AI and Reasoning (AIR) Group, Robert Gordon University Aberdeen, United Kingdom
Homepage	http://cloodcbr.com/
CBR Types	Textual, structural
Applications based on the Framework	Talent management [23] Explainable Artificial Intelligence (via tool iSee) [62]
Documentation about the Framework	**Publications**: [39–41] **Guide and Documentation**: https://github.com/rgu-computing/clood
Source Code	**URL:** https://github.com/RGU-Computing/clood **Last Update:** Version 2.0 (March 2023) **Programming Language:** Python

Knowledge Containers

Vocabulary	**Case Representation**: Text, attribute-value, object **General Knowledge Representation**: Ontologies, taxonomies (table-based)
Case Base	**Structure**: Flat **Persistence**: NoSQL Search Engine (OpenSearch), CSV, JSON
Similarity Measures	**Local Measures (18)**: 1 generic measure (equals), 6 string measures (e. g., BM25 measure, word embedding vector-based), 5 numeric measures (e. g., interval, nearest number), 2 categorical measures (distance, table), 1 date measure (nearest date), 1 location measure (nearest location), 2 ontology measures (path-based, feature-based) **Global Measures (1)**: 1 aggregation measure (weighted sum) Supports integration of custom similarity measures via scripts Full list available at https://github.com/RGU-Computing/clood#available-similarity-metrics
Adaptation Knowledge	Not supported

CBR Phases

Retrieve	Parallel linear retriever, supporting strategies best match or minimum for each attribute
Reuse	Null adaptation, supports custom adaptation via scripts
Revise	Supports manual adjustment of values via GUI
Retain	Maintenance tasks like forgetting cases or recomputing of similarities

Interfaces

GUI	AngularJS-based client dashboard "Clood CBR Dashboard", visualizing complete CBR cycle Parallel plots visualization
API	Generic REST API, Docker interface

Additional Information

Distributed CBR	Supports building distributed, highly scalable, and generic CBR systems based on a microservices architecture
Special Features	API token management, cloud native implementation (serverless framework), multiple programming languages supported (e. g., Java, Python, JavaScript), scalable
Planned Features	Support graph data structures for case representation, include case adaptation methods, incorporate common case base maintenance techniques, improve access management for multi-tenancy by enhancing the user management module

4.2 eXiT*CBR

eXiT*CBR supports structural CBR and follows a modular approach. The framework is specifically developed for use in medical systems but is also applied in other domains. eXiT*CBR has an executable version with a comprehensive GUI.

General Information

Developer	Control Engineering and Intelligent Systems (eXiT) Research Group, University of Girona, Spain
Homepage	http://exitcbr.udg.edu/
CBR Types	Structural
Applications based on the Framework	Medicine and Healthcare: Cancer prognosis [33], premature baby monitoring [29], insulin dose recommendation [32, 58] Industry: Fault detection, plastic injection molding process Full list available at: exitcbr.udg.edu/publications.html
Documentation about the Framework	**Publications**: [30, 31, 43] **User Tutorial**: http://exit.udg.edu/files/eXiTCBR-4.0/exitCBRUserTutorial.pdf
Source Code	Source code is not publicly available, but sharable upon request. **URL:** http://exitcbr.udg.edu/downloads.html (Executable Version) **Last Update:** Version 4.1 (January 2017) **Programming Language:** Java

Knowledge Containers

Vocabulary	**Case Representation**: Attribute-value **General Knowledge Representation**: Not supported
Case Base	**Structure**: Flat **Persistence**: CSV
Similarity Measures	**Local Measures (9)**: 1 measure for unknown attributes, 1 string measure (Hamming distance), 2 numeric measures (Hamming and Euclidean distance), 1 boolean measure (equals), 1 date measure (distance-based), 3 sequence-measures (e. g., distance-based) **Global Measures (6)**: 6 aggregation measures (e. g., Euclidean, mean)
Adaptation Knowledge	Not supported

CBR Phases

Retrieve	Linear retriever
Reuse	Copy, 2 probabilistic methods for binary cases, 3 majority methods, 2 probabilistic multi-class methods
Revise	Not supported
Retain	Methods for deleting or adding cases are provided

Interfaces

GUI	GUI application for configuration, retrieval, visualization of results, and data export
API	Not supported

Additional Information

Distributed CBR	Supported since version 2.0 as a multi-agent-based system (partially supported with the executable version for experimental purposes, but not for deployment in a real distributed environment.)
Special Features	Provides plugin capabilities for special purpose: genetic algorithms (for feature learning), family risk calculator to manage GEDCOM data.
Planned Features	Adding existing methods for CBR phases reuse and retain from special-purpose development branches

4.3 jColibri

jColibri [46] supports textual, structural and conversational CBR [14]. The framework has a model to describe domain-specific knowledge. jColibri is a further development of the Colibri architecture [46]. It provides a library of problem-solving methods for knowledge-intensive CBR based on its own ontology. According to the developers, no further updates are currently planned, but the framework continues to be used in research like *Explainable Case-Based Reasoning* [50].

General Information

Developer	Group for Artificial Intelligence Applications (GAIA), Universidad Complutense de Madrid, Spain
Homepage	https://gaia.fdi.ucm.es/research/colibri/jcolibri/
CBR Types	Textual, structural, conversational
Applications based on the Framework	Medicine: Pain therapy [45], cancer [36], decision support [13] Industry: Energy optimization [19], knowledge management [36] Full list available at: https://gaia.fdi.ucm.es/research/colibri/index.php#users
Documentation about the Framework	**Publications**: [8, 14, 44, 46] **Examples**: https://gaia.fdi.ucm.es/research/colibri/jcolibri/index.php#examples **API Documentation**: https://gaia.fdi.ucm.es/research/colibri/jcolibri/doc/apidocs/index.html
Source Code	**URL:** https://sourceforge.net/projects/jcolibri-cbr/ **Last Update:** Version 3.0 (Jan. 2019) **Programming Language:** Java

Knowledge Containers

Vocabulary	**Case Representation**: Text, attribute-value Case divided into description, solution, justification of solution and result **General Knowledge Representation**: Ontology that includes cases
Case Base	**Structure**: Flat, index-based (KD tree), ontology-index **Persistence**: databases, plain text, XML, CSV, ontology
Similarity Measures	**Local Measures**: 1 generic measure (equals), 2 string measures (case-independent equals, maximum common substring), 4 numeric measures (e. g., interval, threshold), 2 categorical (e. g., enumeration distance), 4 ontology measures (e. g., cosine, path-based) **Global Measures (1)**: 1 aggregation measure (average)
Adaptation Knowledge	Not supported

CBR Phases

Retrieve	Linear retriever, index-based retriever, filter-based retriever, MAC/FAC retriever, ontology retriever
Reuse	Null adaptation, numeric proportion
Revise	Generic constructs given, no methods implemented
Retain	Generic constructs given, methods for deleting or adding cases are provided, noise removal methods available (case base maintenance)

Interfaces

GUI	Tool "Colibri Studio" enables no-code usage of the framework (https://gaia.fdi.ucm.es/research/colibri/colibristudio/) Case base visualization tools
API	Java interface

Additional Information

Distributed CBR	Not supported
Special Features	Includes metrics for evaluating the case base (including visualization), ReColibry (https://gaia.fdi.ucm.es/research/colibri/recolibry/index.php): extension for the construction of case-based recommender systems, xColibri(https://gaia.fdi.ucm.es/research/colibri/xcolibri/): evolution of the Colibri platform, for the explanation of intelligent Systems with CBR
Planned Features	**Short Term**: Inclusion of case-based explanation methods **Long Term**: Deployment of PyColibri, a python port of jColibri

4.4 myCBR

myCBR is designed for structural CBR. A focus is put on knowledge-intensive measures and the retrieval phase. In myCBR, elements of the CBR Works framework (see Sect. 3) have been adopted. The last release of myCBR3 documented on the project website is from May 2015. However, the framework is being further developed in two different branches. In the following profile, features that are only available in one branch are marked with either *UH* or *NTNU*.

General Information

Developer	Competence Center Case-Based Reasoning (CCCBR), DFKI, Germany School of Computing and Technology, University of West London, UK University of Hildesheim (UH), Germany Norwegian University of Science and Technology (NTNU), Norway
Homepage	http://mycbr-project.org/ NTNU: https://github.com/ntnu-ai-lab/mycbr-sdk
CBR Types	Textual, structural
Applications based on the Framework	Cooking domain (CookIIS), planning in games, config. of racing cars [61] UH: Recruiting processes [51], architectural floor plans [15,49], aircraft maintenance [47] NTNU: selfBACK (https://www.selfback.eu/), SupportPrim https://www.ntnu.no/supportprim
Documentation about the Framework	**Publications**: [4,6,22,55], UH: [56], NTNU: [5] Tutorial: http://mycbr-project.org/tutorials.html
Source Code	**URL:** http://mycbr-project.org/download.html Version 3 (May 2015) UH: https://github.com/jmschoenborn/myCBR-SDK (Dec. 2022) NTNU: https://github.com/ntnu-ai-lab/mycbr-sdk (May 2019) **Programming Language:** Java

Knowledge Containers

Vocabulary	**Case Representation**: Text, attribute-value, object **General Knowledge Representation**: Taxonomies
Case Base	**Structure**: Flat **Persistence**: CSV, databases
Similarity Measures	**Local Measures (8)**: 4 string measures (e. g., equals, word-based, character-based, taxonomy-based), 2 numeric measures (default and user-defined distance), 2 categorical measures (order-based, table-based) **Global Measures (4)**: 4 aggregation measures (weighted sum, Euclidean, minimum, maximum) Supports the integration of custom similarity measures written in Jython
Adaptation Knowledge	Manual acquisition of adaptation rules

CBR Phases

Retrieve	Linear retriever, index-based retriever
Reuse	Null adaptation, adaptation rules
Revise	Not supported
Retain	Generic constructs given, no methods implemented

Interfaces

GUI	Tool "myCBR Workbench", provides modeling similarity measures and case base view
API	Java SDK (UH), REST-API (NTNU)

Additional Information

Distributed CBR	Not supported
Planned Features (UH)	**Short Term**: Adding Jaro-Winkler distance for string comparison, integration of explainability via explanation patterns, counterfactuals, and visualization, integration of maintenance methods (work in progress) **Long Term**: Extension of explanation capabilities with additional approaches, integration of the case factory maintenance approach with domain-specific language, explanation capabilities for maintenance actions, integration of adaptation step for rule-based adaptations, knowledge modeling and visualization with a virtual reality component

4.5 ProCAKE

ProCAKE (*Process-oriented Case-based Knowledge Engine*) is a domain-independent CBR framework that focuses on structural and process-oriented CBR [10]. ProCAKE has evolved from CAKE (*Collaborative Agile Knowledge Engine*) [9]. It provides a generic data type model for custom case representations, various syntactic and semantic similarity measures, and several retrieval algorithms [10].

General Information

Developer	Department of Business Information Systems II, University of Trier, Germany Experience-Based Learning Systems, DFKI Trier Branch, Germany
Homepage	https://procake.uni-trier.de
CBR Types	Textual, structural, process-oriented

Applications based on the Framework	Cooking domain: CookingCAKE (https://cookingcake.wi2.uni-trier.de/) IoT/Smart factory data: Adaptive production, cyber-physical systems [34] Business and scientific workflows: Modeling, adaptation [63], flexible execution [17] Full list available at: https://procake.uni-trier.de/publications
Documentation about the Framework	**Publications**: [10] **Demo Project**: https://gitlab.rlp.net/procake/procake-demos/ **Wiki**: https://procake.pages.gitlab.rlp.net/procake-wiki/
Source Code	**URL:** https://gitlab.rlp.net/procake/procake-framework **Last Update:** Version 4 (Apr. 2023) **Programming Language:** Java

Knowledge Containers

Vocabulary	**Case Representation**: Text, attribute-value, object, (semantic) graphs, collections (list, set), custom data classes (XML, Java) **General Knowledge Representation**: Ontologies, taxonomies
Case Base	**Structure**: Flat **Persistence**: JSON, XML, CSV
Similarity Measures	**Local Measures (41)**: 2 generic measures (table-based, equals), 12 string measures (e. g., Levenshtein, taxonomy-based), 5 numeric and date measures (e. g., distance, Sigmoid), 6 sequence measures (e. g., DTW, SWA), 1 interval measure (distance-based), 8 ontology measures (e. g., path-based, equivalence), 7 graph/process measures (e. g., DTW, SWA, Levenshtein, A*-based mapping) **Global Measures (7)**: 7 aggregation measures (e. g., average, maximum, minimum, Minkowski) Possibility to write own similarity measures (Java) Full list available at https://procake.pages.gitlab.rlp.net/procake-wiki/
Adaptation Knowledge	Domain-independent adaptation manager

CBR Phases

Retrieve	Linear retriever, parallel linear retriever, MAC/FAC retriever A*-parallel retriever (for graphs)
Reuse	Null adaptation, integration of custom adaptation methods supported
Revise	Not supported
Retain	Methods for deleting or adding cases are provided

Interfaces

GUI	Object editor for all system and custom data classes (https://gitlab.rlp.net/procake/procake-gui)
API	Java interface, generic REST API (https://gitlab.rlp.net/procake/procake-rest-api)

Additional Information

Distributed CBR	Not supported
Special Features	Embedding-based similarity measures and retrieval methods for structural and process-oriented cases (https://gitlab.rlp.net/procake-embedding) [21] Consideration of inter-case dependencies [27] Evaluation methods for retrievers
Planned Features	**Short Term (Work in progress):** integration of existing adaption algorithms (e.g., generalization/specialization, rule-based), integration of existing learning methods (e.g., embedding approaches) **Long Term**: Support for distributed CBR, cluster-based retrieval [37], conversational CBR [64], additional interfaces (GUI, docker, command line), explanation by visualization of similarities [52]

5 Comparison of Selected CBR Frameworks

In this section, the five presented frameworks are compared regarding the categories of the fact sheets[9]. Table 2 shows the supported case representations.

Tab. 2. Supported Case Representations per Framework.

CBR Framework	Text	Attribute-Value	Object	Graph
CloodCBR	X	X	X	
eXiT*CBR		X		
jColibri	X	X		
myCBR	X	X	X	
ProCAKE	X	X	X	X

In general, all frameworks support structural CBR. Except for eXiT*CBR, all frameworks also support textual CBR, storing texts as individual cases and providing corresponding similarity measures. jColibri is the only framework with a conversational component, while ProCAKE is the only one that can handle graphs or processes as cases. Some frameworks have been applied in the same application domains. For example, eXiT*CBR and jColibri are used in the medical domain, while myCBR and ProCAKE are used in the cooking domain. All frameworks are written in Java, except for CloodCBR, which is implemented in Python. The published source code of CloodCBR and ProCAKE is currently the most up-to-date. However, eXiT*CBR and myCBR are both still under active development. Only for jColibri, no further updates are currently planned.

ProCAKE is the only one that explicitly supports defining user-specific data classes. For general knowledge representations, CloodCBR, myCBR, and ProCAKE support taxonomies, and CloodCBR, jColibri, and ProCAKE support ontologies. While all frameworks provide a flat case base structure, jColibri follows an index- and an ontology-based approach, enabling more efficient storage. The case bases can be imported in different formats. It stands out that jColibri can connect to ontologies and databases and that CloodCBR connects to NoSQL databases. The built-in similarity measures differ greatly by the framework. ProCAKE has most of the similarity measures, both at the local and global level. For semantic measures, myCBR and ProCAKE support taxonomy-based measures, and CloodCBR, jColibri, and ProCAKE support ontology-based measures. CloodCBR incorporates neural language measures. For incorporating adaptation knowledge, myCBR provides manual acquisition of adaptation rules.

Regarding the CBR phases, it can be seen that the focus of all frameworks is on the retrieval and reuse phases. CloodCBR and eXiT*CBR each provide a single linear retriever. The retrieval is parallelized in CloodCBR by default. ProCAKE has several retrievers built in. MAC/FAC retrievers are implemented

[9] A complete overview table is available at: https://git.opendfki.de/easy/overview-and-comparison-of-cbr-frameworks/-/blob/main/Overview-Table.pdf

in jColibri and ProCAKE. myCBR and jColibri also have index-based retrievers. jColibri has an ontology-based retriever, while ProCAKE has a special graph retriever. In the reuse phase, all frameworks support null adaptation. eXiT*CBR provides several probabilistic and majority methods, jColibri provides numeric proportion. myCBR supports adaptation rules. CloodCBR enables the integration of custom adaptation methods via scripts. ProCAKE does this similarly via its adaptation manager. The frameworks do not provide specific support for the revise phase. CloodCBR offers a manual adaptation of values in the interface, jColibri has generic methods built in. CloodCBR has maintenance tasks for the retain phase, such as forgetting or re-computing similarity values. jColibri provides methods for noise removal. myCBR and jColibri provide a generic framework for retainment. Basic maintenance tasks can be performed in any framework. In general, it can be seen that the retrieve and reuse phases are well-supported, whereas support for revise and retain can be enhanced in the frameworks.

Regarding the interfaces, the frameworks differ in the provided GUIs and APIs. CloodCBR, eXiT*CBR, and jColibri can be configured via GUI without programming. Hence, they are recommended for no-code development. myCBR and ProCAKE only provide partial GUIs for specific tasks. Java and REST interfaces are available for CloodCBR, myCBR, and ProCAKE. CloodCBR and ProCAKE also provide a Docker interface.

Each framework provides some special features listed in the respective fact sheets. With the ever-growing amounts of data, it is assumed that Distributed CBR will become more relevant. jColibri, myCBR, and ProCAKE do not support distributed approaches yet. eXiT*CBR allows for the development of a multi-agent system for experimental purposes. CloodCBR also offers Distributed CBR through its microservices architecture and is the most advanced system in this respect. However, it can be stated that building applications for larger real environments needs further development of all frameworks.

To conclude, no framework is the most feature-rich or best in all categories. They all have different strengths or unique features. For specific application requirements, more suitable frameworks can be identified. For instance, eXiT*CBR or CloodCBR is suitable for Distributed CBR, jColibri supports dialog-oriented applications, and graph or process-oriented applications can be implemented with ProCAKE. It is desirable to provide interfaces between the different frameworks to combine the strengths of the different frameworks. For example, frameworks could integrate similarity measures or adaptation methods of other frameworks or case bases, and general knowledge could be exchanged more easily.

6 Conclusion

This paper aims to overview and compare the CBR frameworks developed to facilitate the development of CBR applications. As the most recent open-source frameworks, CloodCBR, eXiT*CBR, jColibri, myCBR, and ProCAKE have been identified and presented based on various criteria. Similarities and differences between the frameworks are examined, whereby it becomes apparent that each framework has its strengths and that certain frameworks are more suitable for different application areas. Some features, such as the support for Distributed CBR, are built in the frameworks with varying extent and maturity, so further investigation is needed to select the most suitable framework as a basis for developing specific applications. This work aims to provide decision support for this purpose.

The developers of the frameworks have various plans for further development of existing and for adding new features, from which trends for future research can be derived. A common goal of myCBR, jColibri, and ProCAKE is integrating explanation methods, which aligns with the recognized need for explainable Artificial Intelligence. myCBR and ProCAKE plan to extend the visualization components. CloodCBR, eXiT*CBR, and myCBR plan to integrate maintenance methods and approaches. The implementation of reuse methods (eXiT*CBR) or the inclusion of adaptation methods (CloodCBR and ProCAKE), as well as the extension of adaptation (myCBR), are other common goals. Certain features already provided in some frameworks will also become available in other frameworks according to the development plans. For example, support for graphs is planned for CloodCBR, ProCAKE plans to support distributed and conversational CBR, and jColibri plans to port the framework to Python. As a novel planned feature, a virtual reality component for knowledge modeling and visualization is mentioned for myCBR.

We thank all developers for their efforts and for making the frameworks available to the public.

Acknowledgments. We would like to thank the following people for contributing to the compilation of the CBR framework fact sheets: Ikechukwu Nkisi-Orji and Chamath Palihawadana (CloodCBR), Beatriz López, Oscar Raya, and Jonah Fernández (eXiT*CBR), Juan Antonio Recio García (jColibri), and Pascal Reuss (myCBR). This work is funded by the Federal Ministry for Economic Affairs and Climate Action under grant No. 01MD22002C *EASY*.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. **7**(1), 39–59 (1994)
2. Abasolo, C., Plaza, E., Arcos, J.L.: Components for Case-Based Reasoning Systems. In: LNAI. pp. 1–16. No. 2504, Springer (2002)
3. Atanassov, A., Antonov, L.: Comparative Analysis of Case Based Reasoning Software Frameworks jColibri and myCBR. J. Chem. Technol. Metall. **47**(1), 83–90 (2012)
4. Bach, K., Althoff, K.: Developing Case-Based Reasoning Applications Using myCBR 3. In: 20th ICCBR Proc. LNCS, vol. 7466, pp. 17–31. Springer (2012)
5. Bach, K., Mathisen, B.M., Jaiswal, A.: Demonstrating the myCBR Rest API. In: 27th ICCBR Workshop Proc. CEUR Workshop Proc., vol. 2567, pp. 144–155. CEUR-WS.org (2019)

6. Bach, K., Sauer, C.S., Althoff, K., Roth-Berghofer, T.: Knowledge Modeling with the Open Source Tool myCBR. In: 21st ECAI Workshop Proc. CEUR Workshop Proceedings, vol. 1289. CEUR-WS.org (2014)
7. Begum, S., Ahmed, M.U., Funk, P., Xiong, N., Folke, M.: Case-Based Reasoning Systems in the Health Sciences: A Survey of Recent Trends and Developments. IEEE Trans. Syst. Man Cybern. Part C **41**(4), 421–434 (2011)
8. Bello-Tomás, J.J., González-Calero, P.A., Díaz-Agudo, B.: jColibri: An Object-Oriented Framework for Building CBR Systems. In: 7th ECCBR Proc. LNCS, vol. 3155, pp. 32–46. Springer (2004)
9. Bergmann, R., Gil, Y.: Similarity Assessment and Efficient Retrieval of Semantic Workflows. Inf. Syst. **40**, 115–127 (2014)
10. Bergmann, R., Grumbach, L., Malburg, L., Zeyen, C.: ProCAKE: A Process-Oriented Case-Based Reasoning Framework. In: 27th ICCBR Workshop Proc. (2019)
11. Bogaerts, S., Leake, D.: Technical Report 617 IUCBRF: A Framework For Rapid And Modular Case-Based Reasoning System Development Report Version 1.0. IU Bloomington (2005)
12. Bogaerts, S., Leake, D.B.: Increasing AI Project Effectiveness with Reusable Code Frameworks: A Case Study Using IUCBRF. In: 18th FLAIRS. FloridaOJ. pp. 2–7. AAAI Press (2005)
13. Bruland, T., Aamodt, A., Langseth, H.: Architectures Integrating Case-Based Reasoning and Bayesian Networks for Clinical Decision Support. In: 6th IFIP Proc. IFIP Adv. Inf. Commun. Technol., vol. 340, pp. 82–91. Springer (2010)
14. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A.: Building CBR Systems with jColibri. SCP **69**(1-3), 68–75 (2007)
15. Eisenstadt, V., Langenhan, C., Althoff, K.D.: Generation of Floor Plan Variations with Convolutional Neural Networks and Case-based Reasoning – An Approach for Unsupervised Adaptation of Room Configurations within a Framework for Support of Early Conceptual Design. In: eCAADe SIGraDi Conference, Porto (2019)
16. ElKafrawy, P., Mohamed, R.A.: Comparative Study of Case-Based Reasoning Software. IJSRM **1**(6), 224–233 (2015)
17. Grumbach, L., Bergmann, R.: Towards Case-Based Deviation Management for Flexible Workflows. In: Jäschke, R., Weidlich, M. (eds.) LWDA 2019. CEUR Workshop Proc., vol. 2454, pp. 241–252. CEUR-WS.org (2019)
18. He, W., Wang, F.: Integrating a Case-Based Reasoning Shell and Web 2.0: Design Recommendations and Insights. World Wide Web **19**(6), 1231–1249 (2016)
19. Heilala, J., Klobut, K., Salonen, T., Siltanen, P., Ruusu, R., Urosevic, L., Reimer, P., Armijo, A., Sorli, M., Fatur, T., et al.: Ambient Intelligence Based Monitoring and Energy Efficiency Optimization System. In: ISAM 2011. pp. 1–6. IEEE (2011)
20. Hinkle, D., Toomey, C.: Applying Case-Based Reasoning to Manufacturing. AI Magazine **16**(1), 65–65 (1995)
21. Hoffmann, M., Bergmann, R.: Using Graph Embedding Techniques in Process-Oriented Case-Based Reasoning. Algorithms **15**(2), 27 (2022)
22. Hundt, A., Reuss, P., Sauer, C.S., Roth-Berghofer, T.: Knowledge Modelling and Maintenance in myCBR3. In: 16th LWA Workshop Proc. CEUR Workshop Proc., vol. 1226, pp. 264–275. CEUR-WS.org (2014)
23. Husni, H.S., Ramadhan, A., Abdurachman, E., Trisetyarso, A.: Indonesia Digital Government Auditing Model Using Rule Based and Cloud Case-Based Reasoning. Int. J. Sci. Technol. Res. **1**(2), 60–63 (Jul 2022)
24. Jaczynski, M.: A Framework for the Management of Past Experiences with Time-Extended Situations. In: 6th CIKM Proc. pp. 32–39. ACM (1997)

25. Jaczynski, M., Trousse, B.: An Object-Oriented Framework for the Design and the Implementation of Case-Based Reasoners. In: CBR Workshop Proc. (1998)
26. Kolodner, J.L.: Reconstructive Memory: A Computer Model. Cogn. Sci. **7**(4), 281–328 (1983)
27. Kumar, R., Schultheis, A., Malburg, L., Hoffmann, M., Bergmann, R.: Considering Inter-Case Dependencies During Similarity-Based Retrieval in Process-Oriented Case-Based Reasoning. In: 35th FLAIRS. FloridaOJ (2022)
28. Lebowitz, M.: Memory-Based Parsing. AI **21**(4), 363–404 (1983)
29. López, B., Gamero, F., Coll, J., Mordvaniuk, N., Bassols, J., Bargallo, E., Guarch, B., López-Bermejo, A.: Intelligent System for Premature Babies Healthcare at Home based on Case-based Reasoning. In: 2nd IWBBIO. pp. 1278–1289. Copicentro Editorial (2014)
30. López, B., Pous, C.: eXiT*CBR: A Tool Supporting RRI. In: 21st CCIA. Frontiers in Artificial Intelligence and Applications, vol. 308, pp. 176–179. IOS Press (2018)
31. López, B., Pous, C., Gay, P., Pla, A., Sanz, J., Brunet, J.: eXiT*CBR: A Framework for Case-Based Medical Diagnosis Development and Experimentation. Artif. Intell. Med. **51**(2), 81–91 (2011)
32. López, B., Raya, Ò., Baykova, E., Sáez, M., Rigau, D., Cunill, R., Mayoral, S., Carrión, C., Serrano, D., Castells, X.: APPRAISE-RS: Automated, Updated, Participatory, and Personalized Treatment Recommender Systems Based on Grade Methodology. Heliyon **9**(2), e13074 (2023)
33. López, B., Pous, C., Plà, A., Gay, P., Brunet, J.: Breast Cancer Prognosis Through CBR. In: 27th ICCBR Workshop Proc. pp. 105–110 (2012)
34. Malburg, L., Brand, F., Bergmann, R.: Adaptive Management of Cyber-Physical Workflows by Means of Case-Based Reasoning and Automated Planning. In: EDOC 2022. LNBIP, vol. 466, pp. 79–95. Springer (2022)
35. Mark, W.S.: Case-Based Reasoning for Autoclave Management. In: CBR Workshop Proc. pp. 176–180. DARPA - Information Science (1989)
36. Martín, A., León, C.: Expert knowledge management based on ontology in a digital library. In: 12th ICEIS Proc. pp. 291–298. SciTePress (2010)
37. Müller, G., Bergmann, R.: A Cluster-Based Approach to Improve Similarity-Based Retrieval for Process-Oriented Case-Based Reasoning. In: 21st ECAI. vol. 263, pp. 639–644. IOS Press (2014)
38. Nguyen, T., Czerwinski, M., Lee, D.: Compaq QuickSource: Providing the Consumer With the Power of Artificial Intelligence. In: IAAIC Proc. pp. 142–151 (1993)
39. Nkisi-Orji, I., Palihawadana, C., Wiratunga, N., Corsar, D., Wijekoon, A.: Adapting Semantic Similarity Methods for Case-Based Reasoning in the Cloud. In: 30th ICCBR Proc. pp. 125–139. Springer (2022)
40. Nkisi-Orji, I., Wiratunga, N., Palihawadana, C., Recio-García, J.A., Corsar, D.: Clood CBR: Towards Microservices Oriented Case-Based Reasoning. In: 28th ICCBR Proc. pp. 129–143. Springer (2020)
41. Palihawadana, C., Nkisi-Orji, I., Wiratunga, N., Corsar, D., Wijekoon, A.: Introducing Clood CBR: A Cloud Based CBR Framework. In: 30th ICCBR Workshop Proc. CEUR Workshop Proc., vol. 3389, pp. 233–234. CEUR-WS.org (2022)
42. Peixoto, T.F., Martinazzo, L.A., Weber, R.O.: Cyberinfrastructure Requirements for Research Communities. In: 7th CIKI. vol. 1 (2017)
43. Pla, A., López, B., Gay, P., Pous, C.: eXiT*CBR.v2: Distributed Case-Based Reasoning Tool for Medical Prognosis. Decis. Support Syst. **54**(3), 1499–1510 (2013)
44. Recio, J.A., Sánchez, A., Díaz-Agudo, B., González-Calero, P.: jColibri 1.0 in a Nutshell. A Software Tool for Designing CBR systems. In: 10th UKCBR Workshop Proc. pp. 1–11 (2005)

45. Recio-García, J.A., Díaz-Agudo, B., Jorro-Aragoneses, J.L., Kazemi, A.: Intelligent Control System for Back Pain Therapy. In: 25th ICCBR Proc. LNCS, vol. 10339, pp. 287–301. Springer (2017)
46. Recio-García, J.A., González-Calero, P.A., Díaz-Agudo, B.: jColibri2: A Framework for Building Case-Based Reasoning Systems. SCP **79**, 126–145 (2014)
47. Reuss, P., Stram, R., Althoff, K., Henkel, W., Henning, F.: Knowledge Engineering for Decision Support on Diagnosis and Maintenance in the Aircraft Domain. In: Synergies Between Knowledge Engineering and Software Engineering, Adv. Intell. Syst. Comput., vol. 626, pp. 173–196. Springer (2018)
48. Richter, M.M.: Knowledge Containers. In: Readings in CBR. MKP (2003)
49. Sabri, Q.U., Bayer, J., Ayzenshtadt, V., Bukhari, S.S., Althoff, K.D., Dengel, A.: Semantic Pattern-based Retrieval of Architectural Floor Plans with Case-based and Graph-based Searching Techniques and their Evaluation and Visualization. In: ICPRAM. pp. 50–60 (2017)
50. Schoenborn, J.M., Weber, R.O., Aha, D.W., Cassens, J., Althoff, K.D.: Explainable Case-Based Reasoning: A Survey. In: AAAI-21 Workshop Proceedings (2021)
51. Schoenborn, J.M., Reuss, P., Wenzel, C., Althoff, K.: Towards a Case-Based Decision Support System for Recruiting Processes using T-Shapes. In: Modellierung-C 2020. CEUR Workshop Proc., vol. 2542, pp. 165–171. CEUR-WS.org (2020)
52. Schultheis, A., Hoffmann, M., Malburg, L., Bergmann, R.: Explanation of Similarities in Process-Oriented Case-Based Reasoning by Visualization. In: 31st ICCBR Proc. LNCS, Springer (2023)
53. Schulz, S.: CBR-Works - A State-Of-The-Art Shell for Case-Based Application Building. In: 7th GWCBR Proc. vol. 99, pp. 3–12. Citeseer (1999)
54. Schumacher, J.: Empolis Orenge – an Open Platform for Knowledge Management Applications. In: 1st German Workshop on Experience Mgmt. pp. 61–62. GI (2002)
55. Stahl, A., Roth-Berghofer, T.: Rapid Prototyping of CBR Applications with the Open Source Tool myCBR. In: Proceedings of the 9th ECCBR. LNCS, vol. 5239, pp. 615–629. Springer (2008)
56. Stram, R., Reuss, P., Althoff, K.: Dynamic Case Bases and the Asymmetrical Weighted One-Mode Projection. In: 26th ICCBR Proc. LNCS, vol. 11156, pp. 385–398. Springer (2018)
57. Thakur, N., Chhabra, T., Verma, D., Kumar, D., Dayal, M.: Case Based Reasoning: A Comparative Analysis of CBR Tools. IJIRCST **4**, 11190–11196 (2016)
58. Unsworth, R., Armiger, R., Jugnee, N., Thomas, M., Herrero, P., Georgiou, P., Oliver, N., Reddy, M.: Safety and Efficacy of an Adaptive Bolus Calculator for Type 1 Diabetes: a Randomised Control Cross over Study. Dia. Techn. Ther. (2023)
59. Watson, I.D.: Applying Case-Based Reasoning – Techniques for the Enterprise Systems. MKP (1997)
60. Watson, I.D., Marir, F.: Case-Based Reasoning: A Review. Knowl. Eng. Rev. **9**(4), 327–354 (1994)
61. Wenzel, C., Reuss, P., Rose, K., Althoff, K.D.: Multi-Agent, Case-Based Configuration of Custom-Built Racing Cars. In: 19th UKCBR Workshop Proc. (2014)
62. Wijekoon, A., Wiratunga, N., Palihawadana, C., Nkisi-Orji, I., Corsar, D., Martin, K.: iSee: Intelligent Sharing of Explanation Experience by Users for Users. In: 28th IUI Comp. Proc. pp. 79–82 (2023)
63. Zeyen, C., Malburg, L., Bergmann, R.: Adaptation of Scientific Workflows by Means of Process-Oriented Case-Based Reasoning. In: 27th ICCBR Proc. pp. 388–403. LNAI, Springer (2019)
64. Zeyen, C., Müller, G., Bergmann, R.: Conversational Process-Oriented Case-Based Reasoning. In: 25th ICCBR Proc. LNCS, vol. 10339, pp. 403–419. Springer (2017)