CBR-Based Search in BULCHINO Catalogue

Stanimir Stoyanov, Nadezda Govedarova, Ivan Popchev

University of Plovdiv, Bulgaria
E-Commerce Laboratory
s.stojanov@isy-dc.com, nadig@abv.bg, ipopchev@iit.bas.bg

Abstract. In this paper we present an ongoing work on a Case Based Reasoning approach. For extending the search layer in BULCHINO Catalogue (Cultural, Historical and Natural Objects of Bulgaria). For the implementation of this approach we apply the Object-Oriented framework jCOLIBRI. We provide a general overview of the framework architecture and describe how it could be applied for the configuration of a particular application. To illustrate the practical applicability and usefulness of our approach, we developed the PersonExample Module - a simple application for searching single objects in Personal and corporate name authority in BULCHINO Catalogue.

1 Introduction

The E-Commerce Laboratory (ECL) in the University of Plovdiv is working on the development of Service Oriented Architectures [3] to apply them for the configuration of cultural and educational systems and virtual nets. As a result, two essential projects have been under development: BULCHINO [5, 33], an Intelligent Web-based Catalogue for electronic presentation of culture-historical heritage of Bulgaria and DeLC (Distributed eLearning Center) [30,31,32], whose main objective is to create the organizational and technological infrastructure of the centre. The access to the available services and electronic content, in both projects, is accomplished via specialized portals [25], whose architecture is developed in compliance with the referent model of Delphi Group [7]. The portal reference architecture includes the following nine different layers: presentation, personalization, e-services control, integration, collaboration, search, categorization, and loop process layer.

Currently, we have been working on the development of new versions of these portals. We intend to extend the search layer, so that different models for searching are available. Providing a flexible and efficient method for searching culture-historical objects in BULCHINO Catalogue is of primary importance to us in order to guarantee the user with a set of results actually related to his/her interest. We consider that applying the Case Based Reasoning technology for the implementation of the search task in BULCHINO is an appropriate approach [34, 35, 36].

Case Based Reasoning [1, 19] is generally described as a reasoning paradigm from the Artificial Intelligence Area. Its main idea is to reuse experience to solve new problems. This experience is in the form of cases and is stored in a case base, ready for being reused. Developing a CBR application is a complex task where many

decisions must be made. The system designer has to choose how the cases will be represented, the case organization structure, which methods will solve the CBR tasks and which knowledge will be used by these methods. This process would greatly benefit from the reuse of previously developed CBR applications [16]. Software reuse is one of the main goals of the Software Engineering community and many researches and studies are carried out in this direction. As a result many technologies have appeared in the last years that directly or indirectly promote software reuse: object-oriented frameworks, component technologies, design patterns, model driven architectures and others are just a few. We are particularly interested in the Application Frameworks, targeted at a given application domain providing the design for CBR applications within that domain [17, 18].

In this paper we describe how the Framework jCOLIBRI is adapted and utilized for the configuration of a CBR-based application in the particular domain BULCHINO Catalogue. jCOLIBRI is a technological evolution of COLIBRI [15] that incorporates in a distributed architecture a description logics(DLs) engine[12, 13], GUI clients for assembling a CBR system from reusable components[27] and an objects-oriented framework implemented in Java [4]. Since BULCHINO Catalogue is integrated in J2EE [14] environment and has distributed architecture, we think that the framework could be easily adapted for the development of PersonExample Module – a simple application for searching and retrieving single objects from "Personal and corporate name authority" in BULCHINO. Our aim is to extend this module and integrate it in the next version of the Catalogue, so that the users could make both local queries to a single authority (when the user is searching a particular Cultural Object) and global queries to the whole catalogue (when the user is interested in tourist routes, comprising several Cultural Objects).

The rest of the paper is organized as follows. Next section describes the main characteristics of BULCHINO Catalogue. Section 3 provides a general overview about jCOLIBRI architecture and sets out our motivation for choosing this CBR Framework. Section 4 exemplifies the usage of jCOLIBRI to create the PersonExample Module, where we describe step by step how the new application is configured and got ready to be executed. In Section 5 some final remarks about our future work are presented.

2 BULCHINO

BULCHINO project (Cultural, historical and natural objects of Bulgaria) [30] aims at developing a common catalogue for representation of the Bulgarian culture-historical heritage. It comprises a common meta-catalogue [23] for classification of cultural works, which is based on a proposition of UNESCO. A major aspect to ensure the interoperability of the catalogue is the usage of appropriate standard, which facilitates homogeneous descriptions, efficient search, shared usage as well as the management of cultural-historical works. The employment of such a standard makes possible the exchange of information between different cultural institutions, museums, universities, etc; usage of the catalogue in different areas and its inclusion in the European and worldwide cultural networks.

A working version of the CCO (Cataloging Cultural Objects) [5] standard is first published in February 2005 by Visual Resources Association [37]. It is specially designed for communities and organizations, which deal with description and documentation of work of arts, cultural artifacts and their place of presence productions such as museums, libraries, galleries etc. The CCO standard mainly focuses on four control authorities, one work record and one image record.

- *Work record* – the data is stored using required and additional recommended elements that give concrete information about the cultural-historical work;
- *Subject authority* – contains subjects from the iconographic terminology, names from literatural, mythological and religious nature, historical event;
- *Concept authority* – contains terminology, needed for the description of work of art, the material that the work is made of, the activities involved with the work, style, role of the creator, etc;
- *Geographic place authority* – contains information about geographical locations of the cultural works and their creators;
- *Personal and corporate name authority* – contains information about architectural work, the individuals and corporations, connected with the cultural-historical work;
- *Image record* – contains an image of the cultural work.

3 CBR-Based Framework jCOLIBRI

jCOLIBRI is an object-oriented framework for developing CBR systems[4,16]. It offers an easier development process that is based on the reuse of past designs and implementations. jCOLIBRI formalizes the CBR knowledge using domain independent CBR ontology (CBROnto)[9], which is mapped over the classes of the framework, a knowledge-level description of the CBR tasks[10] and a library of reusable Problem Solving Methods (PSMs).

Our motivation for choosing this framework is based on a comparative analysis between jCOLIBRI framework and other frameworks, designed to facilitate the development of CBR applications. jCOLIBRI enhances the other CBR shells: CAT-CBR[6], CBR*Tools[22], IUCBRF [28], Orenge [24] in several aspects: availability (open source framework), implementation (the Java implementation implies to a great extend usability, extensibility and user acceptance), GUI (the provided graphical tools facilitate the system design). Another justification for our choice is connected with the fact that jCOLIBRI affords the opportunity to apply Case Retrieval Nets (CRN) model [20, 21] as Case Base organization structure. We consider implementing this approach to the Case Base management as a next step in our studies on this issue.

Figure 1 depicts the architecture of jCOLIBRI Framework. It comprises four main modules: Case Base (In-memory Organization of Cases), Task and Method Ontology and jCOLIBRI Core. It also shows the four different data sources types (Persistence Layer), which are connected with the other framework components via objects, referred to as Connectors.

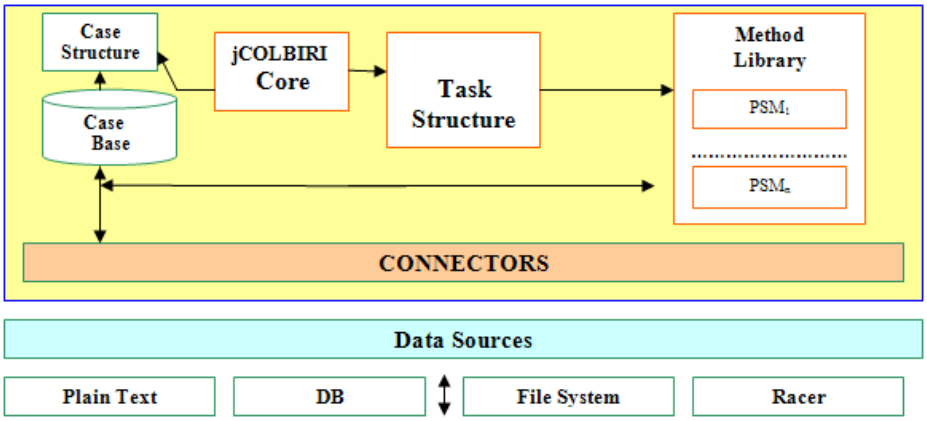

Fig. 1. jCOLIBRI Architecture

3.1 Case Structure

jCOLIBRI represents a case in a very general way[2]. The framework support several case structures, from plain attribute value records to hierarchical trees with composed attributes.

A case is composed of Description (describes the problem by means of several attributes), Solution (contains the description of the solution of the case) and Result (represents the result of applying the case to a real situation). Description and Solution are sets of Attributes, and there are two types of attributes: simple and compound. Simple Attributes are described by Name, Type, Weight and Local Similarity Function. Compound Attributes collect other simple attributes, allowing complex case structure. When two cases are compared, the local similarity functions are used to compare simple attribute values. Global similarity functions are linked to compound attributes and are used to gather similarities of the collected attributes in a unique similarity value. At last the similarity value of two cases is computed as the similarity of their Description concepts.

3.2 Case Base and Connectors

jCOLIBRI splits the problem of Case Base Management into two separate although related concerns: persistency mechanism and in-memory organization [15].

Persistence layer is composed by several connectors that allow developers to change the data storage sources very easily. Connectors are objects that know how to access and retrieve cases from the medium and return those cases to the CBR system in a uniform way. jCOLIBRI implements Plain Text Connector, JDBC Connector, File System Connector and Racer Connector [26].

The second layer of Case Base Management is the CaseBase (data structure) used to organize the cases once read and loaded by the connector into memory.

3.3 Tasks /Methods Ontology

The most well known knowledge level analysis applied to CBR systems describes the general CBR cycle in terms of four tasks at the highest level of generality: Retrieve the most similar case/s, reuse its/their knowledge to solve the problem, Revise the proposed Solution and Retain the experience [10]. Each one of the CBR tasks involves a number of more specific subtasks. jCOLIBRI includes additional methods to solve the Preparation and Maintenance Tasks, named PreCycle (loads the cases from data sources) and PostCycle (stores the learned cases in the persistence layer).

3.4 jCOLIBRI Core

The Core is the most important component of the framework. It is in charge of maintaining the CBR configuration and executing the application [15]. When a user generates a CBR application template, s/he is generating the Java code that configures the Core components with the appropriate tasks, methods, data types and case structures. The Core is composed of CBRState (maintains the tasks and method configuration, CBRContext (contains the Case Base and working cases), Packages (manage the remaining components, such as similarity functions, case structure).

4 Applying jCOLIBRI Framework in Bulchino Catalogue

Often the users of the BULCHINO cannot describe clearly in their queries to the system the cultural object they are interested in. Besides, there are not enough information about the origin and designers of the cultural artifacts or at least - different speculations and interpretations are supported. Exploiting standard data base techniques both for storage and retrieval of information about a given cultural object, in this case, may produce unsatisfactory results, since standard query tools are not able to retrieve information that only partially match the user specification. With the development of CBR-based searching application, we expect to alleviate some of the above problems as well as to improve the efficiency of searching process in the Catalogue [34].

4.1 Our Approach

We intend to extend the search layer in the next version of BULCHINO catalogue by integrating a CBR-based application to implement the search task. With regards to the distributed catalogue architecture, the searching method could be applied locally to every single authority and work record (local queries) or globally to the whole catalogue (global queries). It should be considered, as well, that in BULCHINO domain the CBR-based application could be used for searching single objects, group of objects or integrate tourist routes (an ordered set of objects after some criteria) [35].

We propose an approach for carrying out the development process in a modular way to avoid potential failure in large systems. Initially, we consider applying the CBR techniques for searching only in one of the authorities in BULCHINO catalogue: "Personal and Corporate Name Authority". For the implementation of this approach we apply the jCOLIBRI Framework. To test the practical applicability and usefulness of the framework, we developed a simple application (PersonExample) to search single objects only in the „Person and Corporate Name Authority". Our goal with this example is to show how the framework jCOLIBRI could be adapted and applied for the configuration of this particular application and how the local queries to BULCHINO could be managed. As a next step we are going to extend the PersonExample Module, so that global queries to the whole catalogue are also manageable. Some of our investigations in this direction concern the implementation of ontology based CBR search in jCOLIBRI, where for computing the retrieval and adaptation processes DLs reasoning is applied. We consider this approach being an appropriate one in regard to the fact that we anticipate the development of ontologies to store the knowledge in the next version of BULCHINO.

4.2 PersonExample Module

This section exemplifies the use of jCOLIBRI to create the PersonExample application. It is developed via the jCOLIBRI GUI Tools that guide the instantiation process of the framework. The configuration of a CBR application using this interface implies the following steps:

- Defining case structure – simple plain cases, represented as a set of Attribute-Value Records;
- Defining the Case Base Organization – cases in PersonExample are stored in PlainText Case Base;
- Configuring the connector – the access to the cases from the Case Base is implemented by PlainText Connector;
- Configuring the behavior of the CBR application – PersonExample aims only to retrieve the most similar cases without reusing, revising or retaining them.

Case Structure. PersonExample uses cases with 9 attributes, which we defined on the basis of the object descriptions from "Personal and Corporate Name Authority. As sensitive attributes we chose Name, BirthDate, DeathDate, Natioality, Role, Gender, EarliestActive, LatestActive and Location. The case structure is presented in Figure 2, which shows attribute names together with their types, weights and corresponding local similarity functions.

Local similarity function. PersonExample is a simple application that does not take advantage of the extensions, provided by the framework. For the definition of case attributes we use only the core data types and local similarity functions.

Attribute weights. For the definition of the attribute weights we use heuristics, where we assume that the users of PersonExample will be mainly interested in the working periods and activity of a given historical person.

To create the case structure we use the graphical tools, provided by jCOLIBRI (Figure 2). The left panel displays the structure of the cases as a tree, and the right panel shows the property values of the selected attribute. Once defined, the case structure is stored in XML File.

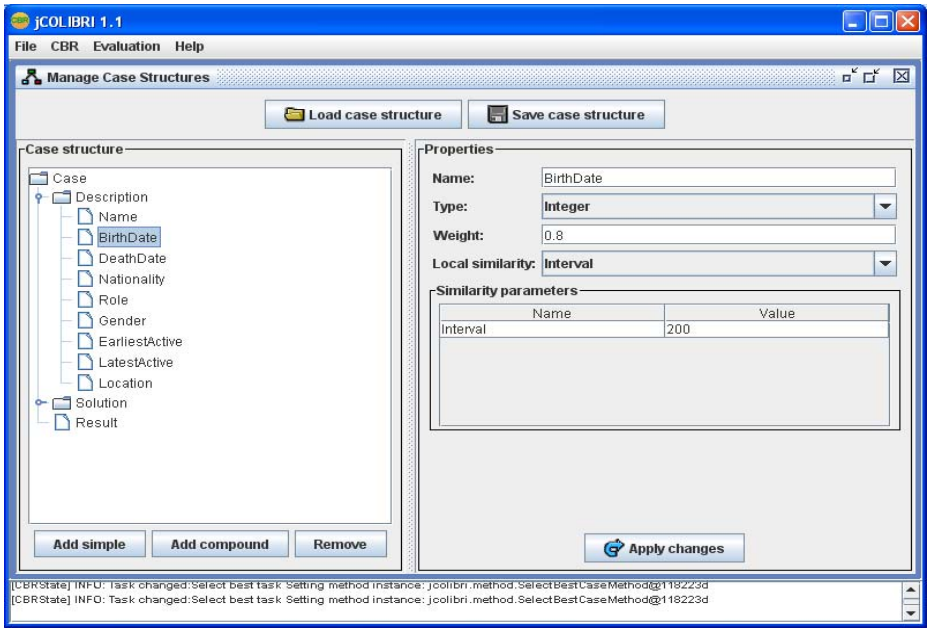

Fig.2. Defining the Case Structure

Case Base. A small section of the Case Base (Plain Text File), which contains the descriptions of three Bulgarian authors, is given in Figure.3.

```
#PersonExample PlainText CaseBase
#Columns are: caseId, Name, BirthDate, DeathDate, Nationality, Role,
Gender, EarliestActive, LatestActive, Location

case1, Vazov,   1850, 1921, Bulgarian, author, male,   1868, 1920, Sopot
case2, Botev,   1847, 1876, Bulgarian, poet,   male,   1867, 1876, Kalofer
case3, Bagrqna, 1893, 1991, Bulgarian, poet,   female, 1907, 1983, Sofia
```

Fig.3. PersonExample CaseBase

Connector. For our application we configured a PlainText Connector to work with cases stored in PersonExample PlainText CaseBase. The graphical interface helps mapping the case structure (Figure 2) with the columns from the PlainText file, storing the PersonExample Case Base (Figure 3). Like the case structure the connector configuration is saved in a XML file, generated by jCOLIBRI tools.

Task/Methods. After defining the case structure and configuring the connectors to store the Case Base of our application, we need to select the tasks and methods of our application. This simple example represents the ideal situation where we find every task and method needed for the development of our application.

Since PersonExample aims only to retrieve the most similar cases, without reusing, revising or retaining them, we associate a method only to Retrieve Task, leaving the other tasks unsolved.

Having configured the CBR application, we can test the task execution. Its first task is Obtain query task, which obtains the query that is going to be used to retrieve the most similar cases (Figure 4).

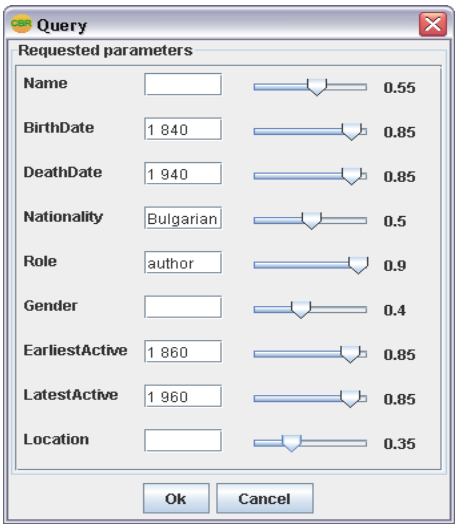

Fig. 4. Query Form for PersonExample

We assume that the users of BULCHINO Catalogue will be mainly interested in finding some information about a given person (persons), who lived and worked in a particular period. Since the user is not likely to know their names, gender or city (country) they lived in, s/he will be not able to fill in the corresponding fields in the query form.

The "Results" dialog (Figure 5) shows the execution log of PersonExample: PSMs results, Working cases and query. In the lower part of the window the description of the query parameters, as they are filled in by the user, is shown. The results of the comparison of the Query with all cases from the Case Base are presented in the upper part of the figure. They show the similarity value for each case, which is a result from the consecutive implementation of several functions. First the values of the local similarity functions (Interval, MaxString), assigned to every attribute of the case, are computed. Here the weights of the attributes are taken into consideration. Next a list of these values is returned (relationEvalList) and finally the global similarity between the cases is computed as an arithmetic average of this list of values (Average), i.e. in our case the sum of the values should be divided into 9 (the number of the attributes).

Working cases depend on the execution step. Since our application implements only the retrieve task, the working cases from the dialog will present the retrieved cases. It should be noted that we have to expect as a result from the system information about only one person (case) that will satisfy the user requirements to the greatest extent, i.e. the working cases will comprise only one retrieved case - the most similar case. As can be seen in the figure, for our example this is case 1, which has the highest similarity value – 0.62.

Note that the result (case 1) contains the Name, Gender and Location of the person, although these fields were left unfilled in the query form.

Once we have created and run our application, we generate a code template with most of the code that we need to run the PersonExample application. This template contains the tasks and methods invocation code and can be modified as needed.

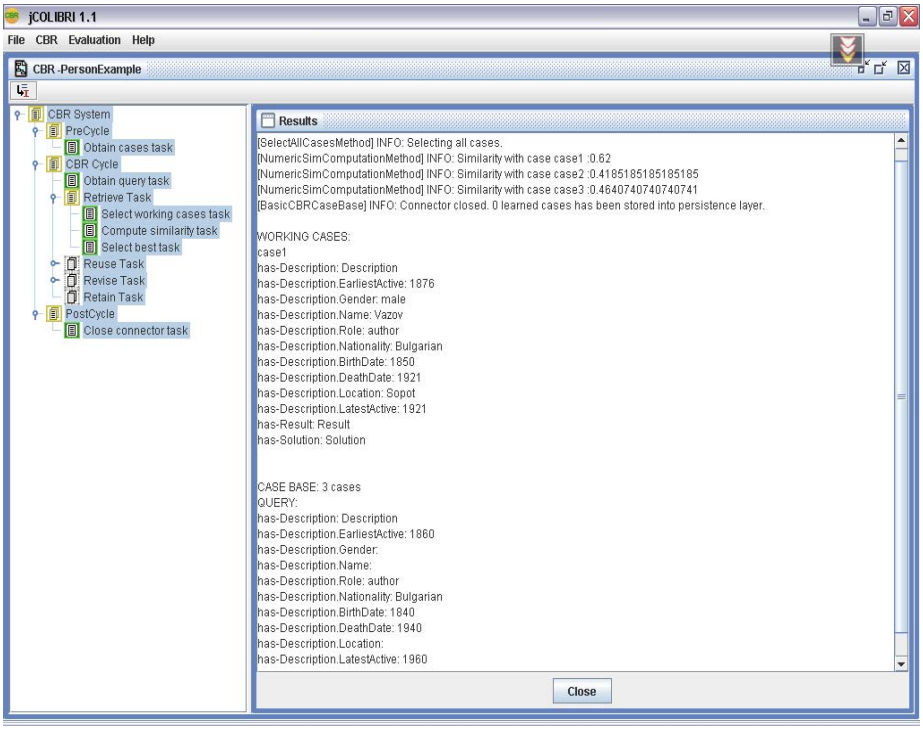

Fig. 5. Results

5 Conclusions and Future Work

In this paper we presented a Case Based Reasoning approach for extending the search layer in BULCHINO catalogue. To reach this goal we propose a modular design

process that is based on applying Object-Oriented Framework jCOLIBRI and will be implemented by stages. To show how the framework is adapted and applied in the particular domain BULCHINO, we developed the PersonExample module, whose main objective is to search objects initially only in "Personal and Corporate Name Authority".

The experimental results presented in this paper are not conclusive, but they do motivate further research in using and extending the CBR Framework jCOLIBRI. This work has raised several issues that can be talked and improved, one of which is the Case Base Management. The next step will be the development of Relational Data Base for storing the descriptions of persons from "Personal and Corporate Name Authority" and the configuration of JDBC Connector. We have been studying, as well, the synergies of jCOLIBRI with Case Retrieval Nets Model for structuring the Case Base.

Since jCOLIBRI is designed as an extensible framework and new elements could smoothly be integrated with the available infrastructure (as long as they follow framework design), another improvement will be the definition of our own components and their integration in the next version of PersonExample application. More specifically, we intend to extend the system core data types, provided by the framework, by defining and adding our own enumerated types (the field in the query form for the attribute with enumerated type will consist of a fixed set of constants). As a result, by filling in the query form the user will be able to choose among different values of a given attribute in a combo box. We believe that this way the user will be alleviated by defining the values of the listed attributes-s/he will only have to choose, and not think up one, what will reduce the imprecise queries.

Our ambitious goal is to develop an integrate module to reason with the knowledge from all components and authorities of BULCHINO Catalogue. Our efforts for achieving this goal are mainly targeted at the definition of the global similarity concept, where the local similarities will be established between the separate components of the catalogue

Acknowledgements. The authors wish to acknowledge the financial support of the Bulgarian Ministry of Education and Science and Research. Project Ref.No.MI-1502/2005.

References

1 Aamodt, A., Plaza, E., Case-base reasoning: Foundational issues, methodological variations, and system approaches, AI Communications, 7(1), 1994, 39-59

2 Antonio Sánchez, Juan A. Recio, Belén Díaz-Agudo, and Pedro González-Calero, Case structures in jCOLIBRI, Best Poster Award, Twenty-fith SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intelligence, AI 2005, Cambridge, UK. http://www.bcs-sgai.org/ai2005 (to date)

3 Asif Akram, Dharmesh Chohan, Xiao Dong Wang, Xiaobo Yang,Rob Allan, A Service Oriented Architecture for Portals Using Portlets, CCLRC e-Science Center, CCLRC Daresbury Laboratory Warrington WA4 4AD, UK, http://www.wi2.uni-trier.de/publications/PaperBled99.pdf, (to date)

4 Bello-Tomás, J.J., González-Calero, P.A., Díaz-Agudo, B., JColibri: an Object-Oriented Framework for Building CBR Systems, In Funk, P., González-Calero, P.A., (Eds.): Advances in Case-Based Reasoning, Procs. of the 7th European Conference on Case-Based Reasoning, ECCBR 2004, Lecture Notes in Artificial Intelligence, 3155, Springer, 2004.
5 Cataloging Cultural Objects, A guide to Describing Cultural Works and Their Images, 2005, http://www.vraweb.org/ccoweb/cco/index.html (to date)
6 C.Abasolo, E.Plaza, and J.-L.Arcos, Components for case-based reasoning systems, Lecture Notes in Computer Science, 2504, 2002.
7 Delphi Group, http://www.delphigroup.com/ (to date)
8 Díaz-Agudo, Belen, González-Calero, Pedro A., Gómez-Martín, Pedro Pablo and Gómez-Martín, Marco Antonio, On Developing a Distributed CBR Framework through Semantic Web Services, Proceedings of Workshop OWL: Experiences and Directions, at International Conference on Rules and Rule Markup Languages for the Semantic Web, 2005
9 Díaz-Agudo, B., González-Calero, P.A., CBROnto: a Task/Method ontology for CBR. In (Eds.): The 15th International FLAIRS 2002 Conference (Special Track on Case-Based Reasoning), AAAI Press, 2002.
10 Díaz-Agudo, B., González-Calero, P.A., An Architecture for Knowledge Intensive CBR Systems, In Blanzieri, E., Portinale, L., (Eds.): Advances in Case-Based Reasoning (Procs. of the 5th European Workshop on Case-Based Reasoning, EWCBR 2000), Lecture Notes in Artificial Intelligence, 1898, Springer, 2000.
11 Fernández-Conde, C.J., González-Calero, P.A.,Case-based Framework Instantiation, Procs. of the 15th International Conference on Software Engineering and Knowledge Engineering (SEKE 2003), ACM Press, 2003
12 Gómez-Albarrán, M., González-Calero, P.A., Díaz-Agudo, B. Fernández-Conde, C., Modelling the CBR Life Cycle Using Description Logics, In Althoff, K.D., Bergmann, R., Branting, L.K., (Eds.): Case-Based Reasoning Research and Development (Procs. of the 3rd International Conference on Case-Based Reasoning, ICCBR'99), Lecture Notes in Artificial Intelligence, 1650, Springer, 1999
13 González-Calero, P.A., Gómez-Albarran, M., Díaz-Agudo, B., Applying DLs for Retrieval in Case-Based Reasoning, Proc. of the 1999 International Workshop on Description Logics (DL'99).
14 J2EE, java.sun.com/j2ee/1.4/docs/tutorial/doc
15 jCOLIBRI Theoretical Foundations, http://gaia.fdi.ucm.es/projects/jcolibri/docs.html/ (to date)
16 Juan A. Recio-Garía, Antonio Sánchez, Belén Díaz-Agudo, and Pedro A. González-Calero, jCOLIBRI 1.0 in a nutshell. A software tool for designing CBR systems, In M. Petridis, editor, *Proccedings of the 10th UK Workshop on Case Based Reasoning*, pages 20-28. CMS Press, University of Greenwich, 2005.
17 Jiménez Díaz, G. and Gómez Albarrán, M, A Case-Based Approach for Teaching Frameworks, Proceedings of the PhD-Workshop at 18th European Conference on Object-Oriented Programming, 2004.
18 Jiménez Díaz, G., Gómez Albarrán, M.and González Calero, P. A, UnderFrame: Understanding Object-Oriented Frameworks Using a Case-Based Teaching Approach, Proceedings of the 18th European Conference on Object-Oriented Programming (Poster Session), 2004.
19 Lenz, M., Bartsch-Sporl, B., Burkhard, H., and Wess, S. Case-Based Reasoning Technology – From Foundation to Applications,Lecture Notes in Artificial Intelligence 1400, Springer Verlag, 1998
20 Lenz, M., Burkhard, H. D., Case retrieval Nets: Foundations, Properties, Implementation, and Results

21 Mario Lenz, Hans-Dieter Burkhard, Case Retrieval Nets: Basic Ideas and Extensions, Proceedings of the 20th Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence, September 17-19, 1996
22 M.Jaczynski and B.Trousse, CBR*Tools: An object-oriented framework for the design and the implementation of case-based reasoners, In Proceedings of the 6th German Workshop on Case-Based Reasoning, 1998
23 М. Трендафилова, Инфраструктура на интелигентен електронен каталог, Международна конференция "DeLC: решения и проблеми (DeLC: Solutions & Problems)", Пампорово, 2-4 Декември 2005
24 Orenge , http://www.ovitas.com/PDF/orengeWhitepaper.pdf (to date)
25 R.J.Allan, C.Awre, M.Baker, A.Fish, Portals and Portlets 2003, Technical Report UKeS-2004-06 Proc. NeSC Workshop 14-17/7/2003 (CCLRC and NeSC, 2004) http://www.nesc.ac.uk/technical_papers/UKeS-2004-06.pdf (to date)
26 Racer, http://www.racer-systems.com/ (to date)
27 Recio-García, J.A., Díaz-Agudo, B., An introductory user guide to jCOLIBRI 0.3, Technical Report No 144/2004. Dep. Sistemas Informáticos y Programación. Universidad Complutense de Madrid, Spain.
28 S.Bogaerts and D.Leake, IUCBRF: A Framework For Rapid And Modular Case-Based Reasoning System Development, Novembre, 2004, http://www.cs.indiana.edu/~sbogaert/CBR/ (to date)
29 Sascha Smitt, Ralph Bergmann, Applying Case-Based Reasoning Technology for Product Selection and Customization in Electronic Commerce Environments, Twelfth International Bled Electronic Commerce Conference Bled, Slovenia,June 7-9, 1999
30 S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, From CBT to e-Learning, Journal "Information Technologies and Control", No. 4/2005, Year III, Pp. 2-10, ISSN 1312-2622
31 S.Stoyanov, I.Popchev, Evolutionary Development of an Infrastructure Supporting the Transition from CBT to e-Learning, Cyberbetics and Information Technologies (CIT), 2/2006, Bulgarian Academy of Sciences, 101-114
32 Stoyanov S., I. Ganchev, M. O'Droma, I. Popchev, DeLC Project:: Current State and Future Development, In Proceedings 2nd International Wprkshop on eServices and eLearning, eds. S. Stojanov, M. O'Droma, A. Schmietendorf, September 28-29, 2004, University of Plovdiv, Smoljan, Bulgaria, Pp.113-130, ISBN 3-929757-76-1, A Publication of the University of Magdeburg
33 S. Stoyanov, M. Trendafilova, E-Catalogue "Culture-historical heritage and nature objects in Bulgaria", Conference "New education technologies", 16-17 May, 2003, Sofia, 289 – 298 (in Bulgarian)
34 S.Stojanov, M. Trendafilova, CBR-Search in Electronic Catalogs, In Proc. of the International Conference "Automatics and Informatics '03", vol.1, Pp.65-68, 6-8 October,2003, Sofia, Bulgaria. ISBN 954-9641-34-1
35 Stojanov, S., D. Chaushkova, M. Trendafilova, Applying Case Based Reasoning for Generation of Tourist Routes, In Workshop "Concurrency, Specification & Programming", vol.3: Multiagent Systems and Applications, Pp. 563- 575, Caputh, September 24-26 2004, Humboldt-University of Berlin, ISNN 0863-095X
36 Stojanov S., I. Popchev, D. Chaushkova, M. Trendafilova, A Case based reasoning Approach for Development of Intelligent Services, Journal "Information Technologies and Control", No. 3/2004, Year II, Pp. 31-34, ISSN 1312-2622
37 Visual resources association, http://www.vraweb.org/ (to date)