Towards a Recommender System for Statistical Research Data

Daniel Bahls[1], Guido Scherp[1,2],
Klaus Tochtermann[1], and Wilhelm Hasselbring[2]

[1] Leibniz Information Centre for Economics (ZBW), Kiel, Germany
[2] Software Engineering Group, Kiel University, Germany

Abstract. To effectively promote the exchange of scientific data, retrieval services are required to suit the needs of the research community. A large amount of research in the field of economics is based on statistical data, which is often drawn from external sources like data agencies, statistical offices or affiliated institutes. Since producing such data for a particular research question is expensive in time and money—if possible at all— research activities are often influenced by the availability of suitable data. Researchers choose or adjust their questions, so that the empirical foundation to support their results is given. As a consequence, researchers look out and poll for newly available data in all sorts of directions due to a lacking information infrastructure for this domain. This circumstance and a recent report from the High Level Expert Group on Scientific Data motivate recommendation and notification services for research data sets.
In this paper, we elaborate on a case-based recommender system for statistical data, which allows for precise query specification. We discuss required similarity measures on the basis of cross-domain code lists and propose a system architecture. To address the problem of continuous polling, we elaborate on a notification service to inform researchers on newly avaible data sets based on their personal request.

Keywords: Research Data Management, Semantic Digital Data Library, Linked Data, Statistics, Recommender Systems, Case-Based Reasoning

1 Introduction

At present, efforts are being made to pick up research data as bibliographic artifacts for re-use, transparency and citation. Data publications will be submitted to digital archives and registered in central catalogs which lays the ground for information services to support the scientific community in finding relevant data. Since every scientific discipline brings its own challenges in this endeavor, specific solutions are required, so that valuable, and hence accepted, services can be offered to the scientific community [1]. The High Level Expert Group on Scientific Data recommends to provide data recommendation services that suggest relevant research data to the individual scientist [2]. This appears to be particularly applicable in the domain of economics where research activity is influenced by the availability of statistical research data sets.[3] Researchers adjust to what data is available and adapt research questions so that the empirical foundation can be given.

As a consequence, researchers look out and poll for newly available data in all sorts of directions due to a lacking information infrastructure for this domain. They exchange news on newly available data at conferences, at meetings or simply at lunch time or during coffee-break. They also revisit websites of data agencies, repositories and familiar institutes to run their personal portfolio of keyword-based queries on regular web search engine interfaces—trying to express their request for specific data sets. Although best practice at present, this strategy seems effortful and insufficient in returning a complete list of relevant data sets. This picture was shared with us in interviews we have conducted with researchers in economics.

Having catalogs of registered research data sets puts us in a good position to address the above problem and develop well-conceived search tools and services for our scientific community. Besides the fact that the catalog itself lays the ground for a more organized search, this paper tries to address the following two aspects of the identified problem:

1. Phrasing several queries with different keywords and filters of all kinds to cover the range of relevant data sets.
2. Continuous polling at regular time intervals.

The remainder of the paper is structured as follows. We review related work and decide on our approach in Section 2. Section 3 concludes the findings and formulates the functional requirements for our proposed system. Since we follow a case-based recommendation approach, we examine case base and case structure in Section 4 and elaborate on a similarity measure design on the basis of common code lists subsequently in Section 5. We propose a system architecture in Section 7. Finally, we close with conclusions and outlook in Section 8.

2 Related Work

In the domain of statistical research data, one main difficulty is given by data protection and usage rights, so that uploading entire data collections to an independent repository causes legal problems. This is one of several reasons why we have decided to use Semantic Web technologies for the data model, which are strong in fine-grained referencing and in dealing with distributed data sources. In particular, we use the RDF Data Cube Vocabulary (QB), which integrates the SDMX standard[4] and is increasingly recognized in the domain of statistics [3]

[3] A large amount of research in the field of economics is based on statistical data, which is often drawn from external sources like data agencies, statistical offices or affiliated institutes. Producing such data for a particular research question is expensive in time and money—if possible at all.

[4] Statistical Data and Metadata eXchange Language `http://sdmx.org/`

[4] [5] [6]. A more detailed argumentation and an overall vision for our research is given in [7].

There are several different types of recommender systems for which a comprehensive overview can be found at [8]. Especially in e-commerce environments, *collaborative filtering* has established as a common technique. Online stores like amazon[5] recommend products on the basis of similar user profiles, following the idea that one might be interested in the products that other users with similar interest patterns have purchased. While this technique can be applied irrespective of the kind of items operated on, it demands large amounts of usage data from a sufficient number of users in order to produce meaningful recommendations. This initial overhead is known as the *cold-start problem* and usually requires user acceptance long before the value of item recommendation can be experienced.

Another technique makes use of the items' digital content[6] which we refer to as content-based recommendation systems [8]. Typically, items are mapped onto a vector space model where distances between them can be calculated using common mathematical means. This technique has established particularly in the context of textual items, where means like[7] are frequently used. However, this approach again depends on an initial set of usage data. While collaborative filtering compares patterns among user profiles, content-based retrieval is based on usage history of a single user and suggests similar items according to what she or he found useful or not useful earlier.

A third system type is based on background knowledge and calculates recommendations merely on the basis of a given user query and domain-specific preference knowledge encoded in the form of rules or specifically designed similarity measures. The approach therefore does not build on usage data at all and thus is not affected by the cold-start problem. Since usage data on statistical research data sets is not easily available to us and difficult to acquire in sufficient quantity, we find this approach most suitable for our domain. The amount of statistical research data is tremendous, and the amount of usage data required scales accordingly if we plan to include all available data sets for recommendation. In addition, recommending data sets that are similar to the ones used previously may not be helpful in the scientific domain, where researchers often work on various projects simultaneously or change their research area when moving to another organization. The above described systems tend to recommend older items, because usage statistics on newer ones build up slowly[8]. While these drawbacks do not apply for knowledge-based recommenders, another advantage is their strength in explaining results, so that users can understand why a particular recommendation was considered relevant. Furthermore, a lot of background knowledge for statistical data is available and has even been formalized

[5] urlhttp://www.amazon.com

[6] be it metadata, a textual description or the digital item itself like for example in document retrieval scenarios

[7] Term Frequency - Inverse Document Frequency

[8] also known as the time-span problem

in SDMX[9], DDI[10], code lists and the RDF Data Cube Vocabulary (QB), which also encourages a knowledge-based approach.

Knowledge-based recommender systems are typically constraint-based or case-based [8]. While the former uses rule sets and constraint resolvers to produce recommendations, the case-based approach uses specifically designed similarity measures that shall reflect the user's understanding of utility [9]. Eventually, we have chosen to follow a case-based approach on the grounds of positive experiences in earlier projects. As a consequence, a research data set is considered and may be referred to as a *case* in the following. Cases in general can be represented textually, as a feature vector, or as a structured representation [10]. The cases according to our RDF-based data model are already in structured shape, which gives reason to choose a structured CBR[11] approach over a textual, feature-based or other.

Common data repositories do not yet offer recommendation features and focus on providing full text search interfaces and filtering features. Text search algorithms often yield scores that allow for relevance ranking and are applied on textual fields of the respective underlying metadata model. Search criteria given for the more structured part of the model[12] are usually filtered on, meaning that all unmatched items are removed from the ranking [11]. A typical implementation imposes this rather technical and limited viewpoint on the user who switches back and forth modifying query phrase and parameters to cover the whole spectrum of possibly interesting search results, simply to deal with the limitations of such rigid interface[13]. It is to say that these issues are difficult to overcome, and most retrieval algorithms incorporate stemming, query expansion and other strategies while targeting a yet simple interface which certainly is another important design goal. Our aim is to get a clear picture of the user needs first which needs no further editing once specified clearly. Every item that matches the query entirely would be considered a perfect match, and therefore the approach performs like the common ones. In addition, however, the system should be able to find near matches and offer further means of knowledge discovery, which is a more high-level approach in the first place.

3 Functional Requirements

To address research objective 1, the system must provide an interface that allows for precise specification of a data request, enabling researchers to pinpoint to the perfect data set regardless of whether such data exists. This can be done on the basis of the RDF Data Cube Vocabulary which provides a wide range of predicates and attributes to formulate precise queries. The system further needs

[9] Statistical Data and Metadata eXchange Language `http://sdmx.org/`

[10] Data Documentation Initiative `http://www.ddialliance.org/`

[11] Case-based reasoning—or case-based recommending in our case

[12] e.g. creation date, size, country of origin or other domain-specific fields

[13] rephrasing query terms, resetting date ranges, size parameters, geo location and other

to know what aspects of the query are of greater, and what aspects are of lesser significance, which can be handled with the help of user-defined weights [12].

The second objective can be achieved through a notification service that sends out updates on newly available data to the individual user whenever estimated relevant.

An understanding of utility must be encoded in the system, so that data not perfectly matching the user's description can be estimated whether it yet may interest the user. In case-based recommender systems, such knowledge is encoded in similarity measures that are used to determine an estimated degree of utility of a particular case under a given query. Such measures must be designed carefully and must not make assumptions on user preferences where no foundation is given. Case-based recommenders in principle can be applied for our research objectives. However, the value of this approach depends on the question whether meaningful similarity measures can be implemented, which will be investigated in Section 5.

4 Case Structure

CL_OBS_STATUS	Status of an observation with respect events such as the ones reflected in the codes composing the code list.
CL_CONF_STATUS	Coded information about the sensitivity and confidentiality status of the data.
CL_DECIMALS	Gives information on the number of decimal digits used in the data.
CL_FREQ	Indicates the "frequency" of the data (e.g. monthly) and, thus, indirectly, also implying the type of "time reference" that could be used for identifying the data with respect time.
CL_SEX	Provides information on the gender.
CL_TIME_FORMAT	Time Format as written in the SDMX-EDI and SDMX-ML messages; these codes (based on the ISO 8601 standard) indicate the type of time references used in the data. The numeric codes below (203, 102,,702) are used only in the SDMX-EDI messages; and the alphanumeric codes (P1DPT1M) only in the SDMX-ML messages.
CL_UNIT_MULT	Unit Multiplier; indicates the magnitude in the units of measurements.
CL_AREA	Reference area and/or counterpart area; geographical areas, defined as areas included within the borders of a country, region, group of countries, etc.
CL_CURRENCY	Provides code values for currencies.

Table 1. Cross-domain code lists as provided by the SDMX consortium

Since we use the RDF Data Cube Vocabulary to organize statistical research data, the number of available attributes to describe research data sets is very large, also because RDF-based descriptions are per se extensible, which might be made use of when dealing with long tail research data of individual researchers. Hence, we only review some of the common attributes in order to assess the value of this approach. Table 1 gives an overview to the Cross-Domain Code lists issued by the SDMX consortium [13] [14]. First of all, we need to clarify the notion of a case and how to map the RDF data to a case base. Figure 1 illustrates the structure of a case, and where the SDMX code list attributes are located.

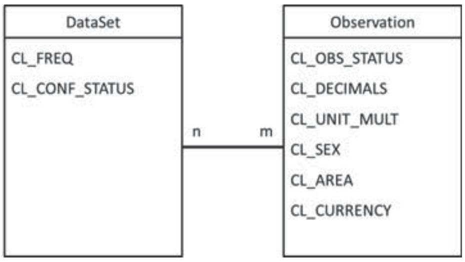

Fig. 1. The case structure

This structured representation suggests to apply the *local-global-principle*, which is an established paradigm in the CBR domain [9]. Local similarity measures are used to determine similarities on attribute level, while global similarity measures aggregate the resulting values on object level. There are two types of objects: DataSet and Observation, and thus, two global similarity measures are needed. Instances of DataSet are the items to be retrieved or recommended, while instances of Observation make for a large portion of its actual content. Because of the *n,m* relation between DataSet and Observation, we need measures for dealing with multiple values [15]. In the following, we write *sim(q,c)* to denote the similarity function of a query value *q* and a case value *c*, whereas both variables *q* and *c* are elements of the respective attribute's value range.

5 Similarity Measures

5.1 Local Similarity Measures

CL_CURRENCY specifies the currency used in a data set. If the user explicitly queries for data sets with Euro as a currency, only such data sets should be considered suitable. Suggesting that a data set using USD would be more useful than one using CHF has no basis.[14] Thus, the similarity measure for such type should be totally uninformed:

$$sim(q,c) = \begin{cases} 1, & \text{if q=c} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

We find a different situtation for the area code, where groups of countries and regions build up taxonomies. Whether data about Bavaria is useful when Germany was specified in the query depends on interpretation: It may be interpreted as "data about any region in Germany is fine" or "data about Germany on the national level is needed". Sophisticated user interfaces would be required for disambiguation, and we rather try to bypass this problem and approach a more vague but generic measure. This is supported by the consideration that even with a more precise query, the utility of a data set on other regions still remains hard to assess in general. When a data set on Bavaria is requested and a data set on Brandenburg is given, one may argue that merely the data on Bavaria represents the population a researcher wants to do research on, and any other are simply unsuitable. In contrast, one may argue as well that both regions are siblings in the sense that Germany is the subsuming parent, and a similarity value above zero appears resonable, as the data set may still reflect some of the features the researcher is after, while a data set on Idaho (USA) may not be suitable anymore, and yet another dataset on Chengdu (China) cannot be used at all. Several techniques are available for implementing a taxonomy-based similarity measure. One generic option is given to calculate a value based on the length of the shortest path. A more specific option depends on the actual query semantics and needs further consideration and discussion.

Other codes have ordered range sets. The CL_DECIMALS code list denotes the number of decimals used in the data. It seems reasonable to assume that any data providing a higher number of decimals suits just as well as the data queried for, since numbers can always be rounded. In contrast, a smaller number of decimals than requested should be assumed as less suitable, as it means a lack in precision. And since the degree of precision decreases proportionally with the difference between case and query value, it suggests a typical *more is better* similarity measure:

$$sim(q,c) = \begin{cases} 1, & \text{if } q \leq c \\ \frac{q-c}{d}, & \text{otherwise} \end{cases} \tag{2}$$

where d denotes the maximal difference between query and case, which is ten in this case.

A similar case is found for the code list CL_FREQ. Quarterly data can be aggregated from monthly or daily data. But if monthly data is requested, and quarterly data is given, the request is not perfectly met. Such data might yet be more useful than yearly data, so that a similar measure like the above could be reasonable. While cl_decimals was based on numbers, cl_freq is symbolic.

[14] U.S. Dollars (USD), Swiss franc (CHF)

Therefore, we could define an order and map frequency symbols to integers, so that a similar function as the above can be applied.

For the free text fields as listed in [13], the measure should be based on common techniques like TF-IDF[15] or n-gram. However, it must be ensured that the value is normalized, so that resulting similarity values can be set in relation to the ones of other attributes when aggregating in the global measure.

5.2 Aggregation of Similarity Values

A so-called global similarity measure is used to aggregate the results from the attribute-level similarity calculations. As we are still in a stage of considerations, there is no point in arguing whether weighted means, Euclidean or other types of aggregation is the right method to choose. We state, however, that the measure should enable user-defined weights in the query, as it allows the researcher to emphasize on the one or other parameter.

To complete the similarity measure, we further need to specify how multiple values are dealt with. For example, the researcher may request data on the geographic locations France, Germany and United Kingdom. The utility of a data set that represents the populations of England and France may then be calculated by finding best partners for every requested country[16] and build minimum, maximum or average for the overall similarity value of the geographic attribute. Which strategy to choose depends on the particular attribute and should be examined carefully in evaluation with end users [15].

5.3 Undefined values

There are some special cases we need to consider. When a query specifies a value for a particular attribute, for which the case compared does not provide any value, the measure must yield some value as well. For instance, if `male` is specified for `gender` in the query, and the attribute value is not given in the case, utility should be considered zero, because the user explicitly stated that represented population should be male. It appears reasonable to take this as the default measure. However, if the user specifies `free` for confidentiality status, and the case does not provide any information in this regard[17], the data set is not necessarily unsuitable. A reasonable way to deal with this issue could be to simply ignore this attribute in the global similarity function.

Another special situation occurs when a researcher requests data that contains values of some currency, but she does not want to specify more precisely on it. She is certain that monetary values must be part of the data while the currency unit itself is subordinate. One way to cater for this is to introduce a special value * and let sim(*,c)=1 for any case c.

[15] term frequencyinverse document frequency

[16] best with respect to the local similarity measure

[17] due to incomplete annotation

6 Notification Service

Due to the impression that empirical research is quite data-driven, and researchers need to conintuously look out for new data sets in order to stay up-to-date, we want to make some considerations on a notification service[18]. As our approach was to capture the researcher's request for data in high precision, we are in a position to test incoming data sets for relevance and send out messages[19]. One strategy in this regard would be to notify about every data set that meets a user-defined similarity threshold. From experience, however, similarity values tend to accumulate in a particular range, which is highly dependent on the similarity measure design and the respective user query, and thus, it may be difficult to be provide a specific threshold value. In that sense, the values calculated with the help of the similarity measure should rather be regarded as scores that give means for a ranking. To bypass this problem, we suggest to send out such rankings of newly registered data sets on a regular basis as per user settings. The user may then take a closer look at the top matches and estimate their utility individually.

7 Proposed Architecture

The recommender component is considered part of a larger digital archive system that manages statistical research data. Figure 2 gives an illustration of the entire system architecture, where three main components depict the relevant parts of the recommender system.

The case retrieval engine requires access to the data base that contains the data sets and the similarity measures which should be contained in a separate data base as to allow for independent editing whenever administrative review is needed. The archived research data usually is maintained in its specific data format, which in our case is based on RDF[20]. If the retrieval engine is implemented using sequential similarity calculation, the data repository can be accessed as a case base directly, since no further indexing is required. This, however, leads to long computation times in case of a large case base. For more efficient retrieval, more optimized methods like Case Retrieval Nets [16] should be considered, which builds its own data structure from case base and similarity measures. Therefore, the recommender component needs to be notified whenever there are updates on the data repository or similarity measures.

The notification service needs access to the users' notification queries and their e-mail addresses, which are stored in the user preferences data base. The component should be notified whenever updates occur on the data repository, so that new data sets can be tested for relevance immediately.

[18] cf. Google Alerts

[19] whenever a relevant candidate is detected or collated, per e-mail, twitter or other channel

[20] Resource Description Framework

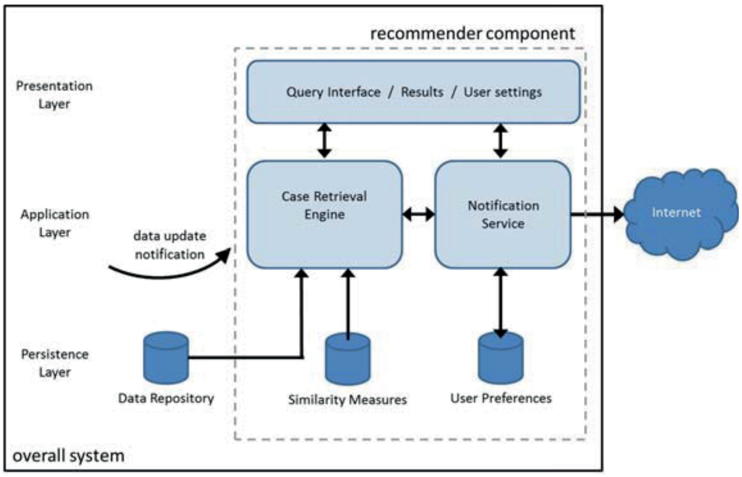

Fig. 2. System architecture

A detailled discussion on the user interface exceeds the scope of this paper. However, it must provide for query specification, display of results and configuration of user settings regarding the notification service as discussed in Section 6. Furthermore, we suggest to integrate explanation features in order to provide transparency to the user on how results were retrieved. A generic interface design and implementation can be found at [15], and some idea on how a query interface particularly designed for statistical research data using the RDF Data Cube Vocabulary can be found at [7].

8 Conclusion and Outlook

We have examined some of the common code lists for statistical data with respect to their specification and found indicators that motivate a particular similarity measure design. For some of them we were able to reason a specific design, whereas other code lists are difficult to make assertions on and suggest rather uninformed measures. Eventually, a final assessment on utility of a particular data set can only be done by the researcher. A similarity measure can only approximate a common sense of utility [9]. It easily fails due to limited query expressiveness and inability to interpret its actual semantics and the actual user needs. One option to overcome this problem is to allow for customization and personalization of similarity measures. Whenever a user is presented with unexpected results, an explanation may be given and the user may give feedback on the similarity measure. Since structural CBR systems in general are easily equipped with explanation support and customization of similarity measures [17] [18], some of the open similarity design questions could be answered by the individual user within a particular research context. However, ordinary measures for dealing with multiple values and the application of user-defined weights in the aggregating function enable a more gradual scaling of retrieval results with respect to user needs.

Another common practice in empirical research is the use of proxy variables, where some data highly correlates with other. Such information could be useful for recommending relevant data. A similarity measure could again be extended to make use of such relations if represented in the data model.

The proposed recommender system is based on the RDF Data Cube Vocabulary. The user is therefore in a position to specify precisely on the kind of data needed, and the system has the required means to assess suitability of available data sets. In addition, provided the measure reflects a reasonable understanding of utility, the introduced notification service helps researchers keep up to date and thus, both research goals defined in Listing 1 were met. Nevertheless, an evaluation is yet to be carried out, which is subject of future work. With further progress on a research data management infrastructure and the continuing exchange with the scientific community, we will get a clearer picture on the applicability of this approach.

Eventually, a prototype is needed in order to gain feedback from the research community we are addressing, which we consider implementing as we proceed with the reasearch on a data management infrastructure.

References

1. Feijen, M.: What researchers want - a literature study of researchers' requirements with respect to storage and access to research data (February 2011)
2. Wood, J., Andersson, T., Bachem, A., Best, C., Genova, F., Lopez, D.R., Los, W., Marinucci, M., Romary, L., Van de Sompel, H., Vigen, J., Wittenburg, P., Giaretta, D.: Riding the wave: How Europe can gain from the rising tide of scientific data. European Union (2010) Final report of the High Level Expert Group on Scientific Data: A submission to the European Commission.
3. Gottron, T., Hachenberg, C., Harth, A., Zapilko, B.: Towards a semantic data library for the social sciences. In: SDA'11: Proceedings of the International Workshop on Semantic DigitalArchives. (2011) in Preparation.
4. Cyganiak, R., Field, S., Gregory, A., Halb, W., Tennison, J.: Semantic statistics: Bringing together sdmx and scovo. In Bizer, C., Heath, T., Berners-Lee, T., Hausenblas, M., eds.: LDOW. Volume 628 of CEUR Workshop Proceedings., CEUR-WS.org (2010)
5. Miloevi, U., Janev, V., Spasi, M., Milojkovi, J., Vrane, S.: Publishing statistical data as linked open data. In: Proceedings of the 2nd International Conference on Information Society Technology, Information Society of the Republic of Serbia (2012)
6. Halb, W., Raimond, Y., Hausenblas, M.: Building Linked Data For Both Humans and Machines. In: WWW 2008 Workshop: Linked Data on the Web (LDOW2008), Beijing, China (2008)
7. Bahls, D., Tochtermann, K.: Addressing the long tail in empirical research data management. In: 12th International Conference on Knowledge Management (I-KNOW '12), Graz, Austria, ACM (2012) in Preparation.

8. Burke, R.: Recommender systems: An introduction, by dietmar jannach, markus zanker, alexander felfernig, and gerhard friedrich. International Journal of Human-Computer Interaction **28**(1) (2012) 72–73
9. Bergmann, R., Richter, M.M., Schmitt, S., Stahl, A., Vollrath, I.: Utility-oriented matching: A new research direction for case-based reasoning. In: In professionelles Wissensmanagement: Erfahrungen und Visionen. Proceedings of the 1st Conference on Professional Knowledge Management. Shaker. (2001) 264–274
10. Bergmann, R., Kolodner, J., Plaza, E.: Representation in case-based reasoning. Knowl. Eng. Rev. **20**(3) (September 2005) 209–213
11. Bridge, D., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Engineering Review **20** (September 2005) 315–320
12. Richter, M.M.: Case based reasoning and the search for knowledge. In: Proceedings of the 7th industrial conference on Advances in data mining: theoretical aspects and applications. ICDM'07, Berlin, Heidelberg, Springer-Verlag (2007) 1–14
13. Guidelines, S.C.o.: Annex 1: cross-domain concepts 2009. Area (2009) 1–47
14. Guidelines, S.C.o.: Annex 2: cross-domain code lists 2009. Area (2009)
15. Stahl, A., Roth-Berghofer, T.: Rapid prototyping of cbr applications with the open source tool mycbr. In Althoff, K.D., Bergmann, R., Minor, M., Hanft, A., eds.: ECCBR. Volume 5239 of Lecture Notes in Computer Science., Springer (2008) 615–629
16. Lenz, M.: Case retrieval nets as a model for building flexible information systems (1999)
17. Roth-Berghofer, T.R.: Explanations and case-based reasoning: Foundational issues. In Funk, P., Gonzlez Calero, P.A., eds.: Advances in Case-Based Reasoning. Volume 3155 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2004) 195–209
18. Bahls, D., Roth-Berghofer, T.: Explanation support for the case-based reasoning tool mycbr. Proceedings of the TwentySecond AAAI Conference on Artificial Intelligence July 2226 2007 Vancouver British Columbia Canada (2007) 1844–1845