End-User Access to Multiple Sources – Incorporating Knowledge Discovery into Knowledge Management

Katharina Morik, Christian Hüppe and Klaus Unterstein

University of Dortmund, Computer Science Department, LS VIII
morik@ls8.informatik.uni-dortmund.de
http://www-ai.cs.uni-dortmund.de

Abstract. The End-User Access to Multiple Sources, the EAMS system integrates document collections in the internet (intranet) and relational databases by an ontology. The ontology relates the document with the database world and generates the items in the user interface. In both worlds, machine learning is applied. In the document world, a learning search engine adapts to user behavior by analysing the click-through-data. In the database world, knowledge discovery in databases (KDD) bridges the gap between the fine granularity of relational databases and the coarse granularity of the ontology. KDD extracts knowledge from data and therefore allows the knowledge management system to make good use of already existing company data.
The EAMS system has been applied to customer relationship management in the insurance domain. Questions to be answered by the system concern customer acquisition (e.g., direct marketing), customer up and cross selling (e.g., which products sell well together), and customer retention (here: which customers are likely to leave the insurance company or ask for a return of a capital life insurance). Documents about other insurance companies and demographic data published in the internet contribute to the answers as do the results of data analysis of the company's contracts.

1 Introduction

Knowledge management is the acquisition, offering, distribution, and maintenance of knowledge. Diverse users are to exploit knowledge from diverse sources for their working procedures. In the EAMS system system, we personalize the results of document retrieval to diverse user types. Since the need for personalization is not questioned, we do not introduce into this part of our work here. The second contribution to knowledge management, however, the integration of relational databases into the content of knowledge management systems is a new challenge which requires some justification. Hence, we introduce into this new aspect of knowledge acquisition, offering, and maintenance.

Knowledge management establishes new collections of information, e.g., experience reports [1] or skill profiles, most often in the form of documents that are accessed via the intranet [2], [3], [4]. Business processes are modeled and used for the distribution of knowledge as well as for the integration of knowledge sources [5], [6]. Even laws and company regulations are formalised in order to justify and modify certain steps in a business process [7]. However, the preparation of new information is time consuming and could possibly lead to yet another isolated information source in the broad range of a company's systems. Therefore, the integration of given data sources is considered the major challenge of knowledge management systems. Technically, the integration is enabled by the use of ontologies ([8], originally called knowledge interchange formats [9], [10], or reference schemas [11]) together with wrappers for the knowledge source [12] or explicit annotations [13]. Some systems implement agents that gather, classify, and enter information into a memory that is organised according to an ontology [14], [15], [16]. These approaches are almost always accessing document collections, although XML document collections are sometimes called databases [17]. The main data sources that actually exist in all organisations are, however, databases or data warehouses. As far as we are aware, only the TSIMMIS project aims at accessing relational databases for knowledge management [18]. Why is the most comprehensive information source of organisations that neglected?

One reason which excludes given databases from knowledge management is the different granularity of the ontology and the database. The ontology concept of a customer, for instance, does not match any attribute or relation directly, but is spread over several tables and their attributes. The translation from an ontology concept to a database query is already quite some SQL programming. The second reason is the different answer set of database and knowledge management systems. The excerpt even of aggregated data is not yet the required knowledge. We might, for instance, create a view which lists customers together with their contracts. This table corresponds to a concept and its features in the ontology. Most likely, however, the user does not want to look at that table but would like to see answers to questions like:
Which are the frequently sold products?
What are the attributes of my most frequent customers?
On-line analytical processing is capable of answering those questions using statistical procedures and the data cube. To go even further, we want to extract knowledge from the database that answers questions like the following:
Which contracts are frequently sold together?
Which customers are most likely to sell their contract back to the company before it ends?
KDD can indeed answer such questions on the basis of the given database. Each such question corresponds to a KDD case. The collection of KDD cases forms a knowledge source of the knowledge management system. In this way, we bridge the gap between data and knowledge and indirectly integrate databases as a source of knowledge into a knowledge management system.

The paper is structured as follows. Section 2 gives an overview of the system. Section 3 presents the learning search engine which adapts to differse users. Section 4 explains how we store and retrieve KDD cases and how they are adapted to updated databases (i.e., knowledge maintenance). Section 5 shows the KDD results used in our insurance knowledge management.

2 System Architecture

The EAMS system integrates the document and the database world using an ontology. The ontology concepts are linked with search strings which are sent to the search engine and with the conceptual model of the database in the form of the meta-data model M4 (cf. section 4). Figure 2 shows the system architecture. In order to apply the system to a new application, the meta-data of the database, the conceptual meta-data (ontology), the meta-data about a KDD case (the conceptual case model), and the search strings which correspond to concepts in the ontology need be entered. Currently, we have not yet implemented an editor for meta-data which eases this task. However, within the MiningMart project, a meta-data editor for relations, concepts, and cases is under development. The integration of this is prepared. The EAMS system offers in its Graphical User

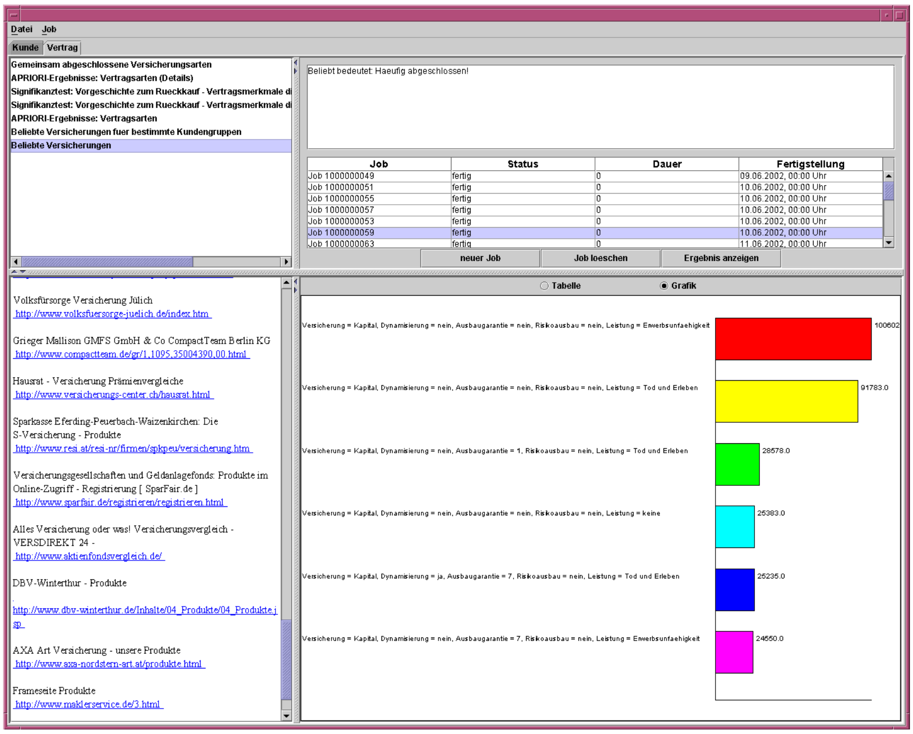

Fig. 1. Graphical user interface

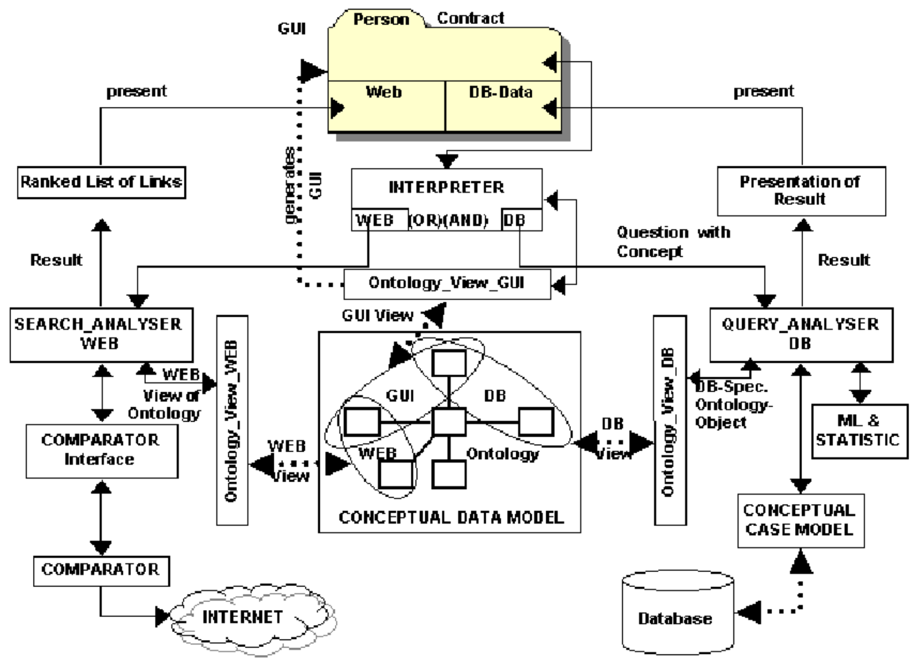

Fig. 2. System architecture

Interface (GUI) the concepts of the ontology. Each concept is presented as a folder label. The lower part of the screen is divided into one part for results from the document world and one part for results from the database world. In the middle, there is space for entering queries by selecting a query type (i.e., frequencies, segmentation, correlation, and classification). If there exist already executed KDD results for a query type and a concept, the GUI presents the KDD cases together with their date. The user may select one. If the desired combination of query and concept has not yet been executed and does not require complex preprocessing, the user may start a new KDD execution. Of course, starting a prepared KDD case anew is always possible. This is sensible, if the last execution is too long ago for the results being up to date. The results of the selected KDD case are presented in the lower database part, either as a table, or as a diagram. See figure 1 for a screenshot of the GUI.

3 Learning Search Engine

Users of the Eams system are provided with access to a document collection. If the document collection has already been indexed by a search engine – as is the world wide web – then such a search engine can be applied. All that is needed is a search string which is stored at a concept of the ontology. In order to personalise the result, the ranking of documents can be changed according to the selection of a user. If an intranet search engine is used, its retrieval function can be learned from clickthrough data. The user selection from a large body of documents can be used as training information in order to optimise the given retrieval software. Both cases demand an algorithm which computes a ranking of documents from user selections. Thorsten Joachims has developed such an algorithm in the form of a variant of his support vector machine (SVM) [19]. Its result is not the classification into a binary target concept, but a binary ordering relation. It minimises the distance between the optimal ordering and the computed ordering analogously to the error minimisation between the true and the computed classification.

The optimal ordering of documents cannot be directly derived from clickthrough data, since the user does not click on a link presented very low in the ordered link list, even if it would be the most relevant one. Users only look at the l top links. Hence, the only information that we gain from clickthrough data is a partial ordering. The clicked link which is furthest down in the list informs us that this link is more relevant to the user than the preceeding links. It does not inform us that it is the most relevant link concerning the overall link list.

For example, the third and fifth link in a list have been selected by a user. The partial ordering is then the following:

$r : link_3 < link_2, link_3 < link_1, link_5 < link_4, link_5 < link_2, link_5 < link_1$

Joachims transforms partial orderings into the input of the SVM. For each query q_i to the document collection, such a partial ordering r_i is computed from the clicked links. The input of the SVM becomes a set $(q_1, r_1), (q_2, r_2), ..., (q_n, r_n)$. The retrieval function to be learned delivers a ranking $r_{f(q)}$.

The learning result is used in order to display the relevant documents as a list which is ordered according to the learned ranking. Since the retrieval function is learned for particular users from their data, the presentation of links is adapted to their relevance assessments. This is the personalisation which we have incorporated in the EAMS system.

4 Knowledge Discovery in Databases

KDD extracts knowledge from databases. It answers high-level questions on the basis of low-level data. The question types correspond to *KDD tasks*. Some of them are:

- Frequencies of attribute values given some other attributes' values, e.g., the number of blue cars given a car series and a year of sales;
- Segmentation (clustering or subgroup detection), e.g., characteristics of subgroups of customers;
- Correlation (or association) of attributes, e.g., products that are frequently sold together (basket analysis);
- Classification (or prediction) of a target attribute.

Each task is solved by several methods. Given that the data are already well prepared, the data mining step of KDD applies an algorithm that performs the task. Currently, the EAMS system uses the following *data mining tools*:

- statistical functions of Oracle and statistical stored procedures for simple frequencies,
- data cube [20], [21] for frequencies,
- APRIORI [22] for correlations,
- C4.5 [23] for classification in the weka-implementation J48, and
- mySVM [24] for classification.

A KDD case is a sequence of steps which lead from the original database tables to an evaluated result. Where machine learning focuses on the data mining step, KDD carefully designs the preprocessing steps which prepare the data. Usually, these consume up to 80 percent of the effort. The design of the appropriate sequence for a given analysis task and database is difficult. For each step the appropriate procedure must be selected. A procedure may be an SQL procedure, a simple statistical method, or a learning method. The best parameters of the method have to be determined. Each step requires several trials.

In the context of knowledge management, we cannot expect users to interactively apply KDD techniques in order to receive answers from the database. KDD cases must be prepared by the database department of a company and be stored within the knowledge management system. A user then selects a case which answers his or her question. The management of KDD cases is part of the knowledge management: a KDD case must be related to the ontology, be stored and retrieved accordingly. For the integration as introduced in section 1, it is not sufficient to store experience about KDD cases in the internet as done in [1]. The results of a KDD case change as soon as the content of the database is changed. Therefore, a stored question together with a KDD result will soon be outdated. Hence, the KDD case should be stored as a sequence of steps which is executable on the database.

The MiningMart project [25], [26] develops a system which stores executable KDD cases. The kernel of the project is an ontology of cases,i.e., the model of meta-data which describe a case. The model of meta-data, M4, is structured into a part describing the database and a part describing the sequence of steps. The meta-data of the database are structured into the conceptual level and the relational level. The conceptual level could be considered an ontology of the domain. The relational level describes the database with its relations and attributes. Most of the relational level comes along with standard database systems. The links between conceptual and relational meta-data are part of an application's meta-data. The sequence of steps is characterised with respect to the conceptual level of the meta-data. It is compiled into executable SQL code for the relational data and into calls to external data mining tools. The same conceptual meta-data can apply to several different relational meta-data. This eases the adaptation of a case to a new relational database, which fits the conceptual model. An engineer must only connect the conceptual meta-data to the own relational meta-data.

For our EAMS system, the storage of KDD cases is the important notion of MiningMart. The KDD cases which we have developed for the insurance application (see section 5) are linked with the ontology. A KDD case can be selected together with its already computed results or run on the database anew, if its content has changed. All the trials and errors of case design have been made by others[1], the user just selects a question from the menu and receives the results[2]. Minor modifications of a case can easily be done by the users. They select an ontology concept and a given KDD task and modify the parameters or the particular concept's features about which KDD should learn.

5 Application

The application we have developed is the support of employees of an insurance company. The main *application objectives* are:

- Customer acquisition,
- customer up and cross selling, and
- customer retention.

Different users are working towards these objectives. They are supported by the knowledge management system which provides them with up to date information about the company's customers (database world) as well as the competetors offers, demographic overviews, and related journal papers (internet world). We collaborated with the SwissLife insurance company, from where we received an anonymized database of 12 tables with at most 31 attributes and at most 1,469,978 rows. We were also given general information about the working procedures within the company.

Questions and Answers about Customers

Companies need information about their customers for offering new products to existing customers. It could be interesting to have a detailed analysis of the customers which allows the end user to look at the entire data set from a bird's eye perspective. **Frequencies** of attribute values like sex, family status, age class or profession group are shown. More complex frequencies can be displayed using multidimensional data cubes according to [27]. An advantage of the EAMS system is, that it is now possible to find out whether the customer distribution corresponds to that of the total population or not. The user might apply the integrated search engine for finding demographical information about population distribution.

[1] Here, we were the case designers, see sections 5 and 5. In general, a knowledge management team in cooperation with the database department would provide users with the required cases.

[2] In the screenshot 1 the question list is displayed in the upper half left, the list of jobs shows executed cases.

In order to acquire new content for the knowledge management system we have developed a KDD case for customer **segmentation**. In order to prevent attrition or churn it might be proper to provide special offerings to the different customer groups. We first choose different properties which allow a segmentation of the data. Some properties like profession or age have a large number of variations, so that we summarize their values into groups. Moreover, it was necessary to create a new table, in which every row contains all properties that belong to a customer. Then, we applied the Apriori algorithm to detect properties that frequently appear together. The results are frequent item-sets like

{age class: 40 to 65, male, married, insurance owner, mode of payment: pays, profession class: 1}.

If we compare the item sets found on the basis of the data about all customers with those found on the basis of only the customers who re-bought their insurcance, we encountered a significant difference for the most common group of customers (i.e., owner, payer and insured person are the same): this group makes up for 79% regarding all customers, but 94% regarding re-buy cases. In the data set reduced to surrender cases, Apriori found 91 frequent item sets. Using the criterion of certainty factors [28], the number of item-sets was reduced to 19.

{male, married, insurance owner, insured person} $\longrightarrow$ insurance surrender

For insurances it is very interesting to know if people will churn a contract, switch to other companies or keep their contract unchanged for years. The detection of high risk customers opens the possibility to offer special contracts to such people. The **prediction** of high risk customers is supplied by using two classification algorithms. On the one hand a decision tree analysis algorithm[3] and on the other hand a support vector machine[4]. However, the prediction of re-buy could not be effectively learned on the basis of customer properties, but required contract data (see below).

Questions and Answers about Contracts

In addition to the information gathered by analyzing the customers, the content needed in the knowledge management system is the structure of contracts. A contract is not a monolithical entity but is composed of different contract components, in which the contract conditions are determined. Each contract component stores the exact amount of premium, payment, for what parts does that specific component apply to, the type (pension, once-only payments, etc.) and the general framework (possibility of contract extensions, dynamic sampling, etc.). The contract itself is only an envelope around the contract components which specify the contract. **Frequencies:** The EAMS system produces a ranking

[3] J48 from the Weka project, http://www.cs.waikato.ac.nz/ml/weka

[4] mySVM according to Stefan Rüping, http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM

by using statistical methods on the frequencies of each contract component ordered by their frequency[5] of occurence for a quick overview on different contract components. We contructed 214 contract types, each being characterized by a combination of the 11 most relevant attributes.[6] These types were applied to classify all contract components with a new attribute.

Correlation: Customer relationship management includes cross selling. Cross selling is the process of offering new products and services to existing customers. One form of cross selling, sometimes called up selling, takes place, when the new offer is related to existing purchases by the customers. Using data mining for cross selling helps to predict the probability or value of a current customer buying these additional products. A data mining analysis for cross selling moves beyond repeating the analysis required for customer acquisition several times, i.e. for each additional product. Here, the key is to optimize the offerings across all customers and products. The goal is to create a win–win situation, in which both the customer and the company benefit. Analyzing previous offer sequences with data mining methods can help to determine what and when to make the next offer. This allows to carefully manage offers to avoid over-soliciting and possibly alienating customers and to keep the costs low. Therefore the EAMS system aids in finding these correlations between contracts (and components) by inspecting the contract component combinations. This is achieved by using Apriori. Again we start with the previously segmented contract components, whose types are used as items for Apriori. A transaction in Apriori terminology accords to one client (the contract is assigned to the insurance holder to get a unique assignment) and his effected insurances. The resulting frequent item sets refer to combined acquired contract types, which are displayed with their support. Most frequently a capital contract with a contract on death or life were effected together.

Prediction: Churn predicition was not possible on the basis of customer data. When learning on contract data, we again applied mySVM and adopted the TFIDF-measure from information retrieval. Each property A_i of a contract component represents a **term**. The **termfrequency** TF is the number of changes of A_i in a contract component C_j defined as:

$$TF_{C_j A_i} = |\{x \in \text{year } | A_i \text{ of } C_j \text{ was changed in year } x\}| \tag{1}$$

Therefore the **document** $\text{TFIDF}_{C_j A_i}$ describes the history of changes of a contract component C_j which is calculated by

$$\text{TFIDF}_{C_j A_i} = TF_{C_j A_i} \cdot \log\left(\frac{|C_{\text{all}}|}{|\{t \in C_{\text{all}} | A_i \text{ changed in t}\}|}\right) \tag{2}$$

Based on this representation the vector for SVM training looks like:

$$C_j : \text{TFIDF}_{C_j A_1}, \ldots, \text{TFIDF}_{C_j A_{23}}, y$$

[5] Both relative and absolute frequency is displayed. The relative frequency is calculated by comparing the absolute frequency with the total number of contract-components.

[6] Note, that most of the combinations of attributes do not occur.

y is the target attribute (insurance surrender occured: yes/no). mySVM[7] delivered excellent results using a linear kernel (recall $= 0,4452$; accuracy $= 0,8646$; absolute error $= 0,3375$). These results are sufficient for predicting an insurance surrender.

Web support: Informations from different competing insurance companies about products are rare. They aren't stored in a textual form to get processed by search engines so the results are insufficient. Therefore the aid from the EAMS system for the product designer on informations about contracts coming from the World Wide Web is rather small. Hence, the use of the learning search engine could not ne tested in this application.

6 Conclusion

In this paper, we have presented a principled approach of how to make available both, document collections and databases to users of a knowledge management system. Both types of information are glued together by the ontology which also generates the user interface. We have argued that the link to databases directly is not appropriate for the users' needs[8]. Instead, KDD bridges the gap between low-level data and high-level information needs of users. A case base of KDD cases is necessary for providing users with knowledge on the basis of their company's databases. Such a case base can be built incrementally, preparing a new case, when necessary. The acquisition of knowledge is eased since the data already given are used as its source. The maintenance of knowledge is guaranteed, since the cases of KDD are operational and run anew on the changed database whenever wanted.

For the personalisation of the ranked lists of documents, we have applied the system of Thorsten Joachims. This is particularly interesting, because it eases the search also in intranets. Personalisation of KDD results seemed not necessary. The presentation form can be selected from a menu, and so are the parameters of the data mining tools.

We have illustrated our approach with an application at the insurance company SwissLife. Since the colleagues with whom we were cooperating have left the insurance company, we could not validate our EAMS system in a practical test by real end-users. We regret this, but we hope that the ideas that were implemented in the EAMS system are nevertheless convincing.

Acknowledgment The EAMS system has been developed in a one year students project (PG402) at the university of Dortmund by Fabian Bauschulte, Ingrid Beckmann, Christian Hüppe, Zoulfa El Jerroudi, Hanna Köpcke, Phillip Look,

[7] kernel: dot, radial; mode: pattern; sample: random-sample (biased); evaluation on population.

[8] In our example, the aggregation and transformation of data into an informative and interesting form uses about 3000 lines PL/SQL for each case. No end-user could afford this affort of programming!

Boris Shulimovich, Klaus Unterstein, and Daniel Wiese under the supervision of Katharina Morik and Stefan Haustein. We want to thank Thorsten Joachims for placing at our disposal his excellent tool for the optimisation of retrieval functions in search engines. We also thank Regina Zücker, Jörg-Uwe Kietz, and Peter Brockhausen for their information about the insurance practice and the anonymised database.

References

1. Bartlmae, K., Riemenschneider, M.: Case Based Reasoning for Knowledge Management in kdd projects. In Reimer, U., ed.: Proc. of the Third Int. Conf. of Practical Aspects of Knowledge Management. (2000)
2. R., R.J.: Skills Management at Swiss Life. In: LLWA 01 - Tagungsband der GI-Workshopwoche Lernen - Lehre - Wissen Adaptivität. (2001) 227–236
3. Sure, Y., Maedche, A., Staab, S.: Leveraging Corpoate Skill Knowledge - From ProPer to OntoProPer. In Reimer, U., ed.: Proc. of the Third Int. Conf. of Practical Aspects of Knowledge Management. (2000)
4. Becerra-Fernandez, I.: Facilitating the Online Search of Experts at NASA using Expert Seeker People-Finder. In Reimer, U., ed.: Proc. of the Third Int. Conf. of Practical Aspects of Knowledge Management. (2000)
5. Karagiannis, D., Telesko, R.: The EU-Project PROMOTE: A Process-Oriented Approach for Knowledge Management. In Reimer, U., ed.: Proc. of the Third Int. Conf. of Practical Aspects of Knowledge Management. (2000)
6. Rainer, T., Dimitris, K.: Realising process-oriented knowledge management: Experiences gained in the PROMOTE-project. In: LLWA 01 - Tagungsband der GI-Workshopwoche Lernen - Lehre - Wissen Adaptivität. (2001) 206–212
7. Margelisch, A., Reimer, U., Staudt, M., Vetterli, T.: Cooperative Support for Office Work in the Insurance Business. Technical report, Swiss Life, Information Systems Research (1999)
8. Benjamins, V.R., Fensel, D., Gómez Péres, A.: Knowledge Management through Ontologies. In Reimer, U., ed.: Proceedings of the Second International Conference on Practical Aspects of Knowledge Management. (1998)
9. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N., Poli, R., eds.: Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands, Kluwer Academic Publishers (1993)
10. Genesereth, M., Singh, N.: A knowledge sharing approach to software interoperation (1993)
11. Genesereth, M.R., Keller, A.M., Duschka, O.M.: Infomaster: an information integration system. In: SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA. (1997) 539–542
12. Vdovjak, R., Houben, G.J.: RDF-based architecture for semantic integration of heterogeneous information sources. In: Workshop on Information Integration on the Web. (2001) 51–57
13. York, S.: On-To-Knowledge. In: LLWA 01 - Tagungsband der GI-Workshopwoche Lernen - Lehre - Wissen Adaptivität. (2001) 213–216

14. Cranefield, S., Haustein, S., Purvis, M.: Uml-based ontology modelling for software agents. In: Proceedings of the Autonomous Agents 2001 Workshop on Ontologies in Agent Systems. (2001) http://cis.otago.ac.nz/OASWorkshop/Papers/oas01-27-cranefield.pdf.
15. Haustein, S., Lüdecke, S.: Towards Information Agent Interoperability. In Klusch, M., Kerschberg, L., eds.: Cooperative Information Agents IV – The Future of Information Agents in Cyberspace. Volume 1860 of LNCS., Boston, USA, Springer (2000) 208 – 219
16. Gandon, F., Dieng, R., Corby, O., Giborin, A.: A Multi-Agent System to Support Exploiting an XML-based Corporate Memory. In Reimer, U., ed.: Proc. of the Third Int. Conf. of Practical Aspects of Knowledge Management. (2000)
17. Christophides, V., et al.: Community webs (c-webs): Technological assessment and system architecture. Technical report, FORTH Institute of Computer Science (2000)
18. Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J.D., Widom, J.: The TSIMMIS project: Integration of heterogeneous information sources. In: 16th Meeting of the Information Processing Society of Japan, Tokyo, Japan (1994) 7–18
19. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of Knowledge Discovery in Databases. (2002)
20. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., rali Venkatrao, M., Pellow, F., Pirahesh, H.: Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-to tals. J. Data Mining and Knowledge Discovery **1** (1997) 29–53
21. Harinarayan, V., Rajaraman, A., Ullman, J.D.: Implementing data cubes efficiently. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996. (1996) 205–216
22. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large data bases. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94), Santiago, Chile (1994) 478–499
23. Quinlan, J.: Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research **4** (1996) 77–90
24. Rüping, S.: mySVM-Manual. Universität Dortmund, Lehrstuhl Informatik VIII. (2000) http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/.
25. Kietz, J.U., Zücker, R., Vaduva, A.: Mining Mart: Combining Case-Based-Reasoning and Multi-Strategy Learning into a Framework to reuse KDD-Application. In Michalki, R., Brazdil, P., eds.: Proceedings of the fifth International Workshop on Multistrategy Learning (MSL2000), Guimares, Portugal (2000)
26. Morik, K.: The representation race - preprocessing for handling time phenomena. In de Mántaras, R.L., Plaza, E., eds.: Proceedings of the European Conference on Machine Learning 2000 (ECML 2000). Volume 1810 of Lecture Notes in Artificial Intelligence., Berlin, Heidelberg, New York, Springer Verlag Berlin (2000)
27. Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M.: Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. Data Mining and Knowledge Discovery **1** (1997) 29 – 54
28. Fernando, B., Ignacio, B., Daniel, S., Maria-Amparo, V.: A New Framework to Assess Association Rules. In et al., H.F., ed.: Advances in Intelligent Data Analysis. Volume 2189 of LNCS. Springer Verlag Berlin (2001) 95–104