Universität Oldenburg
Abt. Wirtschaftsinformatik
Prof. Dr.-Ing. Norbert Gronau
Escherweg 2
26121 Oldenburg
Tel. (0441) 97 22 - 150
Fax (0441) 97 22 - 202
E-Mail: gronau@wi-ol.de
www.wi-ol.de

Arbeitsbericht WI - 2003 - 15

Norbert Gronau und Frank Laskowski:

Using Case-Based Reasoning to Improve Information Retrieval in Knowledge Management Systems

Zitierhinweis: Gronau, N., Laskowski, F.: Using Case-Based Reasoning to Improve Information Retrieval in Knowledge Management Systems. In: Menasalvas, E., Segovia, J., Szczepaniak, P. (Hrsg.): Advances in Web Intelligence. Proc. of the First International Atlantic Web Intellingence Conference, AWIC 2003, Madrid, May 2003, S. 94-102

Using Case-based Reasoning to Improve Information Retrieval in Knowledge Management Systems

Norbert Gronau[1], Frank Laskowski[2]

[1]Universität Oldenburg, Department of business information systems, Germany
gronau@wi-ol.de
[2]OFFIS e.V., Oldenburg, Germany
frank.Laskowski@offis.de

Abstract. Complementary to hypertext navigation, classic information retrieval is broadly used to find information on the World Wide Web and on Web-based systems. Among these there are most knowledge management systems, that typically integrate information accessible on the intranet of an organization or company, and present this information to support users with their knowledge-oriented tasks. In this paper we describe a Case-Based Reasoning component extending information retrieval in the context of KM systems. Our basic assumption is, that a user composing a search query simultaneously is describing a problem he or she seeks to solve. Hence our case-based reasoning component handles an information retrieval request as a description of a problem being part of a case. We will explain how this approach enables various benefits for intelligent query processing and a vast potential for synergies of the case based reasoning component and the embedding knowledge management system.

Keywords. Case-Based Reasoning, Information Retrieval, Knowledge Management, Software Architecture

1 Introduction

As part of the TO_KNOW project a case-based reasoning (CBR) component is being designed for integration into typical knowledge management (KM) systems. ([1]) While there are conceivably various possible applications of case-based reasoning in the context of knowledge management, our approach invokes the CBR algorithm on Information Retrieval (IR) requests.

Our basic assumption is, that a user composing a search query at the same time is describing a problem he or she seeks to solve. Hence our CBR component handles the IR request as a description of a problem being part of a case. Thinking straightforward a good solution for such a case would be a good search result, i.e. a set of links to relevant information with respect to the search query. We propose to include an alternative way of describing a solution: Given a search query that does not result in

the optimal set of available information, a good solution is an "improved" query, i.e. performing this query would deliver better support for solving the given problem.

In the following sections we explain how various benefits for intelligent query processing can be achieved by exchanging context information between the CBR component and the embedding KM system. We outline our approach to design the cases, the case base and the CBR algorithm accordingly.

The ideas presented here are linked for WWW environment for two reasons: Conceptually, the Web (and especially the emerging Semantic Web) delivers a good platform to design knowledge management systems, i.e. the software supporting knowledge management, e.g. [2], [3]. Practically, utilizing means such as artificial intelligence (AI) components to improve information retrieval makes sense for systems that provide access to a vast amount of text documents, mostly web-based systems or at least systems accessible via the Web.

2 Merging CBR into IR: The TO_KNOW Approach

In this section we describe the core part of our approach: conceptually plugging CBR into the processing of IR search queries. This extension facilitates improvements of precision, configuration and the user interface of search applications. Section 3 explains how this CBR-supported IR core integrates with the larger framework of a knowledge management system.

2.1 CBR and IR as Problem-Solving Methods

Case-based reasoning is well established as problem solving method (e.g. [4]). It is based on the concept of the "case" consisting of a description of a problem and its solution. A new problem is solved by retrieving and reusing similar experiences from the "case base". If revision indicates that a substantially new solution has to be provided for the new problem, both are retained as new case, i.e. added to the case base for future retrieval and reuse.

Information retrieval is a classic area of information and computer science (e.g. [5]). With the Web emerging and Web-technology establishing in organizations' and companies' intranets, document retrieval has become an omnipresent application, familiar to the everyday user like text processing. An information retrieval process starts with a user's need of information that is translated to a search query. This query is compared to descriptions of what information can be found in each of the accessible documents (or other kinds of entities). The result of the retrieval process is a set of references to those documents whose descriptors correspond to the query, and by this should be relevant to the user's request.

IR is often explained contrasting the "exactness" of data retrieval. But it can also be looked upon as a *method to support problem solving*: Especially when a problem needs a solution that is considerably new to the user an according search query can be interpreted as describing the users problem rather than describing the information the

user is looking for. The search engine delivers references to those contents addressing his or her problem, ideally enabling the user to solve it.

2.2 The Part of CBR in Query Processing

CBR usually is applied to a field of interest, providing solutions or help to find solutions for problems arising in that field, e.g. when deploying Neural Networks to optimize industrial production. The other way around CBR utilizes other techniques for retrieving, reusing, revising or retaining cases, e.g. TO_KNOW uses IR technology for retrieval of cases from the case base.

As already pointed out in the first section, form a user's perspective we deploy CBR and IR on the same level, i.e. both techniques are combined and applied to the same problems. Especially CBR is not used to support a strictly encircled part of the IR process, e.g. pre-translation of "natural language" queries, and on the other hand, it is not used to solve meta problems about IR's "how-tos". However TO_KNOW's CBR component is operating on a meta level in so far as it does not take the documents accessible to IR as cases, although they cover information about problems and their solutions. Instead the *application of information retrieval is recognized as a case*.

Hence the CBR algorithm accompanies the IR process, gathering experiences about it in it's own case base, and interfering with it by proposing alternative solutions to the user. To enable this amalgamation, the CBR component is invoked before and after the search engines operates on a single query. Thus the CBR component can examine and manipulate queries and results of the IR process. This includes changes to the user interface and access to the session data eventually bundling a sequence of several requests as one IR process. Both enable interactive revision of the solutions presented to the users.

Figure 1 illustrates the relationship described above.

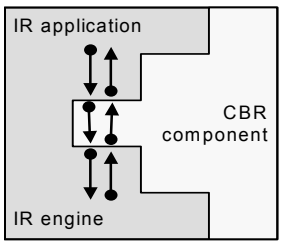

Fig. 1. CBR component plugging into the IR process

2.3 Basic Implications for Case Model and CBR Algorithm

The core of the case model is designed accordingly:

- The *problem description* may include various data (this will be outlined in the following section), but it is centered on the users problem description as it is effectively at hand as search query in an IR application.
- As well, in its central part the *description of the solution* is directly taken from the IR application: At first sight this could be a possibly large set of references to documents that change or even disappear over time and that represent information eventually outdated, especially regarding topics like business statistics or IT tools. As an alternative, we decided to focus on a query, the result of which meets the user's expectations better than the one initially describing the problem. (Remember that a case is made up from a problem that couldn't be handled "as ususal", i.e. where using the search engine did not deliver the desired information.)
  The case model does not ban the use of document references. But these point to data while search queries are coded on the level of information, i.e. categories and indices in terms of IR and the problem domain, as well.

Implications for the CBR algorithm that is usually split up into four phases: "retrieve", "reuse", "revise" and "retain", mainly affect the inner two phases, "reuse" and "revise".

The reuse phase has to adopt or recombine existing solutions making them applicable to new problems. This phase might deploy various AI techniques and has to utilize the "domain model", i.e. knowledge about IR, and especially about manipulation of search queries. This might include drawing conclusions from "query refinement histories", or involving the document descriptors from satisfactory result sets to construct improved queries. Work on this task is in progress in the TO_KNOW project, but results are not expected to affect the shape of the integration aspects described in this paper.

The reuse phase ends, either transparently manipulating the IR process, or presenting a choice of one or more alternative solutions, i.e. queries, to the user. (Think about "google" [6], presenting links to related index pages and a spell-checked version of a query, if applicable.) Right away, the revise phase begins.

The revise phase is decisive for retaining cases, i.e. building up the case base, which is the backbone of the CBR components added value.

Ideally the CBR component has means to automatically revise the quality of a solution. Unfortunately, if a search result fails to satisfy the user's needs and automatic revision on average is able to detect this failure, the CBR component would clearly outrival the IR engine regarding core IR capabilities. We suppose this is not likely to happen, assuming that a typical IR engine already makes use of the most important possibilities of how to automatically figure out, what information is contained in documents, respectively search queries. Nonetheless, automatic revision is useful, starting form examination of the size of the result set.

By choosing an alternative query presented after the reuse phase finished, or by ignoring this option, the user manually revises the proposed solutions.

Complementary the user can be asked how valuable he regards the result set that is currently displayed.

One logical revise phase might invoke retrieval and reuse of cases for several times, going along with the user refining his or her request. Typically the user interface of a search application should not be essentially altered to preserve usability. Hence some algorithm has to decide which queries belong logically together.

A new case is retained, if a logical revise phase requires more than one reuse phase or substantial manual refinement, to lead to a satisfactory result. The new case at least contains the initial query (as spontaneous description of the user's problem) and the query that the revise phase ended with.

Finally, after repeated positive revision of the same case, it can be marked for transparent application, i.e. the reuse phase will overwrite an initial query before it is forwarded to the IR engine, if the query matches the case. This is the most direct improvement to the IR application possible. Status of transparent applicability can be revoked anytime as response to negative revision.

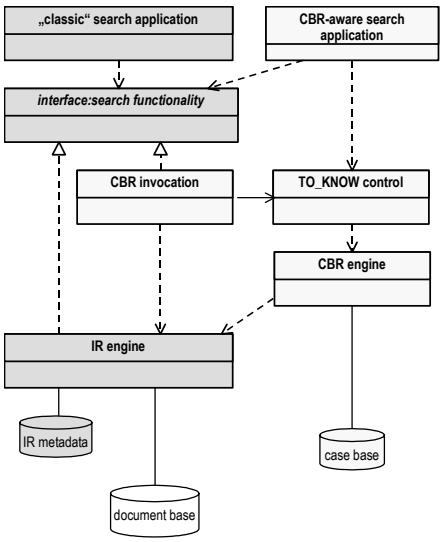

Fig. 2. Architecture of CBR-supported IR application

In [1] we presented several choices for *efficiently implementing and running* a combined system as described above. An outline of the architecture is shown in figure 2.

3 Integrating the CBR Component into a KM System

TO_KNOW's main focus is the area of knowledge management systems. In this section, we explain how the CBR component discussed above enables a search application to adapt to the information context provided by an embedding KM system.

3.1 A Practical Definition of KM systems

In the scope of this paper, we define a KM system as *a set of software* and the data accessed by this software, *deployed* in an organization or a company *to implement knowledge management strategies*.

KM systems range form a Web-Server, providing a single point of access to important document collections, to fully fledged systems integrating many tools, and covering almost all building blocks from the reference architecture shown in figure 3. ([7])

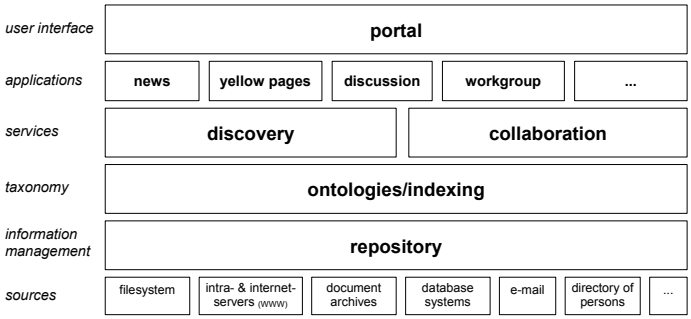

Fig. 3. Reference architecture of a KM system

The third layer in figure 3 provides collaboration and discovery services. Document retrieval is the basic application built on the discovery services, and hence a kind of IR engine should be present in any non-trivial KM system.

As a consequence, improving IR quality by itself already leads to an advanced utilization of the knowledge base.

(We are aware that some definitions intentionally avoid focusing on IT support. We do not suggest to ignore these definitions. However, the issues discussed here regard basic parts of electronic support, and should not be affected by preferring or rejecting any of those technically abstract definitions.)

3.2 Adapting to the System Context

A fully fledged KM system might hold many different kinds of metadata, which potentially could be of use to the CBR component described in section 2. Neither the TO_KNOW CBR component, nor any service, application or component that is being added to a reasonably complex KM system can individually adopt to each of the subsystems from which such metadata originates. Hence components rely on the KM system's information management and/or taxonomy layer to provide a kind of centralized and homogenous access to its metadata.

For the CBR component, the KM system can be a source of information about the *context in which a problem is embedded*: On one hand information can be provided about the search queries, on the other hand about the users. As a consequence, on invocation the CBR component first retrieves the according information from the KM system, extending the problem's context.

Basic elements of a search query are significant terms. As the context of a term we define the most associated terms, according to the KM system's taxonomy. When the CBR component is embedded into a KM system, an appropriate adaptor has to be implemented delivering the system-specific context of a term on request. This context information can be exploited if the organization's taxonomy (e.g. administered by a knowledge engineer) diverges from the taxonomy of the IR engine (e.g. a general taxonomy for the language of the organization's country). Taking the system-specific context of a term into account has potential to improve the recall for retrieval of cases and documents.

The context of a user is defined as a simplified user's profile, containing a list name/value-pairs and a list of groups, which the user belongs to. The context of a user group is defined analogous. To retrieve the according information from the KM system, a second context adaptor has to be implemented. Importing user profiles improves all parts of the CBR component that perform collaborative filtering. The utilization of user context generally offers the potential to improve the precision.

Figure 4 updates figure 3 including the modules needed for integration of the CBR/IR component with the KM system. ([1] describes other associations and components technically facilitating the integration.)

3.3 Benefits for the KM System

The improvements outlined in the preceding section lead to a better utilization of the knowledge base as already mentioned. In addition, "feedback" from the IR/CBR component to the KM system is possible, as the following examples show:

- Queries that cannot be answered satisfactory (neither by the IR engine nor by the CBR system) are stored as negatively revises cases. This kind of cases indicates "information deficiencies" in the knowledge base. A report can help a knowledge engineer to obtain the needed information and add it to the knowledge base.
- Data from the case base can suggest checking whether terminology in the KM system matches users' conventions. E.g., a check is indicated when often cases are

applied, that "translate" a term from the initial search query into another term from the "official company's list of keywords".

- A knowledge engineer can manually revise cases. E.g., cases, which are often applied, can be simplified and hence further optimized for application. Or such cases can be given "titles" or informal descriptions that are displayed like "FAQs" when the solution of the case is proposed to a user.
- The search log of the IR engine might be evaluated for automatic profile generation in the KM system. Additionally logging user interaction from the revise phase provides information that cannot (at least not easily) be extracted from a "classical" search-log.

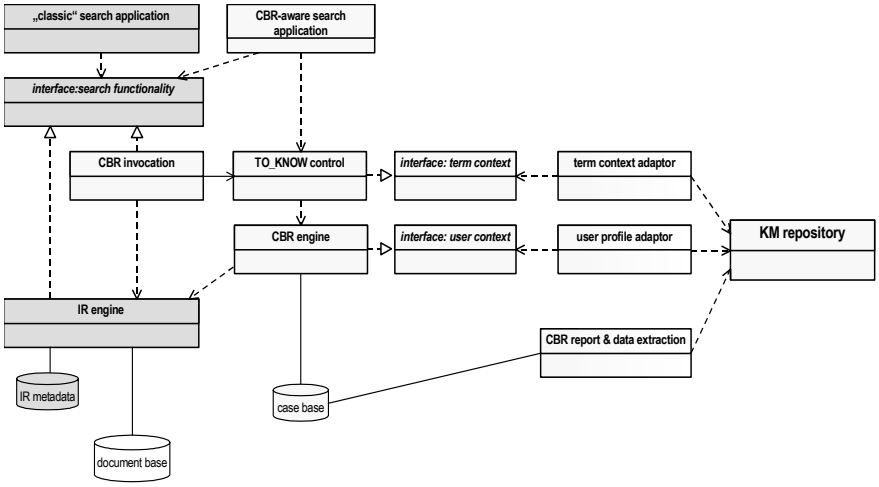

Fig. 4. CBR/IR component integration into KM system

4 Conclusions & Future Work

In this paper we described an approach to use case based reasoning to improve quality and integration of a standard information retrieval engine in a knowledge management system, presenting intermediate results from the TO_KNOW project.

The first part of the paper shows how the IR process can be enhanced by the CBR method, outlining the case model and the adoption of the CBR algorithm. The second part presents the basics of the integration, achieved by exchange of metadata between the CBR component and the KM system. A central concept is to transform the potentially heterogeneous metadata from the KM system to context information, which can be utilized by the CBR component to further improve its ability to support the IR engine, and to deliver rich feedback about the use of the knowledge base.

A prototype of the CBR component described here is currently developed as part of the TO_KNOW project. Actually running the component will enable us to configure and optimize the basic combination of IR and CBR, and to learn more about the potential of different context information. Designing a component (instead of a stand-alone system) raised the question whether a framework could be designed that enables efficient integration of a variety of basic and advanced components alike, and finally lead us to plan the project K_SERVICES ([8]).

Literature

[1] Gronau, N.; Laskowski, F.: An architecture for integrating CBR components into KM systems. In: Mirjam Minor, Steffen Staab (Hrsg.): Proceedings of the 1st German Workshop on Experience Management. Bonner Köllen Verlag, 2002, S. 92-97

[2] http://www.hyperwave.com/

[3] http://www.opentext.com/livelink/

[4] Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. AI communications 7 (1994), 1, S. 35-39

[5] van Rijsbergen, C.J.: Information Retrieval. Second Edition, Butterworths, London, 1979

[6] http://www.google.com

[7] Gronau, N.: A Procedure Model for Evaluating Knowledge Management Systems. In: Arabnia, H.R.; Youngsong, M.; Prasad, B. (Hrsg.): Proceedings of the International Conference on Information and Knowledge Engineering 2002. CSREA Press, Athens, Georgia, S. 78-83

[8] Gronau, N.; Laskowski, F.: K_SERVICES: From State-of-the-Art Components to Next Generation Distributed KM Systems. IRMA'03 accepted paper, Philadelphia 18.5.-21-5.2003

[9] Aha, D.W.; Muñoz-Avila, H. (Hrsg.): Exploring Synergies of Knowledge Management and Case-Based Reasoning: Papers from the AAAI 1999 Workshop. Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, 1999

[10] Berners-Lee, T., Connolly, D.: Hypertext Markup Language - 2.0. http://www.rfc-editor.org/rfc/rfc1866.txt

[11] Berners-Lee, T.: Information Management: A Proposal. http://www.w3.org/History/1989/proposal.html

[12] Berners-Lee, T.; Swick, R.: The Semantic Web. http://www.w3.org/2000/Talks/0516-sweb-tbl/all

[13] Choo,Chun wei: The Knowing Organization - How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions. Oxford University Press, 1998

[14] Rissland, E.L.; Daniels, J.J.: Using CBR to Drive IR. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 400-407, 1995

[15] Wilson, D.; Bradshaw, S.: CBR Textuality. In Proceedings of the Fourth UK Case-Based Reasoning Workshop, 1999